

Метод последовательных приближений для построения модели динамической полиномиальной регрессии*

А. Г. Головкина, В. А. Козыниченко, И. С. Клименко

Санкт-Петербургский государственный университет, Российская Федерация,
199034, Санкт-Петербург, Университетская набережная, 7–9

Для цитирования: Головкина А. Г., Козыниченко В. А., Клименко И. С. Метод последовательных приближений для построения модели динамической полиномиальной регрессии // Вестник Санкт-Петербургского университета. Прикладная математика. Информатика. Процессы управления. 2022. Т. 18. Вып. 4. С. 00–00.
<https://doi.org/>

Прогнозирование поведения некоторого процесса во времени является важной задачей, возникающей во многих прикладных областях, при чем, информация о породившей процесс системе может как полностью отсутствовать, так и быть частично ограниченной. Единственное доступное знание — это накопленные данные о прошлых состояниях и параметрах процесса. Такая задача может успешно решаться с использованием методов машинного обучения, однако если речь идет о моделировании физических экспериментов или об областях, где к важным относится способность модели к обобщению и интерпретируемость прогнозов, то большинство методов машинного обучения не удовлетворяют указанным требованиям в полной мере. Проводится решение задачи прогнозирования с помощью построения модели динамической полиномиальной регрессии, и предлагается метод нахождения ее коэффициентов, опирающийся на связь с динамическими системами. Таким образом, построенная модель соответствует детерминированному процессу, потенциально описываемому дифференциальными уравнениями, а связь между ее параметрами может быть выражена в аналитическом виде. В качестве иллюстрации применимости предлагаемого подхода к решению задач прогнозирования был рассмотрен синтетический набор данных, сгенерированный как численное решение системы дифференциальных уравнений, которая описывает осциллятор Ван дер Поля.

Ключевые слова: полиномиальная регрессия, динамические системы, отображение Тейлора.

1. Введение. Регрессионные задачи возникают во многих областях, где требуется по известным значениям независимых переменных X построить выражение для зависимой переменной Y , т. е. найти отображение M такое, что

$$M : X\{x_1, x_2, \dots, x_n\} \rightarrow Y\{y_1, y_2, \dots, y_m\}.$$

Здесь Y — вектор целевых переменных, X — вектором независимых переменных, который называют вектор признаков.

В зависимости от вида отображения M различают регрессионные модели разного типа: линейные, полиномиальные, древовидные и др. Кроме того, если вектор признаков X и целевых переменных Y включают в себя совокупность значений в последовательные моменты времени, то модели можно классифицировать на авторегрессионные, когда строится задача построить отображение последовательности прошлых величин признака в последовательность его будущих значений, и модели динамической регрессии, когда целевые переменные в будущем определяются их значениями и вектора признаков в прошлом. Динамическая регрессия является мощным методом

*Работа выполнена при финансовой поддержке Санкт-Петербургского государственного университета (проект ID 93024916)

прогнозирования, так как она позволяет учитывать влияние совокупности переменных на прогноз. В настоящей работе рассматривается модель динамической регрессии следующего вида:

$$M: \mathbf{X}\{x_1, x_2, \dots, x_n\}_{t-1} \rightarrow \mathbf{X}\{x_1, x_2, \dots, x_n\}_t. \quad (1)$$

Среди способов построения регрессионных моделей вида (1) условно можно выделить четыре группы методов. К первой относятся построение линейных моделей регрессии [1], а также включение нелинейных (полиномиальных) преобразований входных данных в состав признаков [2]. Однако известно, что линейная регрессия не всегда дает удовлетворительный результат на данных, соответствующих сложному нелинейному взаимодействию, а включение большого количества нелинейных преобразований может привести к переобучению модели. В литературе предлагаются определенные подходы для оптимального выбора нелинейных членов [3, 4], а также методы факторизации [5, 6], которые позволяют снизить количество свободных коэффициентов в случае использования нелинейностей высокого порядка.

Вторая группа методов включает нелинейные модели машинного обучения, к числу которых относятся: метод опорных векторов, регрессия на основе гауссовского процесса, случайный лес, бустинговые алгоритмы и др. [7]. По своей природе указанные подходы представляют собой модели «черного ящика», изначально способные решать только интерполяционные задачи и не предоставляющие возможность в явном виде установить вид отображения M . Таким образом, усложняется оценка взаимосвязи между целевыми переменными и признаками, а следовательно, и интерпретация прогнозов модели. А это критически важно, когда речь идет об моделировании физических или технических параметров узлов производственных систем или о прогнозировании рисков развития заболеваний.

К третьей группе методов относятся нейронные сети, которые можно разделить на сети прямого распространения и рекуррентные. Использование сетей прямого распространения для решения регрессионных задач продиктовано теоремой об универсальной аппроксимации [8], утверждающей, что сеть с одним скрытым слоем, содержащим конечное число нейронов с гладкими функциями активации, может аппроксимировать любую непрерывную функцию на компактных подмножествах R^n . Рекуррентные сети имеют преимущество перед сетями прямого распространения в том, что они способны сохранять информацию о предыдущем состоянии модели [9], таким образом, реализуя в сети «память», позволяющую анализировать последовательности данных.

К сожалению, такой подход к решению регрессионных задач обладает схожими недостатками с классическими методами машинного обучения. Стоит отметить, что, вообще говоря, обобщающая способность построенных моделей для входных данных, не представленных в диапазоне обучающих данных, не исследована. Следовательно, традиционные и современные модели ML/NN подходят для прогнозирования и интерполяции динамики на новых входных данных, лежащих в диапазоне обучающей выборки, когда не важна интерпретируемость построенных прогнозов.

Четвертая группа методов опирается на фактическую эквивалентность регрессионной задачи (1) построению динамической модели процесса [9, 10]. Если при построении отображения M известна информация об основных физических соотношениях, лежащих в основе рассматриваемого процесса, выраженная в форме дифференциальных уравнений, то задача сводится к идентификации неизвестных параметров по имеющимся данным. Параметры обычно определяются путем минимизации среднеквадратичной ошибки между измеренными и модельными данными, для получения которых требуется использование численных методов интегрирования систем дифференциальных уравнений. В случае, если вид отображения M неизвестен, то реконструкция динамической системы выполняется по заранее выбранной системе функций (полиномы, тригонометрические функции, экспоненты и др.), относительно которой опять же решается задача идентификации неизвестных коэффициентов. Этот подход, например, реализован в библиотеке PySINDY с помощью методов разреженной регрессии [10, 11]. Однако в силу зашумленности и пропусков в реальных данных, восстановление системы дифференциальных уравнений может быть осуществлено недостаточно точно,

а донастройка параметров в процессе работы по шумным данным может приводить к осцилляциям параметров модели и снижению точности прогнозирования.

