

RECONSTRUCTION OF ORDINARY DIFFERENTIAL EQUATIONS FROM IRREGULARLY DISTRIBUTED TIME-SERIES DATA

A.G. Golovkina^a, V.A. Kozynchenko, N.V. Kulabukhova

Saint Petersburg State University, 7/9 Universitetskaya emb., Saint Petersburg, 199034, Russia

E-mail: ^a a.golovkina@spbu.ru

The present paper aims to develop a reconstruction method for the right side of a system of ODEs in polynomial form from sparse and irregularly distributed time-series data. This method doesn't require any additional knowledge about the system and has several steps. The scarcity of the data through the trajectory length is compensated by the artificially generated points using approximating trigonometrical polynomials. Then, we get uniformly spread data points with the step conditioned by the desired accuracy of derivatives approximation in ODEs. This let to further use conventional reconstruction algorithms described in the literature. We test the proposed method on time series data generated from known ODE models in a two-dimensional system. We quantify the accuracy of the reconstruction for the system of ODEs as a function of the amount of data used by the method. Further, we solve the reconstructed system of ODEs and compare the solution to the original time series data. The method developed and validated here can now be applied to large data sets for physical and biological systems for which there is no known system of ODEs.

Keywords: System identification, ODEs reconstruction, dynamical systems

Anna Golovkina, Vladimir Kozynchenko, Natalia Kulabukhova

Copyright © 2021 for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

1. Introduction

The modeling and identification of dynamical systems from time-series data is a field of increasing interest mainly because of forecasting and model-based control applications. A mathematical description of interconnections between the measured data is a helpful tool to understand the considered dynamical system, predict its behavior or respond to different control actions.

System identification problems can be categorized into two groups: reconstruction of the mathematical form of the measured data (i.e. a solution of the system) or the reconstruction of the equations of motion of the underlying system. Irrelative to the type of problem, when the structure of the model is known the only necessary thing is to estimate the unknown coefficients from the time-series data either in general solution or in the right side of the differential equation. Otherwise, identification requires a prior step of choosing the most suitable mathematical form. The situation becomes more complicated when the measured data demonstrates nonnegligible nonlinear system dynamics. Unlike linear models for which a complete theory exists, nonlinear modeling still lacks well-established identification algorithms.

The previous work of the authors [1] introduces an algorithm of learning dynamical systems from time-series data including both of mentioned above methods as two sequential steps. At first, reconstruction of ordinary differential equations (ODEs) with a polynomial right side with only one measured system trajectory. Then, applying the Taylor mapping technique to represent the solution of ODEs. The solution is written in a polynomial form establishing a relationship between the phase variables at the current and past moments of time. In other words, a regression formula with predefined weights initialized from the reconstructed ODEs. Such a representation is convenient for further fine-tuning of the weights according to other available training data.

The present paper aims to strengthen the previous research [1] by enhancing a reconstruction algorithm of ODEs with polynomial right side when only rare or irregular measurements are available. Under ideal conditions, i.e. noise-free and high sampling frequency a variety of schemes, including sparse regression schemes [2-3], reservoir computing [4] and neural approaches [5-6]. However, real life data are often corrupted by noise and/or observed partially. In such situations, the above-mentioned approaches are most likely to fail to uncover unknown governing equations. To address this challenge, we need to jointly solve the reconstruction of governing equations and the identification of the hidden dynamics [1].

Here, we develop a method to build a polynomial right side of a system of ODEs that will correspond to the time-series data of a dynamical system. This method doesn't need any input except the time series data and includes several steps. We first identify a basis to approximate the sparse time series data. Here we use trigonometrical bases, but it can be chosen arbitrarily. The second step is an augmentation of a measured system trajectory by a linear segment to satisfy the periodicity condition at the ends of the time interval. Then, a choice both the order of trigonometrical polynomials and data points used for approximation. After the best approximation function consistent with the measured data is found, we use it further for data points generation through the trajectory length and solving a system of linear equations. We test our ODEs reconstruction method on time series data generated from known ODEs models in a two-dimensional system. We quantify the accuracy of the reconstruction for the system of ODEs as a function of the amount of data used by the method. Further, we solve the reconstructed system of ODEs and compare the solution to the original time series data. The method developed and validated here can now be applied to large data sets for physical and biological systems for which there is no known system of ODEs.

