Data-Driven Methods for Investigating and Modelling Complex Dynamical Systems

Davide Maria Fabris
Mechanical Engineering Department
Politecnico di Milano
Milan, Italy
davidemaria.fabris@polimi.it

***Abstract*—This paper serves as a guideline for the application of data-driven methods in the analysis and in the discovery of the dynamical behaviour of complex systems. Moreover, this work aims at the application of some of the most diffused methods for performing data analysis and processing. Starting from raw data, the dimensionality of the systems is reduced and, in the newly discovered reference frame, valuable insights are sought after. Several methods are considered while investigating the systems at hand, ranging from the Singular Value Decomposition to Neural Networks. Eventually, the obtained results are presented, mainly in a graphical manner, so to possibly ease their understanding.**

***Index Terms*—Dynamic Mode Decomposition, Time-Delay Embedding, Koopman Theory, Sparse Identification of Nonlinear Dynamics, Neural Networks.**

I. Introduction

This document is meant to represent a reference for the application of data-driven methods for the analysis of the dynamics of complex systems. Some of the explored data derive from measured quantities, others are obtained by means of the solution of Ordinary Differential Equations (ODEs) or Partial Differential Equations (PDEs). Several case studies are considered, each of these is investigated by relying on mathematical and numerical algorithms.

The first data set is constituted by the historical evolution of the Canadian Lynx and the Snowshoe Hare populations, which were measured from 1845 to 1903. This collection of data is investigated by also making reference to the Lotka-Volterra system of ODEs. The second data set is obtained through the solution of the Kuramoto-Sivashinsky equation, a PDE describing the diffusive instability phenomenon in a laminar flame front. The third data set is gathered by solving a Reaction-Diffusion equation. The fourth data set is derived from the Lorenz system of ODE equations for different parameters and initial conditions, resulting in chaotic solutions. The fifth data set is composed by the spatio-temporal evolution of a Belousov-Zhabotinsky nonlinear chemical oscillator.

Data are explored by means of the Dynamic Mode Decomposition (DMD) algorithm, the Time-Delay Embedding formulation of DMD, the Sparse Identification of Nonlinear Dynamics (SINDy) and eventually via Neural Networks (NN).

The document is structured as follows. Section II introduces the dynamical systems and the data sets that were analysed, it also lays the theoretical foundations of the harnessed methods. Section III reports the algorithmic setup and it highlights the computational results that were achieved. In Section IV the conclusions are drawn.

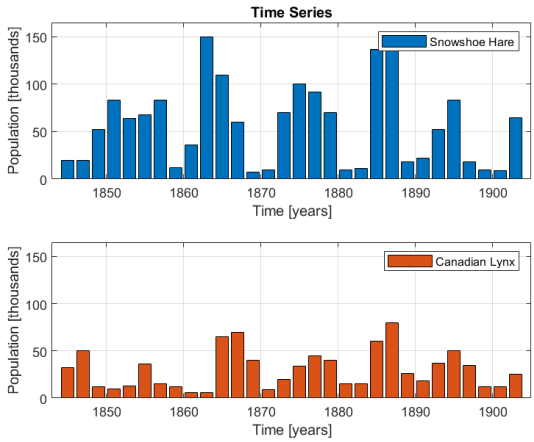

Fig. 1. Historical evolution of the Snowshoe Hare and Canadian Lynx populations from 1845 to 1903. Data are expressed in thousands of units.

II. Theoretical Background

In the following, vectors are column-wise arranged and they are identified by bold fonts (e.g. $\mathbf{u}$). The complex conjugate transpose operator is represented by $\mathbf{x}^*$, the pseudo inverse by $\mathbf{x}^\dagger$. The Hadamard and the scalar product are respectively pointed out by $\circ$ and $\cdot$, time derivatives by $\dot{}$.

A. Mathematical Models

The dynamical systems that were taken into account are presented in the following paragraphs. The Lotka-Volterra model was considered in the analysis of the Lynx-Hare data set. The Kuramoto-Sivashinsky PDE and the Lorenz system of ODEs were numerically solved to populate the relative data collections. Whereas, the data from the Belousov-Zhabotinsky model was processed without making reference to the chemical aspects. Each mathematical model is reported together with a representation of the associated data.

Hare-Lynx: Being x and y the prey and predator populations, the dynamics of the Prey-Predator biological system is modelled by the Lotka-Volterra system of ODEs (1). Fig. 1 depicts the data set that was considered for the application of

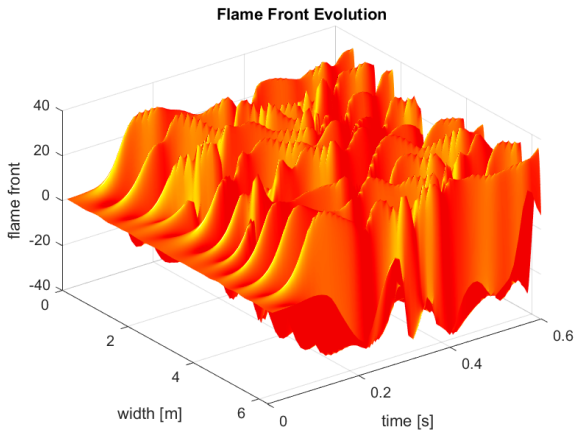

Fig. 2. Example of solution of the Kuramoto-Sivashinsky equation.

