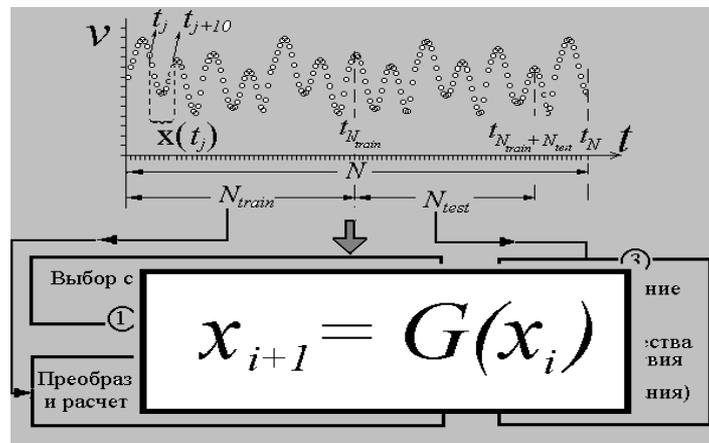


Б.П. БЕЗРУЧКО, Д.А. СМИРНОВ

ПОСТРОЕНИЕ МОДЕЛЬНЫХ ОТОБРАЖЕНИЙ ПО ХАОТИЧЕСКИМ ВРЕМЕННЫМ РЯДАМ



Саратов 2000

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет нелинейных процессов

Кафедра электроники, колебаний и волн

**САРАТОВСКОЕ ОТДЕЛЕНИЕ ИНСТИТУТА
РАДИОТЕХНИКИ И ЭЛЕКТРОНИКИ РАН**

Учебно-научная лаборатория

«Нелинейная динамика (физический эксперимент)»

Поддержано ФЦП «Интеграция» (проект
А0057/99) и грантом «Ведущие научные
школы» (проект РФФИ (96-15-96536))

Б.П. БЕЗРУЧКО, Д.А. СМИРНОВ

ПОСТРОЕНИЕ МОДЕЛЬНЫХ ОТОБРАЖЕНИЙ ПО ХАОТИЧЕСКИМ ВРЕМЕННЫМ РЯДАМ

Учебно-методическое пособие

Государственный учебно-научный центр «Колледж»

Саратов, 2000

УДК 530.18
Б 39

Безручко Б.П., Смирнов Д.А.

Б 39 Построение модельных отображений по хаотическим временным рядам. Учебно-методическое пособие, – Саратов: Издательство ГосУНЦ «Колледж», 2000 – 39 с.

Рассматриваются подходы к построению динамических моделей с дискретным временем (точечных отображений) по скалярным временным рядам, не имеющим видимой закономерности. Описаны методы глобальной и локальной реконструкции отображений. Предлагается, пользуясь готовыми программами, получить хаотические временные реализации эталонных отображений, восстановить по этим временным рядам уравнения дискретной модели и оценить качество реконструкции (две практических работы).

Работы предназначены для практических занятий по курсу «Математическое моделирование» для студентов факультета нелинейных процессов и физического факультета Саратовского госуниверситета.

Рецензент: старший научный сотрудник Саратовского отделения института радиотехники и электроники РАН, к.ф.-м.н. Селезнев Е.П.

© Б.П. Безручко,
Д.А. Смирнов,
2000

© Изд-во ГосУНЦ
«Колледж»,
2000

Содержание

Введение	4
1. Конструирование динамической модели по временному ряду	6
2. Критерии эффективности	10
3. Глобальные модели	14
3.1. Методика глобальной реконструкции	14
3.2. Практическое задание (работа № 1)	16
3.3. Контрольные вопросы	18
4. Локальные модели	18
4.1. Методика локальной реконструкции	18
4.2. Прямой и итерационный методы прогноза	20
4.3. Метод быстрого поиска ближайших соседей	21
4.4. Практическое задание (работа № 2)	22
4.5. Контрольные вопросы	24
Приложение 1. Описание программы MapSimulator	25
Приложение 2. Описание программы GlobalMap	28
Приложение 3. Описание программы LLMap	33
Литература	37

Введение

В отличие от статистического (вероятностного) подхода, оперирующего средними значениями или вероятностью обнаружения искомой величины в определенном интервале значений, динамический (детерминистический) подход к моделированию предполагает точное задание состояния объекта и однозначный прогноз его дальнейшего поведения. Это требует использования соответствующего математического аппарата. Классическим материалом для создания динамических моделей являются дифференциальные уравнения, для которых на уровне теоремы доказана единственность решения при заданных начальных условиях. Многими достоинствами обладают и однозначные отображения¹, которые рассматриваются в данной работе:

$$\mathbf{x}_{i+1} = \mathbf{G}(\mathbf{x}_i), \quad (1)$$

где $\mathbf{x} \in \mathbf{R}^D$ – вектор состояния², D — размерность динамической системы, i – дискретное время, \mathbf{G} — D -мерное отображение.

Ранее, до становления в 70-е годы концепции динамического хаоса, детерминированность ассоциировалась с простотой (регулярностью) поведения маломерных систем. Сложность (тем более, беспорядочность) движений считалась присущей лишь системам с чрезвычайно большим числом элементов и говорила о неизбежности статистического рассмотрения [1]. Обнаружение хаотических решений систем всего лишь трех нелинейных дифференциальных уравнений первого порядка и одномерных нелинейных отображений, доступных анализу на калькуляторе, подтолкнуло к поиску путей построения динамической модели даже по беспорядочному временному ряду.