The rest of the paper is organized as follows. Sec. 2 introduce a step-by-step description of the proposed reconstruction algorithm, while sec. 3 contains the presentation of a test model with the known system of ODEs and the results of its equation reconstruction in polynomial form. Sec. 4 draws the conclusion remarks and discusses the obtained results.

2. Reconstruction algorithm

Let us denote the set of the parameters describing the process as vector X with changing in time components $X_j(t), j = \overline{1, n}$. And we suppose to know the values of the vector function $X(t)$ measured in M discrete times $t_0, \dots, t_{M+1}: X(t_0), \dots, X(t_{M+1})$.

The main our assumption about the collected time-series data describing the multi-parametric dynamical process is that it approximately follows an autonomous ODEs system. We will find its right-hand side in polynomial form, so that the system looks like

$$\frac{dX}{dt} = \sum_{k=0}^N P^k X^{[k]} \quad (1)$$

where t is an independent variable, $X \in \mathbb{R}^n$ is a state vector corresponding to the parameters of the dynamical process, and $X^{[k]}$ means k -th Kronecker's power of vector X . For example, for $X = (x_1, x_2)$ we have $X^{[2]} = (x_1^2, x_1x_2, x_2^2)$, $X^{[3]} = (x_1^3, x_1^2x_2, x_1x_2^2, x_2^3)$ after reduction of the same terms.

Matrices P^k are unknown and should be found from the measurements that we have $X(t_0), \dots, X(t_{M+1})$. If the available data is of high sampling frequency, we can easily compute P^k solving the system of liner equations, that comes in by replacing the derivatives $\frac{dX}{dt}$ in the left side of (1) with finite differences:

$$\frac{X(t_{i+1}) - X(t_{i-1})}{t_{i+1} - t_{i-1}} = \sum_{k=0}^N P^k X^{[k]}(t_i), \quad i = \overline{1, M}. \quad (2)$$

Equation (2) can be expressed in matrix form

$$AP = B, \quad (3)$$

where:

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

$$P = (P^1, \dots, P^N)^T,$$

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

The system (3) includes $n \cdot (n + n_2 + \dots + n_N)$ unknowns and $n \cdot M$ equations. Here n is the dimension of the vector X and $n_i, i \geq 2$ is the dimension of its Kronecker degree $X^{[i]}$. Thus, to obtain a system with an equal number of equations and unknowns, the following condition must be met for the number of measurements M

$$M = n + n_2 + \dots + n_N. \quad (4)$$

However, in case of sparse measurements in time, this approach can hardly be applied because system (3) may be underdetermined (condition (4) is violated) as well as a derivative's finite difference approximation can have insufficient precision.

To address this challenge, we propose at first to approximate the collected measurement by a set of trigonometrical polynomials. The choice is due to the uniform properties of this set of basis functions. For example, approximating a function by finite sum of Chebyshev polynomials has drawbacks related to the behavior of their derivatives [7]. The algorithm includes the following sequential steps.

1. Periodization of the input dataset. Choosing a closing coefficient (c_c).

Let us denote the lattice $T^0: \{t_i^0 \in [t_0, \dots, t_{M+1}], i = \overline{1, m_0}\}$ where the data $F^0: \{X(T_0)\}_{n \times m_0}$ is collected. In general case the collected data doesn't correspond to a periodic or oscillational dynamic process, meaning that $F^0(T_1^0) \neq F^0(T_{m_0}^0)$. To fulfill this condition necessary for trigonometric approximation, we introduce a close coefficient $c_c > 1$ that enlarge the given time interval till the value $t_c = t_0 + c_c(t_{M+1} - t_0)$. This complements the lattice T^0 with the following values $T^{11}: \{t_1^{11}, \dots, t_{m_{11}}^{11}\}$, where $t_1^{11} = T_{m_0}^0$ and $t_{m_{11}}^{11} = t_c$. Let $T^1 = T^0 \cup T^{11}$ and lattice function $F^1 = F^0 \cup F^{11}$, where $F^{11}: \{X(T^{11})\}_{n \times m_{11}} = \{a_j t_i^{11} + b_j\}_{i=\overline{1, m_{11}}, j=\overline{1, n}}$. Unknown coefficients are defined from the following conditions:

$$\begin{cases} F_{j1}^0 = a_j t_{m_{11}}^{11} + b_j \\ F_{jm_0}^0 = a_j t_1^{11} + b_j \end{cases} \quad (5)$$

Satisfying to (5) means that $F^1(T_1^1) = F^1(T_{m_1}^1)$ where $m_1 = m_0 + m_{11}$.

2. Determine coefficients of trigonometrical polynomials. Choosing an order of polynomials (K).

The order of trigonometrical polynomials K defines the number of lattice points $m_2 = 2K + 1$ which are used for the weight coefficients $\alpha_0^j, \alpha_k^j, \beta_k^j, j = \overline{1, n}, k = \overline{1, K}$ calculation in approximation formula

$$f_j(z) = \frac{1}{2} \alpha_0^j + \sum_{k=1}^K (\alpha_k^j \cos(\omega_k z) + \beta_k^j \sin(\omega_k z)), j = \overline{1, n} \quad (6)$$

where $\omega_k = \frac{2\pi k}{L}, L = t_c - t_0, z$ - a lattice point where the polynomial value is computed.

Let us choose m_2 points from the lattice T^1 being at an equal distance from each other and select corresponding lattice function values F^1 . Those points form lattice $T^2: \{t_1^2, \dots, t_{m_2}^2\}$ and lattice function $F^2: \{X(T^2)\}_{n \times m_2}$. According to [8], the unknown coefficients in (6) are found from the equality condition of trigonometrical polynomial values and lattice function in the T^2 mesh nodes $F_j^2 = f_j(T^2)$. It should be noted that we take an odd number of nodes. In this case, the formulas are

$$\alpha_0^j = \frac{2}{m_2} \sum_{i=1}^{m_2} F_{ji}^2, \quad \alpha_k^j = \frac{2}{m_2} \sum_{i=1}^{m_2} F_{ji}^2 \cos \frac{2\pi k i}{m_2}, \quad \beta_k^j = \frac{2}{m_2} \sum_{i=1}^{m_2} F_{ji}^2 \sin \frac{2\pi k i}{m_2}.$$

3. Generating a dense lattice for derivatives approximation. Choosing a stride (s_d).

Trigonometrical approximation (6) fitted in the lattice points T^2 is used further to generate a new lattice function $F^3: \{f(T^3)\}_{n \times m_3}$ with a mesh $T^3: \{t_1^3, \dots, t_{m_3}^3\}$ frequent enough to approximate derivatives (2) with central finite differences. Before doing that and solving a linear system (3) to obtain necessary matrices $P^k, k = \overline{1, N}$ in (1), we should delete the part of the lattice $T^{31} = \{T^3: t_i^3 > t_{M+1}, i = \overline{1, m_3}\}$ corresponding to the added linear closer in p.1. It results in a lattice $T^4 = T^3 \setminus T^{31}: \{t_1^4, \dots, t_{m_4}^4\}$ and lattice function $F^4: \{f(T^4)\}_{n \times m_4}$.

Condition (4) defines the necessary number of points we should take from F^4 to solve (3). For sake of convenience, let us introduce a stride parameter s_d to resample T^4 to M groups of 3 points (2 outermost to calculate the finite difference in the central). Then, $T^5: \{t_1^5, \dots, t_{m_5}^5\}, t_i^5 = t_1^4 + i \cdot s_d, i = \overline{1, m_5}$ and $F^5: \{f(T^5)\}_{n \times m_5}$ defines the matrix A and vector B in the linear system (3) that can be solved with a suitable numerical method.