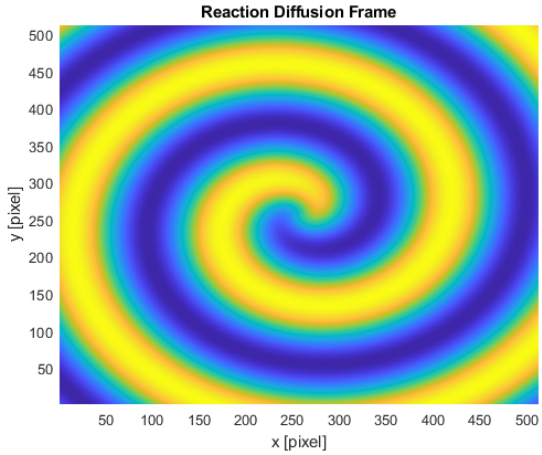

Fig. 3. Single time frame of the reaction-diffusion xy plane.

the data-driven methods in the field of biological interaction among populations.

$$\begin{cases} \dot{x} = (b - py)x \\ \dot{y} = (rx - d)y \end{cases} \tag{1}$$

The positive parameters b, p, r and d describe the influence of the single populations x, y and of their product xy on the dynamics of the two species. The first order system of ODEs can be rewritten in matrix form (2). Note that $\mathbf{x}$ and $\mathbf{y}$ refer to the whole time evolution of the two populations.

$$\begin{bmatrix} \dot{\mathbf{x}}^* \\ \dot{\mathbf{y}}^* \end{bmatrix} = \begin{bmatrix} b & 0 & -p \\ 0 & -d & r \end{bmatrix} \begin{bmatrix} \mathbf{x}^* \\ \mathbf{y}^* \\ \mathbf{x}^* \circ \mathbf{y}^* \end{bmatrix} \tag{2}$$

This expression of the Lotka-Volterra system of ODEs is going to be considered in the following dissertation, specifically in the Lotka-Volterra model fitting.

Kuramoto-Sivashinsky: The Kuramoto-Sivashinsky equation (3) describes the diffusive instability in a laminar flame front. The spatio-temporal dynamics of the flame front, identified by $u(t, x)$, features chaotic behaviour. Fig. 2 depicts a possible solution of the equation.

$$u_t + \nabla^4 u + \nabla^2 u + \frac{1}{2}|\nabla u|^2 = 0 \tag{3}$$

Where ∇^2 is the Laplace operator and its square ∇^4 is the bi-harmonic or bi-laplacian operator.

Reaction-Diffusion: The reaction-diffusion system is solved via a numerical integrator and the solution in then analysed by means of the data-driven methods introduced in the following sections. Fig. 3 depicts a 2D plot of a single time frame solution of the system. Starting from this large data collections, the evolution is analysed and the principal components of the dynamics are highlighted. The same information can be depicted in 3D per each time frame.

Lorenz: The Lorenz system of ODEs (4) features chaotic solutions for certain parameters and initial conditions values.

$$\begin{cases} \dot{x} = \sigma(y - x) \\ \dot{y} = \rho x - xz - y \\ \dot{z} = xy - \beta z \end{cases} \tag{4}$$

For $\sigma = 10$, $\beta = 8/3$ and $\rho = 28$ and nearby values, the solutions of the Lorenz system show chaotic behaviour. Fig. 4 depicts a 3D representation of the chaotic solution obtained from the Lorenz system of ODEs.

Belousov-Zhabotinsky: Consider these chemical reactions.

$$\begin{aligned} A &\to B \\ 2X + Y &\to 3X \\ B + X &\to Y + D \\ X &\to E \end{aligned} \tag{5}$$

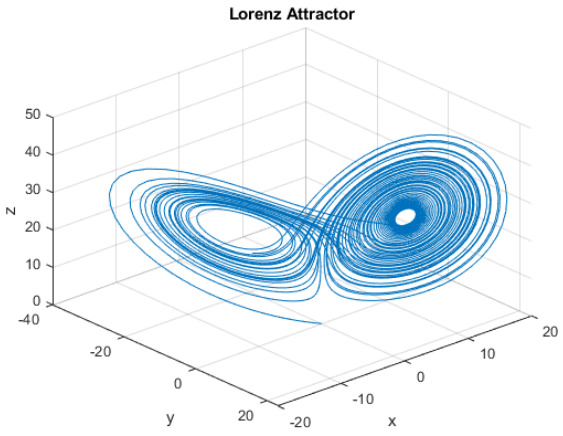

Fig. 4. Solution of the Lorenz system of ODEs showing chaotic behaviour. Different initial conditions $[x_0, y_0, z_0]$ lead to different trajectories $[x(t), y(t), z(t)]$. The solution was calculated for $\sigma = 10$, $\rho = 28$, $\beta = 8/3$.

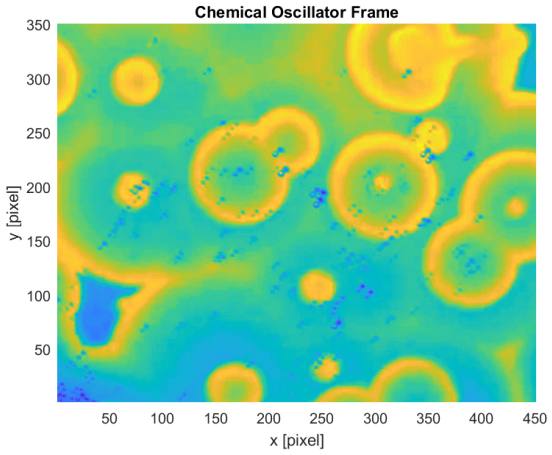

Fig. 5. Single time frame of the Chemical Oscillator xy plane.