¹ Отображение – закон, по которому каждому элементу x некоторого заданного множества X сопоставляется однозначно определенный элемент другого множества Y (при этом X может совпадать с Y).

² Т.е. вектор \mathbf{x} в данный момент времени определяет совокупность скалярных величин x_1, x_2, \dots, x_D в тот же момент времени, которые однозначно задают состояние системы. Здесь и далее векторные величины выделены жирным шрифтом.

Использование отображений для конструирования эволюционных моделей традиционно для популяционной биологии. В настоящее время сфера их приложений исключительно широка — отображения хорошо зарекомендовали себя для описания процессов в системах различной природы, в частности, физических, химических, социальных. Рекордсменом можно считать квадратичное одномерное отображение типа $x_{n+1} = rx_n(1 - x_n)$, где x — динамическая переменная, r — параметр, $n = 0, 1, 2, \dots$ — дискретное время. С его помощью были промоделированы и изучены универсальные закономерности перехода к хаосу через последовательность бифуркаций удвоения периода. На моделировании эволюции квазипериодических движений и переходе к хаосу при их разрушении «прославилось» отображение окружности. Многие закономерности сложной динамики пространственно развитых систем моделируют решетки связанных отображений.

Популярности точечных отображений не мешает даже то, что они оперируют значениями переменной в дискретные моменты времени, не описывая состояние объекта в промежутках между отсчетами, и в этом смысле уступают дифференциальным уравнениям. Во-первых, для многих объектов значения характеризующих величин по своему смыслу определены лишь в дискретные моменты времени и использование отображений для их описания очень естественно. Например, суммарное (годовое) количество пятен на Солнце измеряется ежегодно [1], курсы валют устанавливаются с суточным интервалом, числовые отсчеты появляются на выходе аналого-цифровых преобразователей тоже с некоторой частотой. Во-вторых, при решении некоторых задач информация о поведении между отсчетами не важна, например, если требуется просто классифицировать движения: периодическое квазипериодическое и т.п. В-третьих, дискретные модели «понятны» цифровым ЭВМ — для анализа отображений не требуется осуществлять непростой переход к разностным схемам, временные затраты при их численном исследовании минимальны.

Далее мы познакомимся с необходимыми для этого процедурами, ограничившись скалярными рядами и одномерными моделями. В разделах 1 и 2 будут рассмотрены основные этапы процедуры моделирования по временным рядам. Раздел 3 посвящен *глобальному* моделированию (при котором модель описывает поведение объекта во всем фазовом пространстве). В разделе 4 рассматриваются проблемы *локального* моделирования, когда модель описывает поведение системы «по кускам» в отдельных (локальных) областях фазового пространства. В приложения вынесены описания программ и частные вопросы. Учитывая, что построение отображения не требует, по сравнению с дифференциальными моделями, предварительной обработки экспериментального ряда (в частности, сглаживания перед дифференцированием), что экономит время для объяснения, в данной работе повышенное внимание уделяется тестированию – оценке качества получающихся моделей.

1. Конструирование динамической модели по временному ряду

Разнообразие моделируемых объектов и условий их функционирования неизбежно отражается на проявлении специфических моментов и акцентов при создании соответствующих целям моделирования динамических систем. Тем не менее, можно выделить общие для всех случаев этапы работы с временными рядами. Представим их с помощью следующей блок-схемы рис.1.

Моделирование начинается с выбора структуры модели и базисных функций (этап 1). На этом этапе на основании априорных данных, по оценке размерности множества, восстановленного в фазовом пространстве по скалярному временному ряду $\{v_i\}$, или интуитивно выбирается размерность D модельного отображения, а также вид функций, с помощью которых (комбинацией которых) будет осуществляться аппроксимация, т.е. выбирается некоторый *функциональный базис*.³

³ С 1987 года (с пионерских работ Кратчфилда и Макнамары [2], Кремерса и Хюблера [3], Фармера и Сидоровича [4]) разрабатываются *универсальные* способы построения динамических моделей по временным рядам. Исследователь может применять эти способы автомати-