The algorithm described above has three parameters introduced at each step that make an impact on the solution of (3). Thus, generally they should be chosen as a solution of optimization problem $c_c, K, s_d = \arg \min \|X - Y\|$, where Y is a numerical solution of (1) and X is the measurements that we have.

3. Test model and numerical results

As a training dataset for the introduced reconstruction algorithms let us consider points $(x(t), y(t))$ artificially generated by a model of two-dimensional particle motion in cylindrical deflector:

$$\begin{cases} \dot{x} = y, \\ \dot{y} = -2x + \frac{x^2}{R}. \end{cases} \quad (7)$$

We numerically find a particular solution of (7) at the time interval $[0, 4]$ with the initial condition $X_0: (x_0, y_0) = (-2, 4)$, parameter $R = 10$ and integration step $h = 0.01$. As a training set we consider every 20-th point in the generated trajectory, so T^0 includes 20 points ($m_0 = 20$). Figure 1,a demonstrates F^1 lattice function with the closed orbit used for trigonometrical approximation (step 1), figure 1,b (blue line) illustrates F^4 lattice function (step 2-3) used further for calculating unknown matrices $P^k, k = \overline{1, N}$.

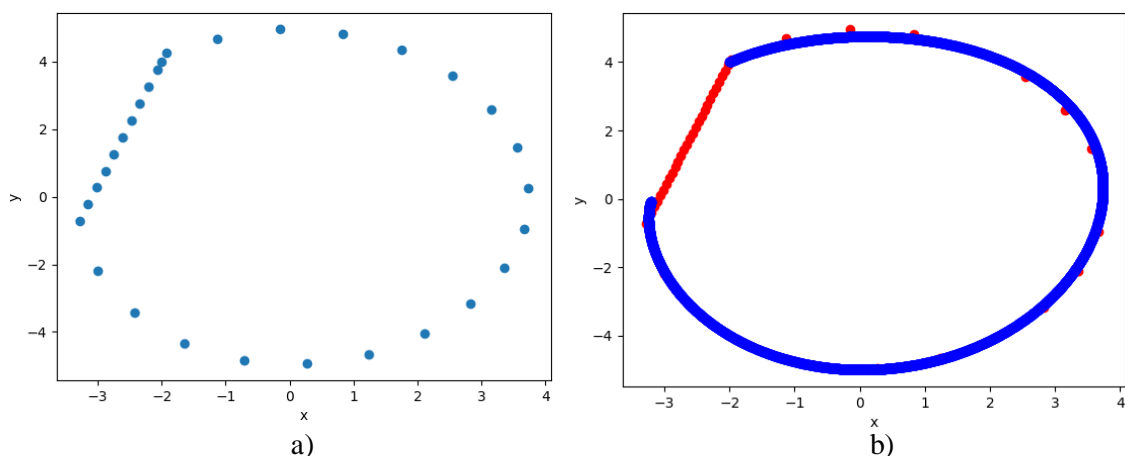


Figure 1. a) Training set F^1 and b) calculated lattice function F^4 with approximation (6)

The final reconstruction results depending on the amount of data used by the method are presented in figure 2 (a) $m_0 = 20$, b) $m_0 = 13$). The blue line corresponds to the true solution of (7) X , and the red line – to Y solution of (7) with the reconstructed matrices P^k . The title of each graphics contains the computed values c_c, K, s_d minimizing the norm of deviation X from Y .