Under the hypothesis of A and B being in vast excess (i.e. $\{A\}$ and $\{B\}$ constant), the dynamics of the chemical concentrations $\{X\}$ and $\{Y\}$ is described by the following system of ODEs (6). Fig. 5 shows a 2D depiction of a time frame of the solution of the Belousov-Zhabotinsky system (6).

$$\begin{cases} \dfrac{d}{dt}\{X\} = \{A\} + \{X\}^2\{Y\} - \{B\}\{X\} - \{X\} \\ \dfrac{d}{dt}\{Y\} = \{B\}\{X\} - \{X\}^2\{Y\} \end{cases} \tag{6}$$

B. Data-Driven Methods

The principal methods that were applied in order to extract meaningful insights from the data sets are presented below. The Dynamic Mode Decomposition method and its formulation with the Time-Delay Embedding are introduced first. These two lead to the identification of the best fit linear reconstruction of the system dynamics. Then, the building elements of the Sparse Identification of Nonlinear Dynamics are laid out. This method allows to identify the underlying dynamical components of the measurement data matrix X (8). Eventually, a brief introduction to Deep Learning and Neural Networks is provided. The main objective of these methods is to model and to possibly reconstruct the dynamics of a given system. In general, the governing equations of the system can be expressed as reported below (7).

$$\frac{d}{dt}\mathbf{x} = f(\mathbf{x}, t, \mu) \tag{7}$$

Dynamic Mode Decomposition (DMD): For the Dynamic Mode Decomposition consider X being an $m \times n$ matrix, whose k-th column $\mathbf{x}_k$ contains the values of the m measured variables at the k-th time instant [1] [2].

$$X = [\mathbf{x}_1,\ \mathbf{x}_2,\ \ldots\ \mathbf{x}_n] \tag{8}$$

Let now X_1 and X_2 be defined as follows.

$$\begin{aligned} X_1 &= [\mathbf{x}_1,\ \mathbf{x}_2,\ \ldots\ \mathbf{x}_{n-1}] \\ X_2 &= [\mathbf{x}_2,\ \mathbf{x}_3,\ \ldots\ \mathbf{x}_n] \end{aligned} \tag{9}$$

The objective is to identify the best-fit linear operator A, such that X_2 could be expressed as a function of X_1 (10).

$$X_2 \approx AX_1 \tag{10}$$

Mathematically, A can be determined by post-multiplying both sides of (10) by $X_1^\dagger$, resulting in the Exact DMD formulation (11).

$$A = X_2X_1^\dagger \tag{11}$$

Another possibility is to set an optimization problem in such a way that the following expression (12) is verified, where the $||\cdot||_F$ stands for the Frobenius norm. This alternative formulation is referred to as Optimized DMD.

$$A = \underset{A}{argmin}||X_2 - AX_1||_F \tag{12}$$

In both cases the dynamics of the system is extracted by means of the calculation of the eigenvalues and eigenvectors of matrix A. Nevertheless, the solution of (11) and (12) implies a conspicuous computational burden, especially in the case of high dimensional systems. This issue lead to the development of several alternative methods for performing DMD.

One of these alternative algorithms relies on the Singular Values Decomposition, it is referred to as SVD-based DMD. It begins with the computation of the SVD of X_1.

$$X_1 = U\Sigma V^* \tag{13}$$

By considering the first r singular values of Σ, the problem can be projected into a lower dimensional space. The results of the SVD of $X1$ are thus reduced to $\tilde{\Sigma}$, $\tilde{U}$ and $\tilde{V}$. By definition, matrix A is defined as reported in (14), while its projection in the r dimensional space $\tilde{A}$ can be computed as shown in (15).

$$A = X_2V\Sigma^{-1}U^* \tag{14}$$

$$\tilde{A} = \tilde{U}^*A\tilde{U} = \tilde{U}^*X_2\tilde{V}\tilde{\Sigma}^{-1} \tag{15}$$

The spectral decomposition of $\tilde{A}$ is calculated, identifying its eigenvalues and eigenvectors matrices Λ and W (16).

$$\tilde{A}W = W\Lambda \tag{16}$$

At this point, it is possible to re-project the low dimensional modes into the high-dimensional modes of A, obtaining the full-rank eigenvectors matrix Φ.

$$\Phi = X_2\tilde{V}\tilde{\Sigma}^{-1}W \tag{17}$$

Once the DMD modes of A are computed, these can be exploited for example in order to do future state forecasting.