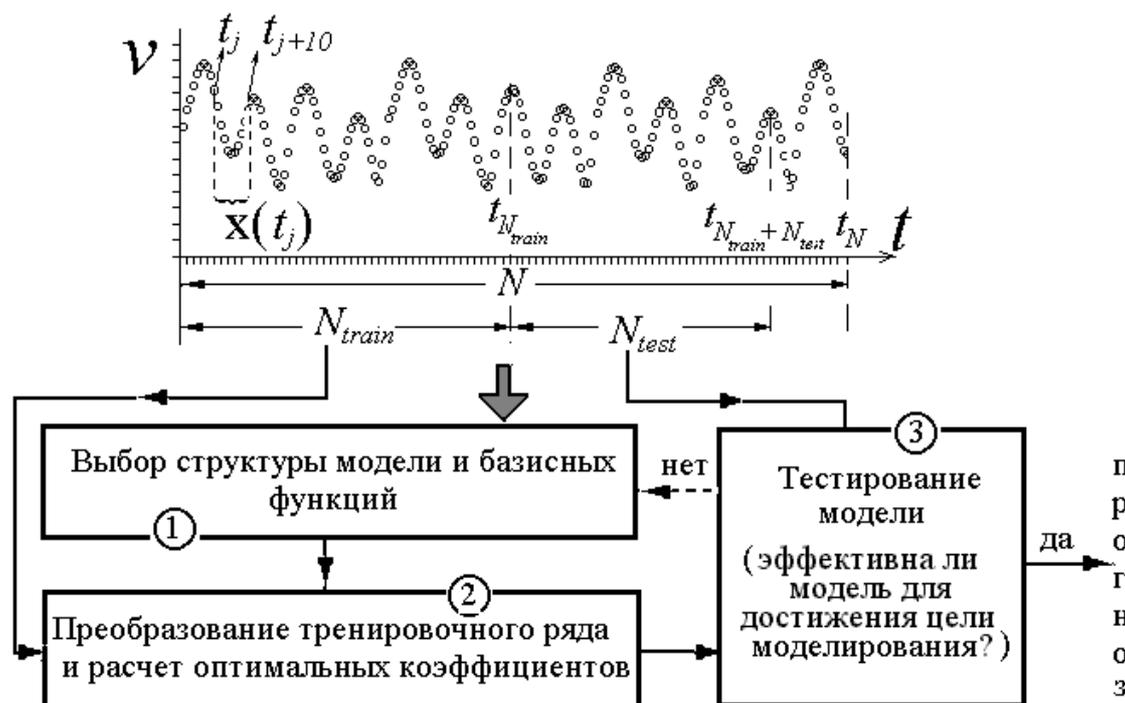


Рис.1. Схема процедуры моделирования по временному ряду.

На следующем этапе (этап 2) тренировочная часть временного ряда преобразуется в соответствии с выбранной структурой модели (числом переменных): ослабляются шумы, с необходимой частотой выбираются точки, восстанавливаются дополнительные переменные (если $D > 1$, исходный ряд – скалярный, и необходим переход к векторному ряду $\{x_i\}$ ⁴) и т.п. Далее, используя подготовленный временной ряд $\{x_i\}$, подбирают функцию G , аппроксимирующую зависимость x_{i+1} от x_i [2,4,7-10]. Для этого искомую функцию представляют, например, в виде суммы выбранных на первом этапе *базисных функций* с неизвестными коэффициентами. Затем вычисляют значения коэффициентов, которые обеспечивают наилучшее соответствие модели и экспериментальных данных (*подгонка* модели к временному ряду).

чески, не задумываясь над выбором вида (структуры) модельных уравнений. Хотя, многие авторы отмечают, что учет дополнительной априорной информации о системе может существенно облегчить построение модели и сделать сами модельные уравнения более эффективными. Однако универсальных рекомендаций для включения в модель априорной информации об объекте дать, по-видимому, невозможно.

⁴Процедуру построения («восстановления») по скалярному временному ряду $\{v_i\}$ ряд векторов состояния $\{x_i\}$ называют *реконструкцией* фазовой траектории [5,6].

На третьем этапе результаты предсказания по полученной модели сравниваются с данными тестовой части ряда. При этом критерии качества, эффективности определяются целями моделирования. Например, критериями могут служить прогностические возможности модели или качественное сходство ее поведения с поведением объекта. Если модель работоспособна, то процесс моделирования прекращается, а если нет – процедура повторяется с измененной структурой уравнений и/или новыми базовыми функциями.

Не вдаваясь в подробности описанной схемы, остановимся на некоторых важных моментах:

1) При переходе к векторному ряду для восстановления дополнительных координат (величин x_i) весьма удобен метод временных задержек. Т.е. координатами вектора состояния системы служат последовательные значения наблюдаемой v в моменты времени, разделенные некоторым временем задержки τ [2–7,9]:

$$\mathbf{x}(t_i) = \begin{pmatrix} v(t_i) \\ v(t_i + \tau) \\ \dots \\ v(t_i + (D-1)\tau) \end{pmatrix}. \quad (2)$$

Причем время задержки можно выбрать практически любым [6].

2) На практике, однако, выбор оптимального значения τ является отдельной задачей⁵. В данном пособии мы не будем касаться этой проблемы. Будем считать время задержки равным интервалу выборки Δt (для рассматриваемых далее в практическом задании временных рядов соседние значения наблюдаемой меняются резко, поэтому такой выбор времени задержки целесообразен):

⁵ Существует несколько вариантов выбора времени задержки. Так в [3] использовался первый ноль автокорреляционной функции. В настоящее время в качестве времени задержки часто используется (см., например, [11,12]) первый минимум функции взаимной информации наблюдаемого процесса.

$$\mathbf{x}_i = \begin{pmatrix} v_i \\ v_{i+1} \\ \dots \\ v_{i+D-1} \end{pmatrix}. \quad (3)$$

3) Выбор размерности модели D можно осуществить на основе предварительного анализа временного ряда⁶. Однако, здесь мы рассматриваем другой подход. Попытаемся сначала построить модель низкой размерности. Если эффективную модель получить не удастся, то размерность следует увеличить на единицу и повторить процесс построения модели. И так далее, пока не будет найдена удовлетворительная модель.

Для примера на рис.1 показан вариант формирования 11-мерного вектора при $\tau = \Delta t$.