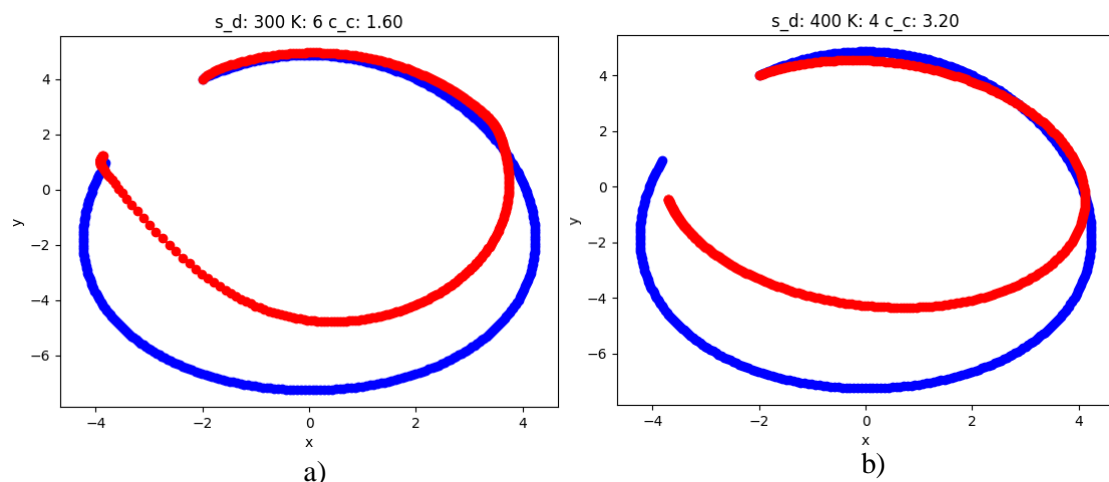


Figure 2. True (blue) and reconstructed (red) solutions of (7) when a) $m_0 = 20$, b) $m_0 = 13$

4. Conclusion

The paper introduces a reconstruction algorithm for the system of ODEs right side in polynomial form from irregularly distributed time-series data. The presented algorithm aims to amplify the results of the authors' previous work [1] built on the assumption that the collected measurements are frequently and uniformly spread in time. Although the solution of reconstructed ODEs doesn't fully congruent to the solution of true ODEs [fig. 2], it captures the dynamics of the system what is more important. According to [1, 9], the reconstructed ODEs are only used for initial weights initialization in the regression formula or neural network. After then, the weights are anyway fine-tuned with additional data, so we claim mild requirements for the accuracy of the ODEs reconstruction.

5. Acknowledgements

The authors would like to thank Saint Petersburg State University for the research grant ID: 75206008.

References

- [1] Golovkina, A. Kozynchenko, V. Kulabukhova, N. Reconstruction and Identification of Dynamical Systems Based on Taylor Maps // Gervasi O. et al. (eds) Computational Science and Its Applications – ICCSA 2021. ICCSA 2021. Lecture Notes in Computer Science, vol. 12956. Springer, 2021
- [2] Brunton, S. L. Proctor, J. L., Kutz, J. N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems // Proceedings of the National Academy of Sciences, vol. 113, No 15, pp. 3932–3937, 2016
- [3] Corbetta, M. Application of sparse identification of nonlinear dynamics for physics-informed learning, 2020 IEEE Aerospace Conference, pp. 1-8, 2020
- [4] Pathak, J. et al. Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach // Physical Review Letters, vol. 120, issue 2, 2018
- [5] Vlachas, P. R. et al. Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks // Proceedings of The Royal Society A, vol. 474, issue 2213, 2018
- [6] Fablet, R. Ouala, S. Herzet, C. Bilinear Residual Neural Network for the Identification and Forecasting of Geophysical Dynamics // 26th European Signal Processing Conference (EUSIPCO), pp. 1477-1481, 2018
- [7] Tal-Ezer, H. Nonperiodic Trigonometric Polynomial Approximation // Journal of Scientific Computing, vol. 60, pp. 345-362, 2014
- [8] Hesthaven, J. Gottlieb, S. Gottlieb, D. Spectral Methods for Time-Dependent Problems (Cambridge Monographs on Applied and Computational Mathematics). Cambridge: Cambridge University Press, 2007
- [9] Ivanov, A. Golovkina, A. Iben, U. Polynomial neural networks and Taylor maps for dynamical systems simulation and learning // Frontiers in Artificial Intelligence and Applications, vol. 325, pp. 1230-1237, 2020