Algorithm 1 SVD-based DMD

Require: X_1 *and* X_2
$U, \Sigma, V \leftarrow svd(X_1)$
Select first r singular values of Σ
$\tilde{\Sigma} \leftarrow \Sigma(1:r, 1:r)$
$\tilde{U} \leftarrow U(:, 1:r)$
$\tilde{V} \leftarrow V(:, 1:r)$
$A = X_2 V \Sigma^{-1} U^*$
$\tilde{A} \leftarrow U^* A U = U^* X_2 V \Sigma^{-1}$
$W, \Lambda \leftarrow eig(\tilde{A})$
$\Phi \leftarrow X_2 V \Sigma^{-1} W$
return Φ

The following equations (18) and (19) report how the reconstruction is performed.

$$\mathbf{x}_k = \sum_{j=1}^{r} \phi_j \lambda_j^{k-1} b_j = \Phi \Lambda^{k-1} \mathbf{b} \tag{18}$$

Consider that $\omega = log(\lambda)/\Delta t$, thus $\Omega = log(\Lambda)/\Delta t$.

$$\mathbf{x}(t) = \sum_{j=1}^{r} \phi_j e^{\omega_j t} b_j = \Phi exp(\Omega t) \mathbf{b} \tag{19}$$

Where the modulation terms $\mathbf{b}$ are obtained from the initial condition $\mathbf{x}_1$ (20). This relation comes from the imposition of k and t to be null, respectively in (18) and in (19).

$$\mathbf{b} = \Phi^{\dagger} \mathbf{x}_1 \tag{20}$$

Time-Delay Embedding: Koopman theory looks for a proper function $g(\cdot)$ (i.e. coordinate transformation) to be applied to each $\mathbf{x}_k$ column of X such that a better representation of the data can be achieved through DMD. An alternative formulation is the introduction of the time-delay embedding [3]. In this case, starting from the data matrix X, the Henkel matrix H is constructed as follows (21).

$$H = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_p \\ \mathbf{x}_2 & \mathbf{x}_3 & \dots & \mathbf{x}_{p+1} \\ \mathbf{x}_3 & \mathbf{x}_4 & \dots & \mathbf{x}_{p+2} \\ \vdots & \vdots & \dots & \vdots \\ \mathbf{x}_{q+1} & \mathbf{x}_{q+2} & \dots & \mathbf{x}_{p+q} \end{bmatrix} \tag{21}$$

The SVD-based DMD algorithm can then be applied to the H matrix in order to identify the DMD modes matrix Φ.

Sparse Identification of Nonlinear Dynamics (SINDy): For the Sparse Identification of Nonlinear Dynamics [4] consider the general formulation for the dynamics of a system (7), whose governing equations are a function of the state $\mathbf{x}$ and of time t and of some parameters μ.

$$\frac{d}{dt}\mathbf{x} = f(\mathbf{x}, t, \mu) \tag{22}$$

Introduce now an $m \times l$ matrix $T(\mathbf{x})$, which is column-wise populated with several functions $t_i(\mathbf{x})$, limited by the user's knowledge (e.g. $sin(\mathbf{x})$, $cos(\mathbf{x})$, $sin(\mathbf{x}) \circ cos(\mathbf{x})$, etc.).

$$T(\mathbf{x}) = [t_1(\mathbf{x}),\ t_2(\mathbf{x}),\ \dots\ t_l(\mathbf{x})] \tag{23}$$

The dynamics is then approximated as the superposition of the nonlinear components of $T(\mathbf{x})$.

$$f(\mathbf{x}, t) \approx \sum_{i=1}^{p} t_i(\mathbf{x}) \xi_i = T(\mathbf{x}) \boldsymbol{\xi} \tag{24}$$

The relative weight $\boldsymbol{\xi}_i$ of each $t_i(\mathbf{x})$ is defined by solving the corresponding system of equations with respect to $\boldsymbol{\xi}$. The $m \times n$ data matrix X is given and it can be exploited in order to define its first order time derivative X'. The central finite difference might be used, the fourth order approximation can be applied to define the k-th column of X'.

$$\mathbf{x}'_k \approx \frac{1}{12h}\left(\mathbf{x}_{k-2} - 8\mathbf{x}_{k-1} + 8\mathbf{x}_{k+1} - \mathbf{x}_{k+2}\right) \tag{25}$$

In the following, an alternative formulation is going to be applied, in which the discrete form of the system is considered. Specifically, the $\Theta(X_1)$ matrix is defined similarly to the $T(\mathbf{x})$ matrix. Note that Θ is a function of X_1, therefore it is column-wise populated by matrices $\theta_i(X_1)$.

$$\Theta(X_1) = [\theta_1(X_1),\ \theta_2(X_1),\ \dots\ \theta_l(X_1)] \tag{26}$$

The following relation (27) can be constructed and the solution of the corresponding system with respect to $\boldsymbol{\xi}$ provides the weighting coefficient associated to every column of $\Theta(X_1)$. The solution can be computed through (28) or even promoting sparsity by means of the Lasso regularized regression (29).

$$X_2 = \Theta(X_1) \boldsymbol{\xi} \tag{27}$$

$$\boldsymbol{\xi} = \Theta(X_1)^{\dagger} X_2 \tag{28}$$

$$\boldsymbol{\xi} = lasso(\Theta(X_1), X_2) \tag{29}$$

Neural Networks: Provided with a training data set, NNs are able to learn to map input data to the associated output data [5], [6]. The input data i is fed to the Neural Network via the input layer. The input information is non-linearly processed by the hidden layers. The output layer returns the associated result o. The training set is used to iteratively update the inner connections (i.e. weights W, biases B) of the NN in such a way that a cost function $C(W, B, i, o, \tilde{o})$ is minimized and the mapping among input and output data is learnt. The learning procedure is composed by a forward propagation and a backward propagation. In the former, the outputs $\tilde{o}$ are calculated from inputs i and these are compared to the ground-truth outputs o. In the former, the gradient of the cost