В настоящей работе представлен подход, опирающийся на комбинацию методов первого и четвертого типов, заключающийся в первоначальном восстановлении уравнений динамики в полиномиальной форме по историческим данным, которые затем приводятся к виду (1) с использованием метода матричных отображений Тейлора [12]. В этом случае вид отображения \mathcal{M} также будет полиномиальный, а коэффициенты при мономах однозначно определены из восстановленной системы дифференциальных уравнений. При необходимости коэффициенты полиномиальной регрессии могут быть донастроены на новых данных, при этом сохраняя связь с системой дифференциальных уравнений. Стоит отметить, что матрицы коэффициентов построенной таким образом полиномиальной регрессии являются сильно разреженными, потому можно рассматривать предварительное восстановление уравнений динамики в качестве L_0 регуляризации. Если говорить о непосредственном построении полиномиальной регрессии по данным, то неизбежно возникает большое количество свободных коэффициентов, которое может привести к переобучению модели в случае нелинейностей высокого порядка.

В п. 2 дана постановка задачи и приводится модель динамической полиномиальной регрессии, кратко описан алгоритм ее построения. Алгоритму реконструкции правых частей системы дифференциальных уравнений в полиномиальной форме по данным временного ряда посвящен п. 3. В п. 4 представлены метод последовательных приближений для нахождения общего решения восстановленной полиномиальной системы обыкновенных дифференциальных уравнений (ОДУ), также альтернативный способ, более удобный в плане численной реализации. В п. 5 содержатся результаты тестирования предложенного подхода на синтетических данных, полученных при численном решении системы дифференциальных уравнений Ван дер Поля.

2. Постановка задачи. Модель динамической полиномиальной регрессии.

Отображение \mathcal{M} , задающее модель полиномиальной регрессии (1), будем искать в следующем виде:

$$\mathbf{X}(t_{i+1}) = R^{11} \mathbf{X}(t_i) + R^{12} \mathbf{X}(t_i)^{[2]} + \dots + R^{1M} \mathbf{X}(t_i)^{[M]}, \quad (2)$$

где $\mathbf{X} \in R^n$, матрицы R_{1i} — регрессионные коэффициенты; $\mathbf{X}^{[i]}$ — i -я кронеровская степень вектора \mathbf{X} , полученная из тензорного произведения $\mathbf{X} \otimes \mathbf{X}^{[i-1]}$ удалением одинаковых с точностью до перестановки множителей элементов. Например, если $\mathbf{X} = (x_1, x_2)$, то $\mathbf{X}^{[2]} = (x_1^2, x_1 x_2, x_2^2)$ и $\mathbf{X}^{[3]} = (x_1^3, x_1^2 x_2, x_1 x_2^2, x_2^3)$. Отображение (2) линейно по матрицам R^{1i} и нелинейно относительно $\mathbf{X}(t_i)$.

Модель полиномиальной регрессии (2) в литературе носит имя отображений или моделей Тейлора, матричного преобразования Ли и др. [12], которое является аппроксимацией общего решения полиномиальной системы дифференциальных уравнений [13–15]. Это свойство позволяет использовать (2) в качестве дискретного представления динамической системы, не требующего применения численных методов решения дифференциальных уравнений [16] и удобного для современных методов обучения.

Таким образом, задача заключается в нахождении неизвестных матричных коэффициентов R^{1i} , которую будем решать в два последовательных этапа.

3. Реконструкция правой части системы дифференциальных уравнений. Обозначим набор параметров, описывающих динамический процесс, как вектор \mathbf{X} , содержащий компоненты $X_j(t)$, $j = \overline{1, n}$. Предположим, что известны значения вектор-функции $\mathbf{X}(t)$, измеренные для M дискретных значений t_0, \dots, t_{M+1} : $\mathbf{X}(t_0), \dots, \mathbf{X}(t_{M+1})$.

Основное предположение о собранных данных временных рядов, характеризующий многопараметрический динамический процесс, состоит в том, что этот процесс может быть приближенно описан системой ОДУ с полиномиальной правой частью

$$\frac{d\mathbf{X}}{dt} = \sum_{k=0}^N P^{1k} \mathbf{X}^{[k]} \quad (3)$$

где t – независимая переменная, $\mathbf{X} \in R^n$ – вектор состояния, соответствующий параметрам динамического процесса.

Матрицы P^{1k} обычно неизвестны, поэтому следует найти приближения для них из измерений $\mathbf{X}(t_0), \dots, \mathbf{X}(t_{M+1})$. Заменяем производные $\frac{d\mathbf{X}}{dt}$ в левой части системы (3) разностными производными:

$$\frac{\mathbf{X}(t_{i+1}) - \mathbf{X}(t_{i-1})}{t_{i+1} - t_{i-1}} = \sum_{k=0}^N P^{1k} \mathbf{X}^{[k]}(t_i), \quad i = \overline{1, M}. \quad (4)$$

Система уравнений (4) представляет собой систему линейных алгебраических уравнений, которую можно записать в матричной форме

$$AP = B, \quad (5)$$

где:

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

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

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

Система (5) содержит $n \cdot (n + n_2 + \dots + n_N)$ неизвестных и $n \cdot M$ уравнений. Здесь n – размерность вектора \mathbf{X} , а $n_i, i \geq 2$, – размерность кронекеровской степени $\mathbf{X}^{[i]}$. Отсюда следует, что для получения системы с равным числом уравнений и неизвестных необходимо выполнение следующего условия на число измерений $3 \cdot M$:

$$M = n + n_2 + \dots + n_N, \quad (6)$$

которое определяет связь между размерностью системы, связанной с количеством параметров процесса, и количеством измерений. При невыполнении условия (6) невозможно однозначно восстановить полиномиальную правую часть системы (3) для требуемого порядка нелинейности.

Стоит отметить, что точность восстановления системы зависит от и точности аппроксимации разностной производной, и от регулярности и частоты следования данных временного ряда [17]. В случае редких измерений с различной частотой потребуется прибегнуть к методам регуляризации входных данных [18, 19].