4) Выбор метода восстановления векторов состояния и функционального базиса для аппроксимации определяет вид (*структуру*) модельных уравнений. Далее задача моделирования сводится лишь к нахождению оптимальных значений коэффициентов, входящих в модельные уравнения (подгонке модели). Так, при указанной реконструкции вектора состояния методом задержек с $\tau = \Delta t$ модельное отображение (1) сводится к более простому виду:

$$\begin{aligned} x_1(t_{j+1}) &= x_2(t_j), \\ x_2(t_{j+1}) &= x_3(t_j), \\ &\dots, \\ x_D(t_{j+1}) &= G(x_1(t_j), x_2(t_j), \dots, x_D(t_j)). \end{aligned} \quad (4)$$

Что можно эквивалентно переписать как⁷

$$v_{j+D} = G(v_j, v_{j+1}, \dots, v_{j+D-1}). \quad (5)$$

⁶ Для оценки размерности использовались различные методы: метод Грассбергера-Прокаччия [12], метод ложных ближайших соседей [11], метод главных компонент [13].

⁷ Так, если отображение одномерное ($D = 1$, а значит $\mathbf{x}(t_j) = v_j$), то функция G отображает точку из некоторого отрезка в точку того же отрезка. График такого отображения представляет собой линию на плоскости v_j, v_{j+1} . В случае $D = 2$ ($\mathbf{x}(t_j) = (v_j, v_{j+1})^T$) двумерное отображение (5) действует уже на точку плоскости v_j, v_{j+1} , а графиком функции G является некоторая поверхность.

Функцию G , аппроксимирующую зависимость v_{j+D} от $v_j, v_{j+1}, \dots, v_{j+D-1}$, можно попытаться записать с помощью одной формулы (выражения). И затем вычислить (оценить) по временному ряду значения коэффициентов, входящих в эту формулу.

5) Полученная формула будет определять функцию G во всем фазовом пространстве — глобально. На этом основана методика так называемой *глобальной реконструкции*, описанию которой посвящен раздел 3, где предлагается практическое задание на применение данной процедуры.

6) Другим возможным подходом является *локальная реконструкция*. При этом отыскиваются выражения для функции G «по кускам» — отдельно для каждой достаточно малой области фазового пространства (т.е. локально). В разделе 4 изложен алгоритм конструирования локальных линейных моделей и приведено практическое задание на применение этого алгоритма.

2. Критерии эффективности модели

После того, как модель построена (значения всех коэффициентов вычислены), необходимо проверить ее работоспособность. Как уже говорилось, критерии эффективности модели определяются целями моделирования. Мы же для иллюстрации будем использовать несколько критериев:

- дальность прогноза, обеспечиваемого моделью,
- качественное соответствие поведения модели и объекта;
- в некоторых примерах (где это возможно) будем также сравнивать полученные значения коэффициентов с их истинными значениями.

Обсудим каждый из перечисленных критериев.

1) Для расчета дальности прогноза используем подход Фармера и Сидоровича [4]. Рассчитаем среднеквадратичную ошибку прогноза, обеспечиваемого моделью, на определенное число шагов вперед, причем для проверки будем использовать различные участки тестового временного ряда.

Зададим начальные условия модели, используя первые значения наблюдаемой из тестового ряда: $x_1 = v(t_{N_{train}+1})$, $x_2 = v(t_{N_{train}+2}), \dots, x_D = v(t_{N_{train}+D})$.

Итерируя модельное отображение (5) L_{test} раз, получим временную реализацию модели длиной L_{test} шагов. Эта реализация содержит *предсказанные* значения v в моменты времени

$$t_{N_{train}+D+1}, t_{N_{train}+D+2}, \dots, t_{N_{train}+D+L_{test}}.$$

Сравним ее с соответствующим участком тестового ряда (рис.2). Обозначим предсказанное с помощью модели значение наблюдаемой v на T шагов вперед (т.е. *прогноз с упреждением T*), начиная с момента времени t_i , как $\hat{v}(t_i, T)$. Квадрат ошибки прогноза с упреждением T определяется выражением

$$\sigma_T^2(t_{N_{train}+D}) = (\hat{v}(t_{N_{train}+D}, T) - v(t_{N_{train}+D+T}))^2.$$

Нижний индекс T означает время упреждения ($1 \leq T \leq L_{test}$). Ошибка прогноза в общем случае зависит от выбранного момента времени t_i , с которого мы начинаем прогноз процесса.

Теперь выберем другие начальные условия модели, сместившись по тестовому временному ряду на S_{test} (где $S_{test} \geq 1$) шагов вперед (см. рис.2): $x_1 = v(t_{N_{train}+1+S_{test}}), x_2 = v(t_{N_{train}+2+S_{test}}), \dots, x_D = v(t_{N_{train}+D+S_{test}})$. Вновь рассчитаем квадрат ошибки прогноза:

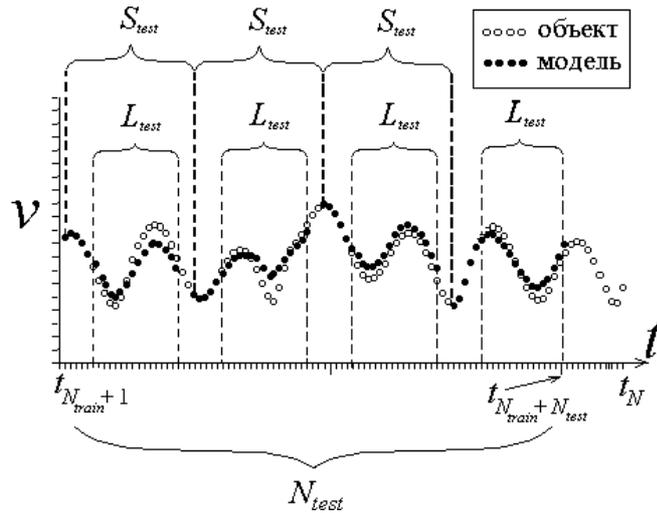


Рис.2. Иллюстрация к вычислению дальности прогноза по тестовому временному ряду. Здесь светлыми кружками показана временная реализация объекта, черными — прогноз для отдельных участков. В данном случае выбрано значение размерности модели $D = 5$ (первые 5 значений исходной и модельной реализаций одинаковы для каждого тестового участка). Длина одного тестового участка $L_{test} = 16$, общее количество тестовых участков $K_{test} = 4$, сдвиг между соседними участками $S_{test} = 23$, длина тестового ряда $N_{test} = 90$.

$$\sigma_T^2(t_{N_{train}+D+S_{test}}) = (\hat{v}(t_{N_{train}+D+S_{test}}, T) - v(t_{N_{train}+D+S_{test}+T}))^2.$$

И так далее рассчитаем ошибки прогноза σ_T^2 для K_{test} различных начальных векторов (различных участков тестового ряда), каждый из которых смещен на S_{test} шагов относительно предыдущего⁸. В итоге рассчитаем *нормированную среднеквадратичную* ошибку прогноза в зависимости от упреждения T :

$$\sigma_T = \sqrt{\frac{1}{K_{test}} \sum_{i=1}^{K_{test}} \sigma_T^2(t_{N_{train}+D+S_{test}(i-1)})} / Var, \quad (6)$$

где $Var = \frac{1}{N_{train}} \sum_{i=1}^{N_{train}} (v_i - \bar{v})^2$ — средний квадрат отклонения наблюдаемой от ее

среднего значения (используется для нормировки). Дальность прогноза τ_{pred} определяется как время возрастания величины σ от нуля (при $T = 0$) до определенного критического значения σ_c : $\sigma(\tau_{pred}) = \sigma_c$. Это критическое значение определяется тем, какая погрешность прогноза допустима. Мы примем значение $\sigma_c = 0.05$ (т.е. допустима погрешность 5 % от общего размаха колебаний).

Обращаем ваше внимание на то, что время прогноза хаотических процессов даже с использованием «идеальной» модели принципиально ограничено и это надо учитывать при оценке качества модели. Для оценки предельно возможной дальности прогноза можно использовать выражение

$$\tau_{pred} = \frac{1}{2\lambda_+} \ln \frac{\sigma_c^2 \cdot Var}{\sigma_v^2 + \sigma_\mu^2 + \sigma_{\Delta M}^2}, \quad (7)$$

где λ_+ — старший ляпуновский показатель, σ_v^2 определяет флуктуации в системе, σ_μ^2 — погрешности измерительных приборов, $\sigma_{\Delta M}^2$ — погрешность модели («шумы незнания») [14].

2) Для качественного сравнения режимов поведения модели и объекта нужно задать начальные условия для модели (из тестового ряда) и получить ее

⁸ Таким образом, общая длина ряда, используемого для тестирования модели, составляет $N_{test} = L_{test} + D + (K_{test} - 1) \cdot S_{test}$ значений.

фазовую траекторию. Затем сравнить проекции фазовых портретов объекта и модели на плоскость (v_i, v_{i+1}) .

Если полученная модель не дает удовлетворительного прогноза или не демонстрирует режима, качественно схожего с экспериментальным, то можно увеличить значения порядка полинома K или размерности модели D и повторить процесс построения модели и проверки ее эффективности. Можно также попытаться использовать другой вид функции G .

3) Относительная погрешность определения значения некоторого коэффициента c_i определяется как

$$\varepsilon_{coeff}^{(i)} = \frac{|c_i - c_i^{(0)}|}{|c_i^{(0)}|} \quad (8)$$

в случае $c_i^{(0)} \neq 0$, где $c_i^{(0)}$ — истинное значение данного коэффициента. Если $c_i^{(0)} = 0$, то ошибку восстановления значения коэффициента определяют как

$$\varepsilon_{coeff}^{(i)} = |c_i|. \quad (9)$$

Общую ошибку восстановления значений коэффициентов по временному ряду ε_{coeff} можно определить, например, как среднее значение индивидуальных ошибок или как максимальную из всех индивидуальных ошибок:

$$\varepsilon_{coeff} = \max_{i=1, \dots, M} \{ \varepsilon_{coeff}^{(i)} \} \quad (10)$$

На этом завершим обсуждение критериев качества модели. Заметим, что введенные здесь обозначения используются далее в описаниях программ для реконструкции модельных отображений (приложения).

В заключение отметим, что для надежного вычисления значений коэффициентов модели необходимо, чтобы длина тренировочного временного ряда была достаточно большой — не меньше некоторого порогового значения. С ростом уровня шума в системе это пороговое значение увеличивается, т.к. требуется больше данных для усреднения. Оно увеличивается также с ростом числа коэффициентов модели.

3. Глобальные модели

3.1. Методика глобальной реконструкции

Пусть имеется скалярный временной ряд, содержащий N значений наблюдаемой величины v , измеренных в последовательные моменты времени t_i : $v_i = v(t_i), i = 1, \dots, N$ (рис.1). Временной ряд из первых N_{train} значений наблюдаемой используем для построения модели, поэтому назовем его «тренировочным». Продолжение тренировочного временного ряда («тестовый» ряд длиной N_{test}) используем *позже* для проверки эффективности построенной модели.