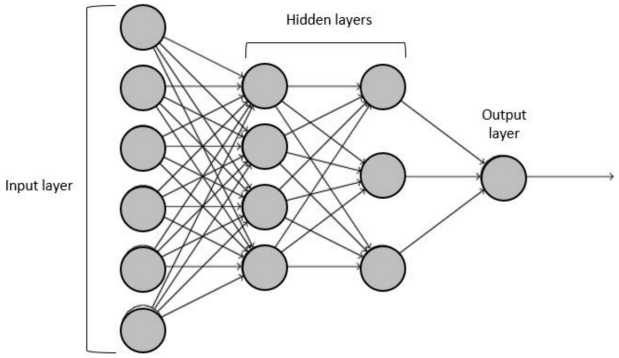

Fig. 6. Schematic representation of a Neural Network architecture.

function is computed and W and B of the network are updated proportionally to the learning rate η (30).

$$\begin{cases} W' = W - \eta \nabla C_W \\ B' = B - \eta \nabla C_B \end{cases} \tag{30}$$

Each neuron in the hidden layers in Fig. 6 is defined by associating it to an activation function. This nonlinear function is fed with the linear combination of the inputs from the precedent layer, weighted by specific layer weights **w** and biases **b**. Some of the most commonly considered activation functions are the sigmoid (σ), the hyperbolic tangent (tanh) and the rectified linear unit (ReLU).

III. Computational Implementation and Results

A. Hare-Lynx

For the Lynx-Hare data set the first objective was the evaluation of the DMD algorithm. Then, the Time Delay Embedding version of the DMD was applied to the data matrix. Successively, the Lotka-Volterra model was fitted to the data. Eventually, the SINDy method was applied in order

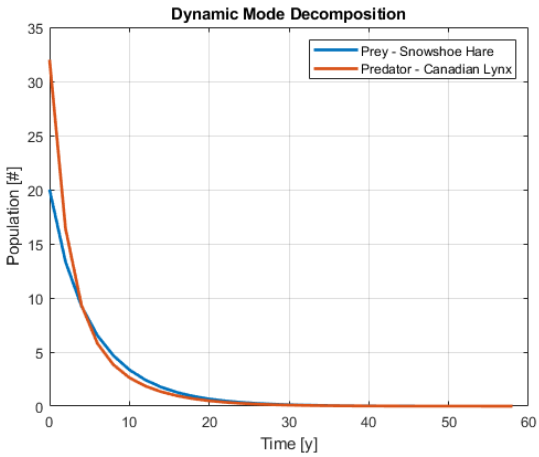

Fig. 7. DMD-based reconstruction considering the data matrix data X.

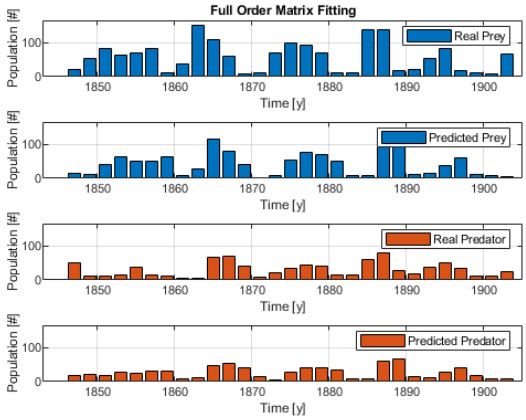

Fig. 8. Fitting of the full order A matrix to the data matrix X.

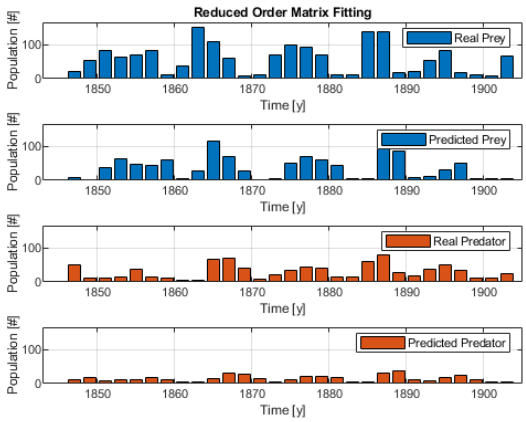

Fig. 9. Fitting of the reduced order $\tilde{A}$ matrix to the data matrix X.

to exploit the regressed non linear system for operating future state prediction.

With reference to the previously reported DMD algorithm, the DMD modes were identified and the corresponding reconstruction was implemented. Nevertheless, since the input matrix featured limited spatial and temporal dimensions, the identified modes could not reconstruct the system dynamics. Specifically, two eigenvalues could be identified and they turned out being real and negative, therefore the reconstruction featured an exponentially decaying shape. The data was then fitted to both a full rank matrix A and to a reduced order matrix $\tilde{A}$. The results of the two fitting solutions are reported in Fig. 8 and Fig. 9. The application of the Time Delay Embedding formulation of the Koopman Theory for DMD lead to the identification of much better dynamical components, these are shown in Fig. 10.