4. Идентификация общего решения системы ОДУ. Будем рассматривать систему ОДУ с полиномиальной правой частью (3), приближенно восстановленной с помощью алгоритмов, указанных в п. 3.

$$\frac{d\mathbf{X}}{dt} = \sum_{k=0}^N P^{1k}(t) \mathbf{X}^{[k]}$$

Для нахождения приближенного решения данной системы будем использовать итерационный алгоритм, основанный на методике, представленной в [12, 17]. Начиная с линейного члена в правой части (3), будем последовательно повышать порядок многочлена в правой части до требуемой степени нелинейности. На первом шаге рассмотрим линейное уравнение

$$\frac{d\mathbf{X}}{dt} = P^{11}(t) \mathbf{X}. \quad (7)$$

Его решение может быть найдено аналитически или численно. (например, для случая автономной системы $R^{11}(t, t_0) = \exp((t - t_0) P^{11})$) в виде

$$\mathbf{X}(t) = R^{11}(t, t_0) \mathbf{X}_0, \quad \mathbf{X}_0 = \mathbf{X}(t_0). \quad (8)$$

Далее рассмотрим укороченную систему с двумя слагаемыми в правой части

$$\frac{d\mathbf{X}}{dt} = P^{11}(t)\mathbf{X} + P^{12}(t)\mathbf{X}^{[2]}. \quad (9)$$

Для решения системы (9), подставим решение (8) линейной системы (7), найденное на предыдущем шаге, во второе слагаемое нелинейной системы (9):

$$\frac{d\mathbf{X}}{dt} = P^{11}(t)\mathbf{X} + P^{11}(t)R^{22}(t, t_0)\mathbf{X}_0^{[2]}, \quad R^{22}(t, t_0) = (R^{11}(t, t_0))^{[2]}. \quad (10)$$

Решение линейной неоднородной системы (10) может быть представлено в виде:

$$\mathbf{X}(t) = R^{11}(t, t_0)\mathbf{X}_0 + R^{12}(t, t_0)\mathbf{X}_0^{[2]}, \quad (11)$$

где

$$R^{12}(t, t_0) = \int_{t_0}^t R^{11}(t, \tau)P^{12}(\tau)R^{22}(\tau, t_0)d\tau.$$

Чтобы получить приближенное решение вида

$$\mathbf{X}(t) = R^{11}(t, t_0)\mathbf{X}_0 + R^{12}(t, t_0)\mathbf{X}_0^{[2]} + R^{13}(t, t_0)\mathbf{X}_0^{[3]},$$

подставим решение системы (11) в нелинейные члены укороченной системы

$$\frac{d\mathbf{X}}{dt} = P^{11}(t)\mathbf{X} + P^{12}(t)\mathbf{X}^{[2]} + P^{13}(t)\mathbf{X}^{[3]}.$$

После подстановки получим линейную систему

$$\begin{aligned} \frac{d\mathbf{X}}{dt} = & P^{11}(t)\mathbf{X} + P^{12}(t)\left(R^{11}(t, t_0)\mathbf{X}_0 + R^{12}(t, t_0)\mathbf{X}_0^{[2]}\right)^{[2]} + \\ & + P^{13}(t)\left(R^{11}(t, t_0)\mathbf{X}_0 + R^{12}(t, t_0)\mathbf{X}_0^{[2]}\right)^{[3]} = P^{11}(t)\mathbf{X} + P^{12}R^{22}(t, t_0)\mathbf{X}_0^{[2]} + \\ & + P^{12}(t)\left(\left(R^{11}(t, t_0)\mathbf{X}_0\right) \otimes \left(R^{12}(t, t_0)\mathbf{X}_0^{[2]}\right) + \right. \\ & \left. + \left(R^{12}(t, t_0)\mathbf{X}_0^{[2]}\right) \otimes \left(R^{11}(t, t_0)\mathbf{X}_0\right)\right) + P^{13}(t)R^{33}(t, t_0)\mathbf{X}_0^{[3]} + \dots, \end{aligned} \quad (12)$$

в которой

$$R^{33}(t) = (R^{11}(t))^{[3]}.$$

Введем обозначение

$$\left(R^{11}(t, t_0)\mathbf{Y}\right) \otimes \left(R^{12}(t, t_0)\mathbf{Y}^{[2]}\right) + \left(R^{12}(t, t_0)\mathbf{Y}^{[2]}\right) \otimes \left(R^{11}(t, t_0)\mathbf{Y}\right) = f(t, t_0, \mathbf{Y}) \quad (13)$$

Решим матричную систему линейных алгебраических уравнений с матрицей неизвестных R^{23} :

$$R^{23}M_Y = M_f \quad (14)$$

В матричной системе (14), $M_Y = (\mathbf{Y}_1^{[3]}, \dots, \mathbf{Y}_p^{[3]})$, $\mathbf{Y}_1, \dots, \mathbf{Y}_p$ – произвольная линейно независимая система векторов, $M_f = (f(t, t_0, \mathbf{Y}_1), \dots, f(t, t_0, \mathbf{Y}_p))$.

Решением системы (14) является матрица $R^{23} = R^{23}(t, t_0)$, которую также можно найти аналитически преобразованием левой части равенства (13). В результате находим, что

$$\begin{aligned} & \left(R^{11}(t, t_0)\mathbf{X}_0\right) \otimes \left(R^{12}(t, t_0)\mathbf{X}_0^{[2]}\right) + \\ & + \left(R^{12}(t, t_0)\mathbf{X}_0^{[2]}\right) \otimes \left(R^{11}(t, t_0)\mathbf{X}_0\right) = R^{23}(t, t_0)\mathbf{X}_0^{[3]}. \end{aligned} \quad (15)$$

Подставляя представление (15) в систему (12) и учитывая только члены до третьего порядка включительно, получим линейную систему

$$\frac{d\mathbf{X}}{dt} = P^{11}(t)\mathbf{X} + P^{12}(t)R^{22}(t, t_0)\mathbf{X}_0^{[2]} + (P^{12}(t)R^{23}(t, t_0) + P^{13}(t)R^{33}(t, t_0))\mathbf{X}_0^{[3]}. \quad (16)$$