Выберем некоторое значение размерности модели D . С помощью метода временных задержек (3) восстановим векторный временной ряд, содержащий $N_{train} - D + 1$ членов. Будем строить модель в виде (5). Функцию G выберем в виде полинома порядка K :

$$G(x_1, x_2, \dots, x_D) = \sum_{l_1, l_2, \dots, l_D=0}^K c_{l_1, l_2, \dots, l_D} \prod_{j=1}^D x_j^{l_j}, \sum_{j=1}^D l_j \leq K. \quad (11)$$

Этот выбор — лишь один из возможных. Его достоинствами являются универсальность и наличие стандартных процедур применения, а недостатком — громоздкий во многих случаях вид модели (большое количество коэффициентов). Для построения полиномиальной модели нужно определить значения M коэффициентов: $M = (D+K)!/(D!K!)$. Эти значения выбираются так, чтобы среднеквадратичная погрешность аппроксимации ε зависимости (5) была минимальна (метод наименьших квадратов):

$$\varepsilon^2 = \frac{1}{N_{train} - D} \sum_{j=1}^{N_{train} - D} [v_{j+D} - G(v_j, v_{j+1}, \dots, v_{j+D-1})]^2 = \min. \quad (12)$$

Поскольку неизвестные коэффициенты c_{l_1, l_2, \dots, l_D} линейно входят в выражение (11), задача сводится к решению системы линейных алгебраических уравнений. Их можно представить в матричном виде следующим образом:

$$\mathbf{A}^T \cdot \mathbf{A} \cdot \mathbf{C} = \mathbf{A}^T \cdot \mathbf{B}. \quad (13)$$

Здесь \mathbf{A} — матрица значений базисных функций (размерности $(N_{train} - D) \times M$), \mathbf{B} — вектор значений аппроксимируемой величины (размерности $N_{train} - D$), \mathbf{C} — вектор коэффициентов модели (размерности M), верхний индекс T означает транспонирование:

$$\mathbf{A} = \begin{pmatrix} 1 & v_1 & \dots & v_D & v_1^2 & \dots & v_D^K \\ 1 & v_2 & \dots & v_{D+1} & v_2^2 & \dots & v_{D+1}^K \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & v_{N_{train}-D} & \dots & v_{N_{train}-1} & v_{N_{train}-D}^2 & \dots & v_{N_{train}-1}^K \end{pmatrix}, \mathbf{B} = \begin{pmatrix} v_{D+1} \\ v_{D+2} \\ \dots \\ v_{N_{train}} \end{pmatrix}, \quad (14)$$

$$\mathbf{C} = (c_{0,0,\dots,0}, c_{1,0,\dots,0}, \dots, c_{0,0,\dots,1}, c_{2,0,\dots,0}, \dots, c_{0,0,\dots,K})^T. \quad (15)$$

Систему уравнений (13) называют нормальной системой уравнений. Ее решение дает оценки наименьших квадратов для значений коэффициентов c_{l_1, l_2, \dots, l_D} .

Отметим, что для глобальной модели можно использовать и дополнительные критерии эффективности, связанные с качеством аппроксимации: погрешность аппроксимации ε и погрешность аппроксимации ε_{test} , рассчитанную по тестовому временному ряду.

1) Величина погрешности аппроксимации ε (12) может служить предварительным показателем качества модели. Удобно использовать *нормированную* величину погрешности $\frac{\varepsilon}{\sqrt{Var}}$ и выражать ее в процентах. Если нормированная погрешность аппроксимации превышает, скажем, 50 %, то трудно рассчитывать на эффективность модели.

2) Казалось бы, чем меньше величина ε , тем лучше модель. Однако обратим внимание, что погрешность ε может обратиться и в ноль, если количество коэффициентов модели равно $N_{train} - D$. Но такая громоздкая модель наверняка не будет сколько-нибудь эффективной для описания исходной системы. Используя значения коэффициентов c_{l_1, l_2, \dots, l_D} , рассчитанные по тренировочному

ряду, можно вычислить погрешность аппроксимации зависимости (5) ε_{test} по тестовому временному ряду:

$$\varepsilon_{test}^2 = \frac{1}{N_{test}-D} \sum_{j=N_{train}+1}^{N_{train}+N_{test}-D} [v_{j+D} - G(v_j, v_{j+1}, \dots, v_{j+D-1})]^2. \quad (16)$$

Это более надежная характеристика качества модели, т.к. тестовый ряд не используется для подгонки значений коэффициентов.

3.2. Практическое задание (работа № 1)

Цель работы. Применить изложенный алгоритм глобальной реконструкции на практике:

- научиться подбирать оптимальные значения размерности модели и порядка полинома,
- отследить влияние длины тренировочного временного ряда на качество восстановленной модели
- отследить зависимость необходимой длины ряда от уровня шума.

Задача. Получить и записать скалярный временной ряд, генерируемый каким-либо эталонным отображением («чистый» и зашумленный) и восстановить по нему различные модельные отображения. На данном примере проследить влияние различных факторов на качество реконструкции.