The Lotka-Volterra model was fitted to the data, its parameters were singularly identified for the two equations of the system of ODEs. The solution of the consequently defined system of equations was solved both via the Pseudo Inverse

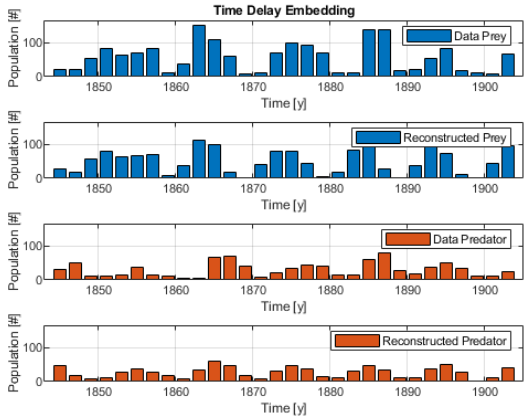

Fig. 10. Reconstruction resulting from the Time Delay Embedding DMD.

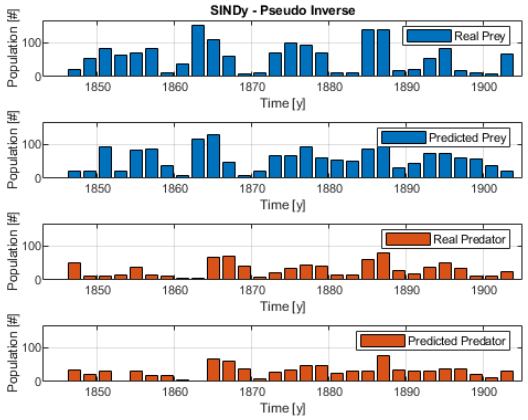

Fig. 12. Sparse identification of nonlinear dynamics via Pseudo Inverse.

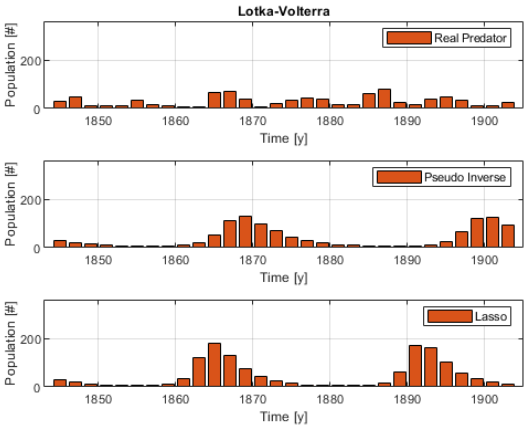

Fig. 11. Lotka-Volterra model fitting to the Predator population.

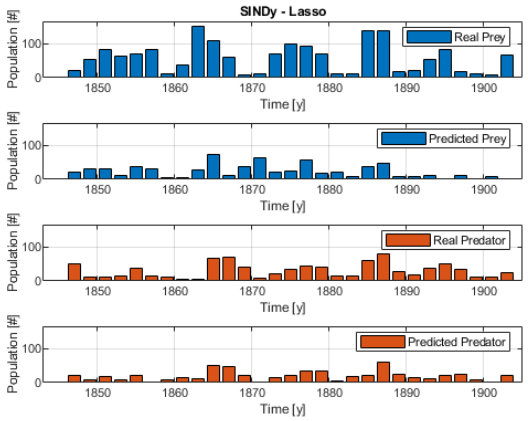

Fig. 13. Sparse identification of nonlinear dynamics via Lasso.

post-multiplication and via the Lasso sparse regression solver. Fig. 11 shows the results that were obtained by performing the fitting on the Lynx population, a similar trend was identified for the Hare population.

The Sparse Identification of Nonlinear Dynamics was applied in order to identify the dynamical components of the data matrix X. Several nonlinear function were inserted in the library matrix $\Theta(X_1)$. Fig. 12 and Fig. 13 show the obtained results from the SINDy application. In this case the Pseudo Inverse solution featured better shapes with respect to Lasso.

B. Kuramoto-Sivashinsky

For the Kuramoto-Sivashinsky equation, the objective was to train a Neural Network to advance the solution from t_k to t_{k+1}. The chosen architecture was fed with the k-th time instant solution and it was requested to output the solution at the following time instant. The gathered results did not look promising. A possible alternative way of performing this might be to reduce the dimensionality of the system and then to let the neural network learn the reduced order modelling, moving then back to the higher rank problem via the inverse SVD transformation.

On the contrary, the analysis of the reaction-diffusion system, the application of the DMD algorithm lead to rather satisfactory results. The normalized value of the Singular Values of the X_1 matrix were plotted, obtaining the classical knee-shaped curve [7]. As a result, the system dynamics was reconstructed by considering $2, 3$ and 5, since after the fifth mode, the contribution of the remaining modes was lower than 10^{-2}.

C. Lorenz

With regards to the Lorenz system of ODEs, two were the main objectives. The first task was to train a neural network architecture to be able to perform the propagation of the Lorenz solution to time instant t_{k+1}, given the solution at time instant t_k. The second objective was to train another neural network to forecast whether a lobe transition was imminent.

For the future state forecasting, the neural network was trained on randomly initialized solutions of the Lorenz system. These were generated by keeping $\sigma = 10$ and $\beta = 8/3$, while

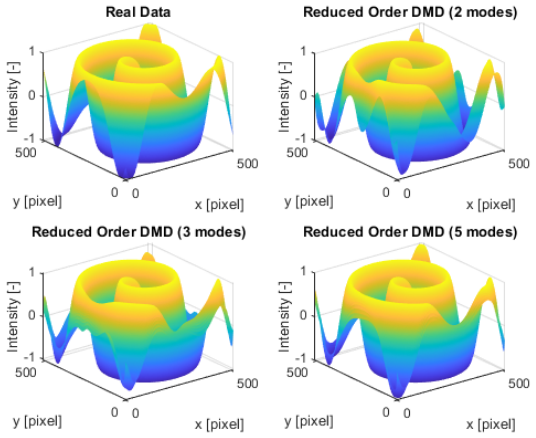

Fig. 14. Reconstruction of a single 3D frame of the Reaction-Diffusion data matrix considering 2, and 5 DMD modes.