Решение системы запишем следующим образом:

$$\mathbf{X}(t) = R^{11}(t, t_0)\mathbf{X}_0 + R^{12}(t, t_0)\mathbf{X}_0^{[2]} + R^{13}(t, t_0)\mathbf{X}_0^{[3]},$$

$$R^{13}(t) = \int_{t_0}^t R^{11}(t, \tau)P^{12}(\tau)R^{23}(\tau, t_0)d\tau + \int_{t_0}^t R^{11}(t, \tau)P^{13}(\tau)R^{33}(\tau, t_0)d\tau,$$

Продолжая итерации до порядка N , получим, что

$$\begin{aligned} \mathbf{X}(t) &= R^{11}(t, t_0)\mathbf{X}_0 + \sum_{k=2}^N R^{1k}(t)\mathbf{X}_0^{[k]}, \\ R^{1k}(t) &= \sum_{j=2}^k \int_{t_0}^t R^{11}(t, \tau)P^{1j}R^{jk}(\tau, t_0)d\tau, \quad R^{ii}(t, t_0) = (R^{11}(t, t_0))^{[i]}, \\ R^{11}(t, t_0) &= \exp((t - t_0)P^{11}). \end{aligned} \quad (17)$$

Матрицы $R^{jk}(t, t_0)$ рассчитываются по алгоритму, аналогичному алгоритму расчета матриц $R^{23}(t, t_0)$. Матрица $R^{jk}(t, t_0)$ также можно найти аналитическим путем, преобразовав левую часть равенства

$$\sum_{\sum_{i=1}^j k_i = k} \left(R^{1k_1}(t, t_0)\mathbf{X}_0^{[k_1]} \right) \otimes \dots \otimes \left(R^{1k_j}(t, t_0)\mathbf{X}_0^{[k_j]} \right) = R^{jk}(t, t_0)\mathbf{X}_0^{[k]}$$

Таким образом, определение матриц $R^{1i}(t, t_0)$ предполагает вычисление интегралов (17). В отдельных случаях эти интегралы могут быть эффективно рассчитаны с помощью символьных вычислений. В случаях, когда использование символьных вычислений неэффективно, для расчета интегралов можно применять численные методы интегрирования, что, как правило, требует больших вычислительных затрат. В качестве альтернативы вычислению интегралов можно рассмотреть следующий алгоритм расчета матриц $R^{ii}(t, t_0)$ [19].

Будем искать решение системы (3) в виде

$$\mathbf{X}(t) = \sum_{k=1}^M R^{1k}(t)\mathbf{X}_0^{[k]}. \quad (18)$$

Подставим представление (18) в обе части нелинейной системы (3):

$$\frac{d}{dt} \left(\sum_{k=1}^M R^{1k}(t)\mathbf{X}_0^{[k]} \right) = \sum_{i=1}^N P^{1i}(t) \left(\sum_{k=1}^M R^{1k}(t)\mathbf{X}_0^{[k]} \right)^{[i]}. \quad (19)$$

Возведя в правой части равенства (19) выражения в скобках в соответствующие кронекеровские степени, приведя подобные и приравнявая выражения при одинаковых кронекеровских степенях $\mathbf{X}_0^{[k]}$, получим совокупность систем дифференциальных уравнений относительно неизвестных матриц $R^{1i}(t, t_0)$:

$$\begin{aligned}
\frac{dR^{11}(t)}{dt} &= P^{11}(t) R^{11}(t), \\
\frac{dR^{12}(t)}{dt} &= P^{11}(t) R^{12}(t) + P^{12}(t) (R^{11}(t))^{[2]}, \\
\frac{dR^{13}(t)}{dt} &= P^{11}(t) R^{13}(t) + P^{12}(t) R^{23}(t) + P^{13}(t) (R^{11}(t))^{[3]}, \\
&\dots \\
\frac{dR^{1k}(t)}{dt} &= P^{11}(t) R^{1k}(t) + \sum_{j=2}^k R^{11}(t, t_0) P^{1j}(t) R^{jk}(t, t_0), \dots
\end{aligned} \tag{20}$$

Здесь $R^{jk}(t, t_0)$ вычисляется по формулам (17). Решение этих систем можно представить с использованием формулы Коши в виде

$$\begin{aligned}
R^{12}(t, t_0) &= \int_{t_0}^t R^{11}(t, \tau) P^{12}(\tau) R^{22}(\tau, t_0) d\tau, \\
R^{13}(t) &= \int_{t_0}^t R^{11}(t, \tau) P^{12}(\tau) R^{23}(\tau, t_0) d\tau + \int_{t_0}^t R^{11}(t, \tau) P^{13}(\tau) R^{33}(\tau, t_0) d\tau, \\
&\dots \\
R^{1k}(t) &= \sum_{j=2}^k \int_{t_0}^t R^{11}(t, \tau) P^{1j}(\tau) R^{jk}(\tau, t_0) d\tau, \dots
\end{aligned} \tag{21}$$

Формулы (21) эквивалентны ранее полученным формулам (17) для матриц $R^{1i}(t, t_0)$. Однако, система (20) с треугольной матрицей может быть решена с использованием численных методов решения систем ОДУ, что в ряде случаев представляет меньшую вычислительную сложность, чем последовательный расчет интегралов (21).

5. Численные результаты. В качестве примера, иллюстрирующего описанный ранее подход для построения модели динамической полиномиальной регрессии, рассмотрим систему ОДУ Ван дер Поля

$$\begin{aligned}
\dot{x} &= y, \\
\dot{y} &= \mu(1 - x^2)y - x.
\end{aligned} \tag{22}$$

Осциллятор Ван дер Поля описывает релаксационные колебания, встречающиеся в физике, биологии, сейсмологии и других областях [21], поэтому он часто применяется для моделирования колебательных процессов или генерации дополнительных тренировочных данных при обучении нейронных сетей [22].

Сгенерируем синтетический набор тренировочных данных с помощью численного решения системы (22) с начальным условием $(-2, 4)$ и параметром $\mu = 1$. Для численного интегрирования использовался решатель LSODA с автоматическим выбором шага [23]. Набор тренировочных данных был сформирован прореживанием результатов численного решения и составил 9 точек (показаны на фазовой плоскости рис. 1 точками на кривой 1).