- 1) С помощью программы **MapSimulator** (приложение 1) получите и запишите не зашумленный хаотический сигнал длиной 1000 значений, генерируемый одним из эталонных отображений (заданным преподавателем). Например, скалярный временной ряд, генерируемый отображением Эно при значениях параметров $a = 1.5$, $b = 0.2$ (см. рис.П.1.1).
- 2) С помощью программы **GlobalMap** (приложение 2) восстановите по тренировочному временному ряду (длиной 1000 итераций) одномерное отображение ($D = 1$), используя аппроксимацию полиномом различных порядков K (от 0 до 10). Отследите зависимости $\varepsilon(K)$, $\varepsilon_{test}(K)$ и постройте их графики

(для построения графиков можно использовать результаты построения моделей из файла отчета — см. рис.П.2.2). Выберите наилучшую модель.

- 3) Если не удастся получить достаточно эффективную модель (т.е. модель, демонстрирующую качественно схожее с объектом поведение и/или дающую хороший количественный прогноз), то по тому же ряду реконструируйте двухмерное отображение ($D = 2$), также используя аппроксимацию стандартным полиномом различных порядков K (от 0 до 10). Найдите K , при котором достигается наилучшее соответствие модели и объекта, используя для этого графики зависимостей $\varepsilon_{test}(K)$ и $\tau_{pred}(K)$ и визуальное сравнение фазовых портретов.
- 4) Если модели вновь не удовлетворительны, увеличьте D и повторите действия п.3. Определите оптимальные значения D и K , позволяющие получить эффективную модель.
- 5) Восстанавливайте отображение с найденными оптимальными значениями D и K , используя различную длину тренировочного ряда N_{train} (для примера из п.1, можно начинать с 8 значений и далее: 10, 15, 25, 50, 100, 250, 500, 750 значений). Постройте графики зависимости $\varepsilon_{test}(N_{train})$, $\tau_{pred}(N_{train})$ и $\varepsilon_{coeff}(N_{train})$. По этим графикам определите оптимальную длину тренировочного ряда N_{opt} — такую, что при дальнейшем увеличении N_{train} результаты, достигнутые с помощью модели, не улучшаются.
- 6) Запишите временной ряд (тот же, что в п.1), добавив шум с некоторым стандартным отклонением σ_{noise} (например, для отношения сигнал/шум в 100 дБ, 80 дБ, 60 дБ, 40 дБ). Прделайте для этих зашумленных рядов задание 5. Сравните максимальную достигаемую дальность прогноза с оценкой (14). Постройте график зависимости $N_{opt}(\sigma_{noise})$ и соответствующий график для наибольшей достигаемой дальности прогноза $\tau_{pred}(\sigma_{noise})$ (а также график $\varepsilon_{coeff}(\sigma_{noise})$, если есть, с чем сравнивать значения коэффициентов восстановленного отображения).

3.3. Контрольные вопросы

1. Что такое скалярный временной ряд? Какие основные подходы к построению модели по временному ряду существуют?
2. В чем состоит принцип конструирования динамической модели по временному ряду?
3. Что понимают под термином «глобальная реконструкция»? В чем состоит процедура построения глобального модельного отображения?
4. Какие критерии эффективности модели могут использоваться? Чем определяется выбор критерия?
5. Как рассчитать дальность прогноза с помощью полученного модельного отображения?
6. Чем обусловлена принципиальная ограниченность дальности прогноза поведения хаотических систем?
7. Какие факторы влияют на качество построенной модели?

4. Локальные модели

4.1. Методика локальной реконструкции

Как уже было сказано в разделе 1, в отличие от глобальных моделей, направленных на описание динамики во *всем* фазовом пространстве, локальные модели описывают поведение системы «по кускам» (по-разному в различных небольших областях фазового пространства). Далее будут рассмотрены особенности построения локальных *линейных* моделей [4,7].

При той же постановке проблемы, что и в разделе 3, будем строить модель в виде (5), не стремясь аппроксимировать функцию G глобально во всем фазовом пространстве.

Методом временных задержек восстановим по тренировочному скалярному временному ряду (длиной N_{train}) векторный временной ряд $\{\mathbf{x}(t_j)\}_{j=1}^{N_{train}-D+1}$: $\mathbf{x}(t_j) = (v_j, v_{j+1}, \dots, v_{j+D-1})^T$. Сопоставим каждый из $N_{train} - D + 1$

векторов тренировочного ряда и первый вектор тестового ряда $\mathbf{x}(t_{N_{train}+1}) = (v_{N_{train}+1}, v_{N_{train}+2}, \dots, v_{N_{train}+D})^T$ (см. рис.1). Используя эти данные, сделаем прогноз значений координат следующего вектора тестового ряда $\mathbf{x}(t_{N_{train}+2})$ (фактически, неизвестным является только значение его последней координаты $v_{N_{train}+D+1}$). Для этого во всем тренировочном ряду найдем векторы, которые являются близкими «соседями» вектора $\mathbf{x}(t_{N_{train}+1})$ (рис.3). Т.е. векторы в некоторые моменты времени t_{n_i} , для которых выполняется условие