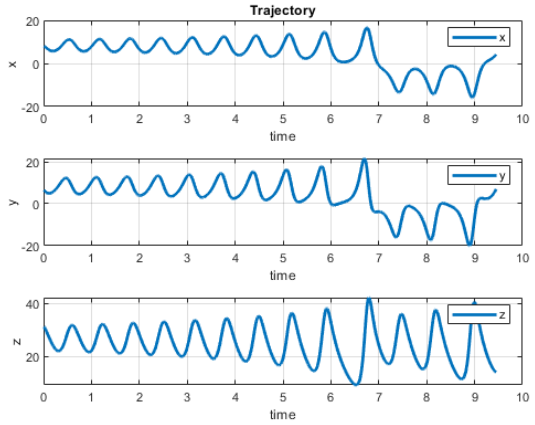

Fig. 15. Example of xyz trajectories from the Lorenz system of ODEs.

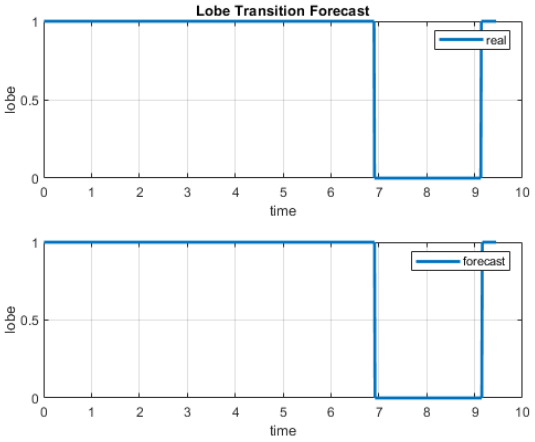

Fig. 16. Example of lobe transition forecast for the xyz trajectories in Fig. 15.

ρ was randomly selected at each solution generation among three values $[10, 28, 40]$. The input to the network was the system state enlarged by ρ, $\mathbf{i} = [x_k, y_k, z_k, \rho]$, evaluated at time t_k. The output was the corresponding solution enlarged by ρ at time t_{k+1}, $\mathbf{o} = [x_{k+1}, y_{k+1}, z_{k+1}, \rho]$. Therefore, the input and the output layers of the network featured 4×1 dimensions. Three fully connected hidden layers were inserted, their activation functions were set to be ReLU ones.

$$\mathbf{i} = \begin{bmatrix} x_k \\ y_k \\ z_k \\ \rho \end{bmatrix} \qquad \mathbf{o} = \begin{bmatrix} x_{k+1} \\ y_{k+1} \\ z_{k+1} \\ \rho \end{bmatrix} \tag{31}$$

The forecasting and extrapolation capabilities of the trained neural network were tested by feeding it with a random initial condition, assuming the ρ parameter of the system to be either 17 or 35. As expected, the neural network was not able to properly forecast the future state. A possible improvement might be obtained by feeding the neural network not only with the enlarged state at time t_k, but also with some of the previous ones. As a matter of fact, this condition highlights the inability of Neural Networks to work in the extrapolation regime. Namely, great results were obtained training, validating and testing the network on a single ρ value, this did not hold true in the multiple ρ configuration. The imminent lobe transition task was solved by feeding the neural network with a $4\mu \times 1$ vector. Where μ is the number of successive system state values that were fed to the network. In addition to the k-th time instant state, the network was also fed with the corresponding Boolean lobe tag, ρ was set to 28. The lobe tag was defined by bisecting the xyz data with the plane perpendicular to the xy plane and intersecting it along the second-fourth quadrants bisector $y = -x$. Given the $4\mu \times 1$ input vector, the network was trained to provide the lobe tag at τ time instants in the future. This configuration was tested for μ values of $[1, 2, 5, 10]$ and τ values of $[10, 50, 100]$. As a result, these tests highlighted that the higher τ the higher the required μ. Fig. 16 shows a prediction for $\mu = 2$ and $\tau = 50$.

$$\mathbf{i} = \begin{bmatrix} \mathbf{x}_k \\ lobe_k \\ \vdots \\ \mathbf{x}_{k+\mu} \\ lobe_{k+\mu} \end{bmatrix} \qquad \mathbf{o} = \begin{bmatrix} lobe_{k+\mu+\tau} \end{bmatrix} \tag{32}$$

D. Belousov-Zhabotisnky

The Belousov-Zhabotinsky data set was analysed by means of the Dynamic Mode Decomposition algorithm. As reported in Section II, the data matrix X was split in the X_1 and X_2 matrices, these were then processed so to extract the reduced order model of the system dynamics. Note that the initial dimensions of the data matrix were $351 \times 451 \times 1200$. The time evolution was then reconstructed by considering a variable number of modes. Results highlight the capability of the DMD