Приведем сгенерированный набор тренировочных данных, использованный для восстановления системы:

$$\begin{array}{cccccccc}
x : & -2.00 & 2.29 & 0.72 & -1.88 & 0.64 & 1.44 & -1.94 & -0.71 & 1.88 \\
y : & 4.00 & -0.47 & -1.43 & 0.50 & 2.64 & -0.83 & -0.65 & 1.43 & -0.50
\end{array}$$

Кроме генерации данных, информация о правой части уравнений (22) больше не используется.

Тренировочные данные используются далее для реконструкции автономной системы ОДУ с полиномиальной правой частью в соответствии с алгоритмом, изложенным в п. 3. Правая часть была восстановлена до третьего порядка нелинейности (23) и найдены матрицы P^{11}, P^{12}, P^{13} .

$$\frac{d\mathbf{X}}{dt} = P^{11}\mathbf{X} + P^{12}\mathbf{X}^{[2]} + P^{13}\mathbf{X}^{[3]} \tag{23}$$

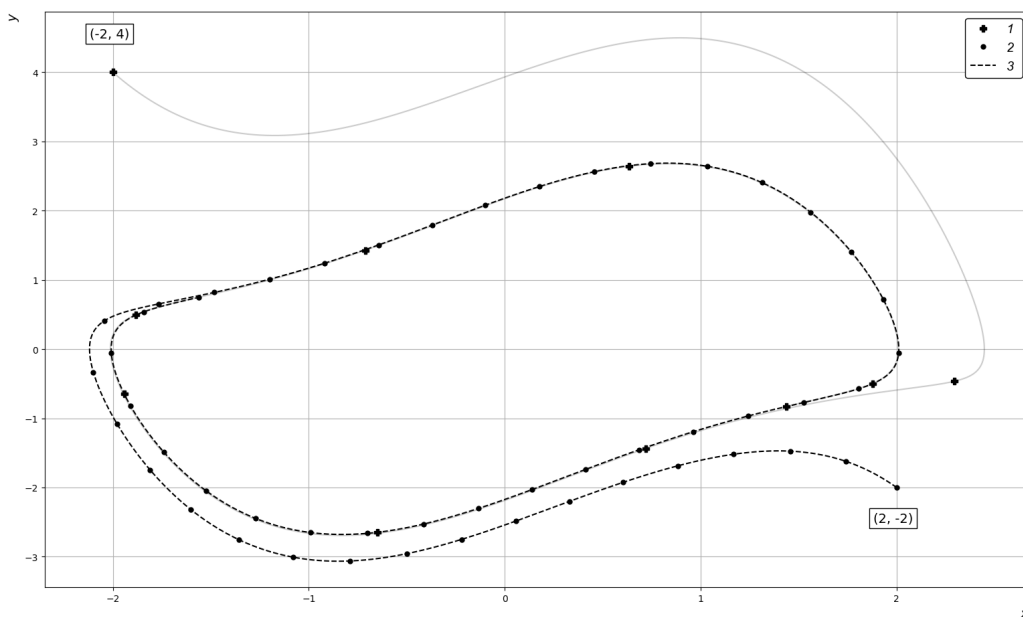


Рис. 1. Тренировочные данные, сгенерированные при $(x_0, y_0) = (-2, 4)$ (1), тестовые данные, сгенерированные при $(x_0, y_0) = (2, -2)$ (2) и спрогнозированные на тестовых данных значения с помощью построенной модели динамической полиномиальной регрессии (3)

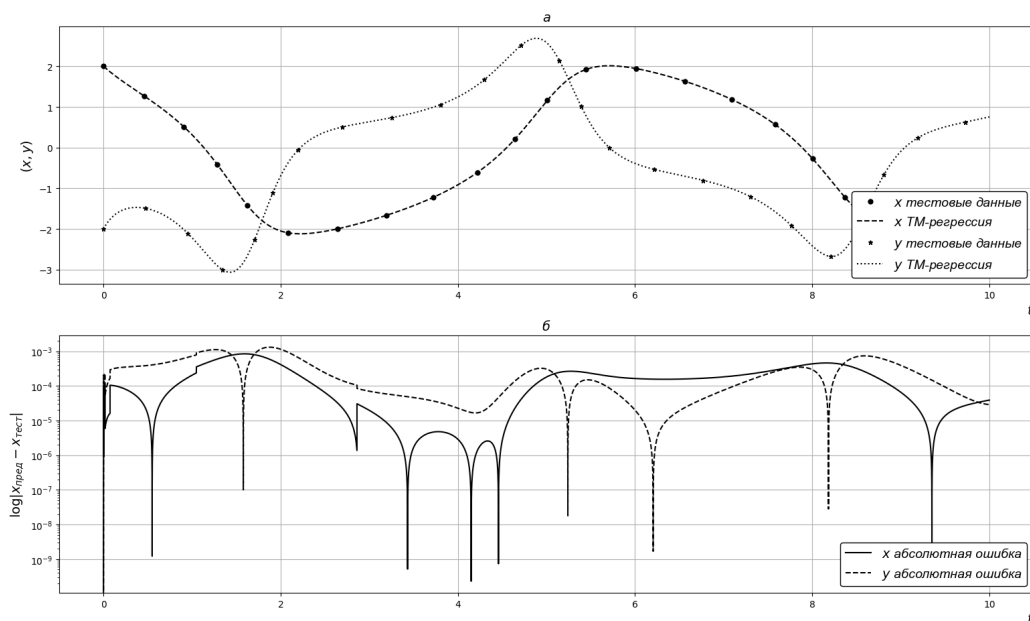


Рис. 2. Сравнение результатов многошагового прогнозирования с помощью построенной модели динамической регрессии четвертого порядка (шаг по времени $5 \cdot 10^{-5}$) на тестовых данных. $a - (x(t), y(t))_{\text{пред}}$ и $(x(t), y(t))_{\text{тест}}$; $b - \log |x_{\text{пред}} - x_{\text{тест}}|$

$$P^{11} = \begin{pmatrix} 4.099 \cdot 10^{-5} & 0.999 \\ -1.000 & 1.001 \end{pmatrix},$$

$$P^{12} = \begin{pmatrix} -3.469 \cdot 10^{-6} & -1.453 \cdot 10^{-5} & -6.111 \cdot 10^{-6} \\ -1.819 \cdot 10^{-5} & -7.761 \cdot 10^{-5} & -3.596 \cdot 10^{-5} \end{pmatrix},$$

$$P^{13} = \begin{pmatrix} -2.904 \cdot 10^{-7} & 2.337 \cdot 10^{-5} & -3.647 \cdot 10^{-5} & 1.976 \cdot 10^{-7} \\ -1.071 \cdot 10^{-5} & -1.000 & 1.696 \cdot 10^{-4} & -1.317 \cdot 10^{-4} \end{pmatrix}.$$

Далее приближенно восстановленная система (23) применяется для прямого вычисления матриц коэффициентов динамической полиномиальной регрессии до требуемого порядка точности с помощью численного решения системы уравнений (20) на промежутке от $t_0 = 0$ до Δt . В силу автономности (23) найденные матричные коэффициенты последовательно используются для определения будущих состояний системы в соответствии с формулой (2).

Например, матрицы коэффициентов $R^{11}(\Delta t)$, $R^{12}(\Delta t)$, $R^{13}(\Delta t)$, $R^{14}(\Delta t)$ до четвертого порядка нелинейности при $\Delta t = 5 \cdot 10^{-5}$ имеют вид:

$$R^{11} = \begin{pmatrix} 1.00 & 5.00 \cdot 10^{-5} \\ -5.00 \cdot 10^{-5} & 1.00 \end{pmatrix},$$

$$R^{12} = \begin{pmatrix} -1.74 \cdot 10^{-10} & -7.27 \cdot 10^{-10} & -3.06 \cdot 10^{-10} \\ -9.09 \cdot 10^{-10} & -3.88 \cdot 10^{-9} & -1.80 \cdot 10^{-9} \end{pmatrix},$$

$$R^{13} = \begin{pmatrix} -1.45 \cdot 10^{-11} & -8.16 \cdot 10^{-11} & -1.82 \cdot 10^{-9} & 9.67 \cdot 10^{-12} \\ 7.15 \cdot 10^{-10} & -5.00 \cdot 10^{-5} & 5.98 \cdot 10^{-9} & -6.59 \cdot 10^{-9} \end{pmatrix},$$

$$R^{14} = \begin{pmatrix} -2.54 \cdot 10^{-19} & 1.82 \cdot 10^{-14} & 1.53 \cdot 10^{-14} & 4.76 \cdot 10^{-18} & 2.26 \cdot 10^{-18} \\ 2.27 \cdot 10^{-14} & 2.03 \cdot 10^{-13} & 1.71 \cdot 10^{-13} & 1.53 \cdot 10^{-14} & 2.88 \cdot 10^{-17} \end{pmatrix}.$$

Благодаря полученной таким образом модели динамической полиномиальной регрессии решена задача многошагового прогнозирования для тестовых начальных условий $(x, y) = (-2, 2)$. На рис. 1 представлен фазовый портрет восстановленной системы (23) при указанных начальных данных, построенный путем многократного применения формулы (2), и фазовый портрет исходной системы (22), полученный с использованием решателя дифференциальных уравнений LSODA. Рис. 2 дополнительно иллюстрирует приведенные результаты в развертке по времени, а также показывает изменение абсолютного отклонения спрогнозированной траектории от фактической. Как можно видеть, абсолютная ошибка не превышает 10^{-3} .

Рисунки 1 и 2 демонстрируют, что модель динамической полиномиальной регрессии с матрицами коэффициентов, найденными в соответствии с рассмотренным алгоритмом, могут давать хорошие результаты при многошаговом итеративном прогнозировании для начальных значений, далеких от представленных в тренировочном наборе данных, который в описанном примере состоял всего из 9 точек. Кроме того, регрессионные коэффициенты можно дополнительно настроить по новым данным с помощью стандартных алгоритмов оптимизации. При этом для сокращения свободных коэффициентов в матрицах при дообучении модели может применяться информация о разреженности, естественным образом полученная после применения предложенного алгоритма.

6. Заключение. В последнее время методы глубокого обучения приобрели большую популярность из-за их широкого использования во многих областях. Рассмотрено применение глубоких нейронных сетей для решения обратных задач, связанных с реконструкцией и идентификацией динамических систем. Однако стандартные алгоритмы машинного обучения вряд ли могут дать физическую интерпретацию изучаемого динамического процесса. Более того, обычно для обучения таких моделей требуются большие наборы данных. В настоящей работе предлагается метод построения модели динамической полиномиальной регрессии, которая естественным образом предлагает интерпретацию результатов прогнозирования. Для сокращения количества свободных коэффициентов, возникающих при построении моделей высоких порядков нелинейности, обучение желательно проводить в два этапа. Динамическая регрессия в полиномиальной форме тесно связана с автономными дифференциальными

уравнениями с полиномиальной правой частью, которые часто встречаются на практике для представления моделей биологических, физических, химических систем. В связи с этим, на первом этапе следует восстановить систему дифференциальных уравнений по имеющимся данным, далее с помощью метода отображений Тейлора однозначно найти матричные коэффициенты общего решения в полиномиальной форме. Такой подход требует значительно меньше данных для обучения и приводит к разреженным матрицам коэффициентов, которые могут быть эффективно дообучены на новых данных. Описанный подход проиллюстрирован в задаче многошагового прогнозирования для осциллятора Ван дер Поля. Для обучения модели использовался набор тренировочных данных, состоящий всего лишь из 9 точек, при этом максимальное абсолютное отклонение при многошаговом прогнозировании от фактических данных составило 10^{-3} . Полученные результаты можно улучшить за счет тонкой настройки построенных матричных коэффициентов по дополнительным данным с помощью стандартных алгоритмов оптимизации.