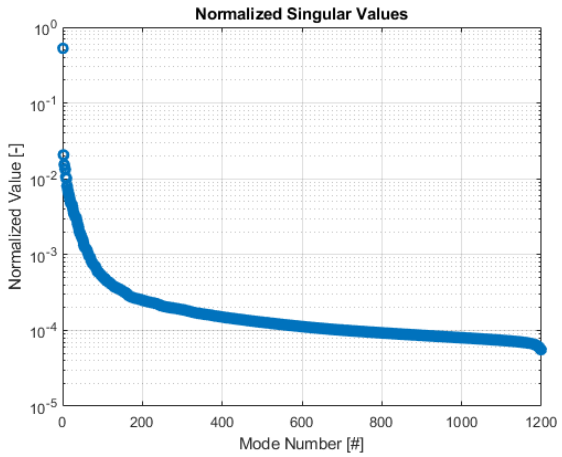

Fig. 17. Normalized Singular Values of the Chemical Oscillator data.

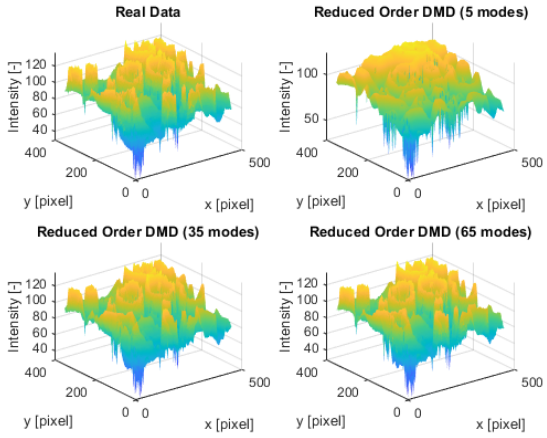

Fig. 19. Reconstruction of a single 3D frame of the Chemical Oscillator data matrix considering 5, 35 and 65 DMD modes.

algorithm to effectively guarantee a solution for the solution of the curse of dimensionality. In Fig. 17 the normalized values of the singular values of the X_1 matrix are plotted, these show the characteristic shape. The selected modes were the first $5, 35$ and 65 ones, the corresponding first frame 2D and 3D reconstructions are depicted in Fig. 18 and Fig. 19.

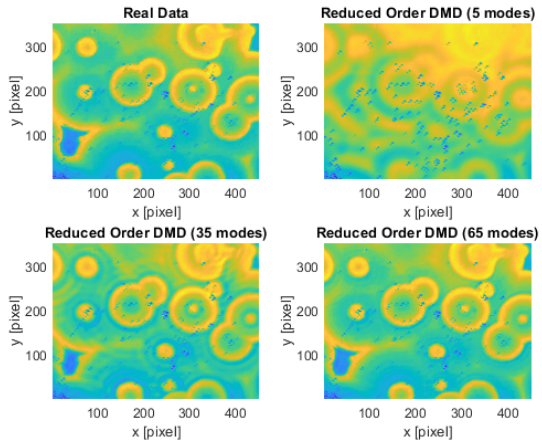

Fig. 18. Reconstruction of a single 2D frame of the Chemical Oscillator data matrix considering 5, 35 and 65 DMD modes.

IV. Summary and Conclusions

Data-driven methods for the analysis of data collections have been gaining large momentum lately. This document shows some of the results that were achieved while trying to apply these solutions to several different problems. Some limitations needed to be faced in some cases, mainly due to the low-data regime in which some tasks needed to be addressed. Furthermore, Neural Networks clearly showed their most important limitation: the inability to perform extrapolation on unseen data. This is the main reason for the failure of these powerful architectures in some applications. In conclusion the applied methods were the following: the Dynamic Mode Decomposition method, its Time Delay Embedding formulation, the Sparse Identification of Nonlinear Dynamics and a specific typology of Neural Networks, the Multi-Layer Perceptron. This document was written in order to serve as a reference for the development of further data analysis, some of the results testify the huge advantage that can be derived from these algorithms.

Acknowledgement

The author would like to thank Professor Nathan J. Kutz from the University of Washington in Seattle for the precious knowledge he shared during his enthralling lectures.

References

[1] Jonathan H. Tu, Clarence W. Rowley, Dirk M. Luchtenburg, Steven L. Brunton, J. Nathan Kutz. "On dynamic mode decomposition: theory and applications.", Journal of Computational Dynamics, 2014, 1 (2) : 391-421. doi: 10.3934/jcd.2014.1.391.

[2] Peter Schmid, Jörn Sesterhenn. (2008). "Dynamic Mode Decomposition of numerical and experimental data." Journal of Fluid Mechanics.

[3] Kamb, Mason, Eurika Kaiser, Steven L. Brunton and J. Nathan Kutz. "Time-delay observables for koopman: theory and applications." arXiv: Numerical Analysis (2018).

[4] Steven L. Brunton, J. Proctor, J. Nathan Kutz, (2016). "Sparse identification of nonlinear dynamics (SINDy)."

[5] Yan LeCun, Yoshua Bengio, Geoffrey Hinton. "Deep learning." Nature 521, 436–444 (2015).

[6] Michael A. Nielsen. "Neural networks and deep learning." Vol. 2018. San Francisco, CA: Determination press, 2015.

[7] M. Gavish and D. L. Donoho, "The Optimal Hard Threshold for Singular Values is $4/\sqrt{3}$." in IEEE Transactions on Information Theory, vol. 60, no. 8, pp. 5040-5053, Aug. 2014.