Литература

1. *Jansson M., Wahlberg B.* A linear regression approach to state-space subspace system identification // *Signal Processing*. 1996. Vol. N 2. P. 103–129. [https://doi.org/10.1016/0165-1684\(96\)00048-5](https://doi.org/10.1016/0165-1684(96)00048-5).
2. *Herceg S., Ujevic Z., Bolf A. N.* Development of soft sensors for isomerization process based on support vector machine regression and dynamic polynomial models // *Chemical Engineering Research and Design*. 2019. Vol. 149. P. 95–103. <https://doi.org/10.1016/j.cherd.2019.06.034>.
3. *Dette H.* Optimal designs for identifying the degree of a polynomial regression // *Annals of Statistics*. 1995. Vol. 23. N 4. P. 1248–1266. <https://doi.org/10.1214/aos/1176324708>
4. *Kim B., Ko Ch.-Y., Wong N.* Tensor Network Subspace Identification of Polynomial State Space Models // *Automatica*. 2018. Vol. 95. P. 187–196. <https://doi.org/10.1016/j.automatica.2018.05.015>
5. *Blondel M., Ishihata M., Fujino A., Ueda N.* Polynomial networks and factorization machines: New insights and efficient training algorithms // *Proceedings of ICML 2016*. New York City, 2016. P. 850–858.
6. *Blondel M., Niculae V., Otsuka T., Ueda N.* Multi-output polynomial networks and factorization machines // *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. Long Beach, 2017. P. 3351–3361.
7. *Chen T., Guestrin C.* XGBoost: A Scalable tree boosting system // *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*. San Francisco, 2016. P. 785–794. <https://doi.org/10.1145/2939672.2939785>
8. *Hornik K., Stinchcombe M., White H.* Multilayer feedforward networks are universal approximators // *Neural Networks*. 1989. Vol. 2. P. 359–366.
9. *Rao S, Sethuraman S., Ramamurthi V.* A recurrent neural network for nonlinear time series prediction: a comparative study // *IEEE 1992 Workshop on Neural Networks for Signal Processing (NNSP'92)*. Helsingoer, 1992. P. 531–539.
10. *Kaheman K., Kutz J. N., Brunton S. L.* SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics // *Proceedings of the Royal Society A*. 2020. Vol. 476. N. 2242. Art. N. 20200279.
11. *Brunton S. L., Joshua L. P., Kutz N.* Discovering governing equations from data by sparse identification of nonlinear dynamical systems // *Proceedings of the National Academy of Sciences*. 2016. Vol. 113. № 15. P. 3932–3937.
12. *Andrianov S. N.* Dynamical modeling of control systems for particle beams. St. Petersburg: Publishing house of Saint Petersburg State University. 2004. 368 p.
13. *Dragt A.* Lie methods for nonlinear dynamics with applications to accelerator physics. 2011. URL: inspirehep.net/record/955313/files/T0C28Nov2011.pdf (дата обращения: 10.08.2022).
14. *Andrianov S.* Symbolic computation of approximate symmetries for ordinary differential equations // *Mathematics and Computers in Simulation*. 2001. Vol. 57. N. 3–5. P. 147–154
15. *Andrianov S.* A matrix representation of the Lie transformation // *Proceedings of the Abstracts of the International Congress on Computer Systems and Applied Mathematics*. St. Petersburg, 1993. Vol. 14. P. 19–23.
16. *Ivanov A., Golovkina A., Iben U.* Polynomial neural networks and taylor maps for dynamical systems simulation and learning // *24th European Conference on Artificial*

Intelligence, including 10th Conference on Prestigious Applications of Artificial Intelligence, PAIS 2020 Proceedings. IOS Press, 2020. P. 1230–1237. (Frontiers in Artificial Intelligence and Applications). <https://doi.org/10.3233/FAIA200223>

17. *Golovkina A., Kozynchenko V., Kulabukhova N.* Reconstruction and identification of dynamical systems based on Taylor maps // Computational Science and Its Applications — 21st International Conference, Proceedings. Pt VIII. Switzerland: Springer Nature. 2021. P. 360–369. (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)). https://doi.org/10.1007/978-3-030-87010-2_26

18. *Golovkina A. G., Kozynchenko V. A., Kulabukhova N. V.* Reconstruction of ordinary differential equations from irregularly distributed time-series data // Proceedings of the 9th International Conference “Distributed Computing and Grid Technologies in Science and Education” (GRID’2021). Dubna, Russia, July 5–9, 2021. Vol. 3041. P. 342–347.

19. *Головкина А. Г., Ганаева Д. Д.* Метод реконструкции нелинейных динамических систем по временным рядам // Процессы управления и устойчивость. 2022. Т. 9. № 1. С. 197–201.

20. *Клименко И. С.* Реализация метода матричных отображений для решения системы дифференциальных уравнений // Процессы управления и устойчивость. 2022. Т. 9. № 1. С. 53–57.

21. *Cartwright M. L.* Van der Pol’s equation for relaxation oscillations // Contributions to the Theory of Nonlinear Oscillations II. Princeton Ann. Math. Stud. Princeton: Princeton University Press, 1952. P. 3–18.

22. *Abrevaya G., Rish I., Aravkin A. Y., Cecchi G., Kozloski J., Polosecki P., Zheng P., Dawson S. P., Rhee J., Cox D.* Learning nonlinear brain dynamics: van der Pol Meets LSTM // bioRxiv 330548. <https://doi.org/10.1101/330548>

23. LSODA. Ordinary differential equation solver for stiff or non-stiff system. (September 2005). URL: <http://www.nea.fr/abs/html/uscd1227.html> (дата обращения: 10.08.2022).

Статья поступила в редакцию 11 августа 2022 г.

Статья принята к печати 01 сентября 2022 г.

Контактная информация:

Головкина Анна Геннадьевна — канд. физ.-мат. наук, доцент; a.golovkina@spbu.ru

Козынченко Владимир Александрович — канд. физ.-мат. наук, доцент; v.kozynchenko@spbu.ru

Клименко Илья Сергеевич — магистрант; st062546@student.spbu.ru

The method of successive approximations for constructing a model of dynamic polynomial regression*

A. G. Golovkina, V. A. Kozynchenko, I. S. Klimenko

St Petersburg State University, 7–9, Universitetskaya nab., St. Petersburg, 199034, Russian Federation

For citation: Golovkina A. G., Kozynchenko V. A., Klimenko I. S. The method of successive approximations for constructing a model of dynamic polynomial regression // Vestnik of Saint Petersburg University. Applied Mathematics. Computer Science. Control Processes, 2022, vol. 18, iss. 4, pp. 00–00. <https://doi.org/> (In Russian)

Predicting the behavior of a certain process in time is an important task that arises in many applied areas, and information about the system that generated this process can either be completely absent or be partially limited. The only available knowledge is the accumulated data on past states and process parameters. Such a task can be successfully solved using machine learning methods, but when it comes to modeling physical experiments or areas where the ability of a model to generalize and interpretability of predictions are important, then the most machine learning methods do not fully satisfy these requirements. The forecasting problem is solved by building a dynamic polynomial regression model, and a method for finding its coefficients is proposed, based on the connection with dynamic systems. Thus, the constructed model corresponds to a deterministic process, potentially described by differential equations, and the relationship between its parameters can be expressed in an analytical form. As an illustration of the applicability of the proposed approach to solving forecasting problems, we consider a synthetic data set generated as a numerical solution of a system of differential equations that describes the Van der Pol oscillator.

Keywords: polynomial regression, dynamic systems, Taylor map.

References

1. Jansson M., Wahlberg B. A linear regression approach to state-space subspace system identification. *Signal Processing*, 1996, iss. 2, pp. 103–129. [https://doi.org/10.1016/0165-1684\(96\)00048-5](https://doi.org/10.1016/0165-1684(96)00048-5).
2. Herceg S., Ujevic Z., Bolf A. N. Development of soft sensors for isomerization process based on support vector machine regression and dynamic polynomial models. *Chemical Engineering Research and Design*, 2019, vol. 149, pp. 95–103. <https://doi.org/10.1016/j.cherd.2019.06.034>.
3. Dette H. Optimal designs for identifying the degree of a polynomial regression. *Annals of Statistics*, 1995, vol. 23, iss. 4, pp. 1248–1266. <https://doi.org/10.1214/aos/1176324708>
4. Kim B., Ko Ch.-Y., Wong N. Tensor Network Subspace Identification of Polynomial State Space Models. *Automatica*, 2018, vol. 95, pp. 187–196. <https://doi.org/10.1016/j.automatica.2018.05.015>
5. Blondel M., Ishihata M., Fujino A., Ueda N. Polynomial networks and factorization machines: New insights and efficient training algorithms. *Proceedings of ICML 2016*. New York City, 2016. P. 850–858.
6. Blondel M., Niculae V., Otsuka T., Ueda N. Multi-output polynomial networks and factorization machines. *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. Long Beach, 2017. P. 3351–3361.
7. Chen T., Guestrin C. XGBoost: A Scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*. San Francisco, 2016. P. 785–794. <https://doi.org/10.1145/2939672.2939785>

*This work was financially supported by Saint Petersburg State University (project ID 93024916).

8. Hornik K., Stinchcombe M., White H. Multilayer feedforward networks are universal approximators. *Neural Networks*, 1989, vol. 2, pp. 359–366.
9. Rao S, Sethuraman S., Ramamurthi V. A recurrent neural network for nonlinear time series prediction: a comparative study. *IEEE 1992 Workshop on Neural Networks for Signal Processing (NNSP'92)*. Helsingoer, 1992. P. 531–539.
10. Kaheman K., Kutz J. N., Brunton S. L. SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics. *Proceedings of the Royal Society A*. 2020. Vol. 476. N. 2242. Art. N. 20200279.
11. Brunton S. L., Joshua L. P., Kutz N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*. 2016, vol. 113, iss. 15, pp. 3932–3937.
12. Andrianov S. N. Dynamical modeling of control systems for particle beams. St. Petersburg: Publishing house of Saint Petersburg State University. 2004. 368 p.
13. Dragt A. Lie methods for nonlinear dynamics with applications to accelerator physics. 2011. URL: inspirehep.net/record/955313/files/TOC28Nov2011.pdf (date of access: 10.08.2022).
14. Andrianov S. Symbolic computation of approximate symmetries for ordinary differential equations. *Mathematics and Computers in Simulation*, 2001, vol. 57, iss. 3–5, iss. 147–154
15. Andrianov S. A matrix representation of the Lie transformation. *Proceedings of the Abstracts of the International Congress on Computer Systems and Applied Mathematics*. St. Petersburg, 1993, vol. 14, pp. 19–23.
16. Ivanov A., Golovkina A., Iben U. Polynomial neural networks and taylor maps for dynamical systems simulation and learning. *24th European Conference on Artificial Intelligence, including 10th Conference on Prestigious Applications of Artificial Intelligence, PAIS 2020 Proceedings*. IOS Press, 2020. P. 1230–1237. (Frontiers in Artificial Intelligence and Applications). <https://doi.org/10.3233/FAIA200223>
17. Golovkina A., Kozynchenko V., Kulabukhova N. Reconstruction and identification of dynamical systems based on taylor maps. *Computational Science and Its Applications – 21st International Conference, Proceedings*. Pt VIII. Switzerland: Springer Nature. 2021. P. 360–369. (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)). https://doi.org/10.1007/978-3-030-87010-2_26
18. Golovkina A. G., Kozynchenko V. A., Kulabukhova N. V. Reconstruction of ordinary differential equations from irregularly distributed time-series data. *Proceedings of the 9th International Conference “Distributed Computing and Grid Technologies in Science and Education” (GRID'2021)*. Dubna, Russia, July 5-9, 2021. Vol. 3041. P. 342–347.
19. Golovkina A. G., Ganaeva D. D. Metod rekonstruktzii nelineinykh dinamicheskikh sistem po vremennym ryadam [Method of nonlinear dynamical systems reconstruction based on time series data]. *Control processes and Stability*, 2022, vol. 9, iss. 1, pp. 197–201. (In Russian)
20. Klimenko I. S. Realizatsia metoda matrichnykh otobrajenii dlya reshenia sistemy differentsialnykh uravnenii [Implementation of matrix map method for solving a system of ordinary equations]. *Control processes and Stability*, 2022, vol. 9, iss. 1, pp. 53–57.
21. Cartwright M. L. Van der Pol's equation for relaxation oscillations. *Contributions to the Theory of Nonlinear Oscillations II*. Princeton Ann. Math. Stud. Princeton: Princeton University Press, 1952. P. 3–18.
22. Abrevaya G., Rish I., Aravkin A. Y., Cecchi G., Kozloski J., Polosecki P., Zheng P., Dawson S. P., Rhee J., Cox D. Learning nonlinear brain dynamics: van der Pol Meets LSTM. *bioRxiv 330548*. <https://doi.org/10.1101/330548>
23. LSODA. Ordinary differential equation solver for stiff or non-stiff system. (September 2005). URL: <http://www.nea.fr/abs/html/uscd1227.html> (date of access: 10.08.2022).

Received: August 11, 2022
Accepted: September 01, 2022

A u t h o r s ' i n f o r m a t i o n :

Anna G. Golovkina — PhD in Physics and Mathematics, Associate Professor; a.golovkina@spbu.ru

Vladimir A. Kozynchenko — PhD in Physics and Mathematics, Associate Professor;
v.kozynchenko@spbu.ru

Ilya S. Klimenko — Master Student; st062546@student.spbu.ru