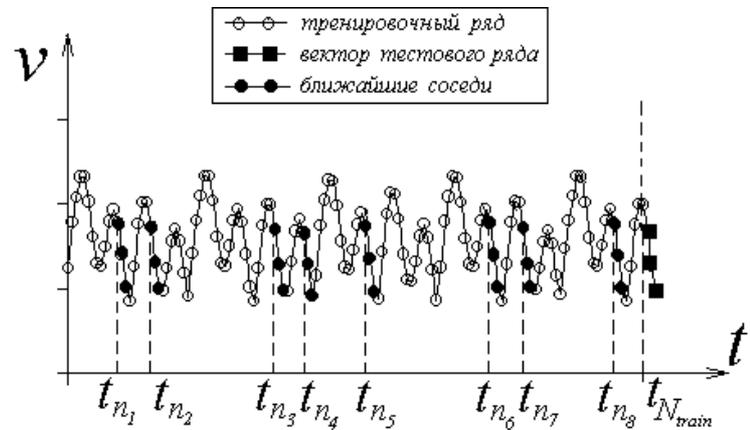


Рис.3. Ближайшие соседи первого вектора тестового ряда (черные квадраты), найденные по тренировочному временному ряду (черные кружки).

$$\|\mathbf{x}(t_{n_i}) - \mathbf{x}(t_{N_{train}+1})\| \leq \varepsilon_{box}, \quad (17)$$

где $\|\cdot\|$ — норма вектора: для упрощения расчетов обычно используется норма⁹ $\|\mathbf{x}\| = \max\{|x_1|, |x_2|, \dots, |x_D|\}$, ε_{box} — размер окрестности (для выбранной нормы окрестность представляет собой гиперкуб со стороной ε_{box}).

Пусть найдено k соседей (на рис.4 они показаны кружками). Предположим, что функция G линейна в окрестности «точки» $\mathbf{x}(t_{N_{train}+1})$:

$$G(x_1, x_2, \dots, x_D) = c_0 + \sum_{i=1}^D c_i x_i. \quad (18)$$

Для каждого из найденных векторов $\mathbf{x}(t_{n_i})$ дальнейшее поведение объекта известно (на рис.4 треугольниками показаны векторы состояния в моменты вре-

⁹ Можно использовать и любую другую норму. Например, привычную евклидову норму: $\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2 + \dots + x_D^2}$, но это приведет к увеличению затрат машинного времени.

мени t_{n_i+1}). По этим данным методом наименьших квадратов можно определить значения коэффициентов функции G :

$$\sum_{i=1}^k [v(t_{n_i+D}) - G(\mathbf{x}(t_{n_i}))]^2 = \min. \quad (19)$$

Для применения метода наименьших квадратов необходимо, чтобы количество найденных соседей k было не меньше количества коэффициентов $D+1$. Поэтому размер окрестности не должен быть слишком малым, чтобы нашлось достаточно соседей. Но размер окрестности не должен быть и слишком большим, чтобы эффективной была линейная аппроксимация. Обоим этим требованиям можно удовлетворить, только если количество данных (длина тренировочного ряда N_{train}) достаточно велико.

Если количество соседей меньше, чем количество коэффициентов, то будем использовать локально постоянную аппроксимацию: $G(\mathbf{x}) = c_0$. Если соседей не нашлось, то прогноз осуществляется по принципу «завтра как сегодня», т.е. $v(t_{N_{train}+D+1}) = v(t_{N_{train}+D})$.

4.2. Прямой и итерационный методы прогноза

Чтобы осуществить прогноз на несколько итераций вперед, можно использовать *предсказанное* значение $v(t_{N_{train}+D+1})$ в качестве нового *известного* значения наблюдаемой. Далее нужно найти ближайших соседей для полученного вектора состояния $\mathbf{x}(t_{N_{train}+2})$, провести аппроксимацию функции G (18) уже в

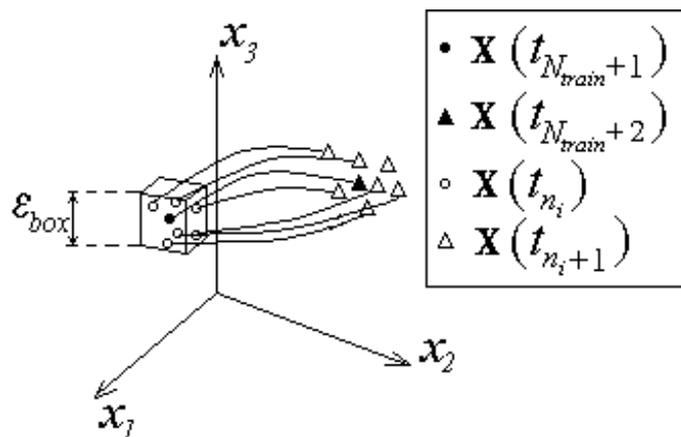


Рис.4. Ближайшие соседи (белые кружки) вектора тестового ряда (черные кружки) и следующие за ними векторы (белые треугольники). По этим данным нужно осуществить прогноз следующего вектора тестового ряда (черный треугольник).

окрестности этого нового вектора и сделать прогноз следующего значения величины v , и т.д. Такой метод прогноза называется *итерационным*.

Возможен и другой метод прогноза, который называется *прямым*. Он состоит в том, что для прогноза на T интервалов выборки вперед (с упреждением T) непосредственно аппроксимируется зависимость $v(t_{n_i+D-1+T})$ от $\mathbf{x}(t_{n_i})$. Таким образом, для прогноза с любым упреждением достаточно только один раз отыскать ближайших соседей вектора $\mathbf{x}(t_{N_{train}+1})$.

В [4] приведена следующая оценка для ошибки прогноза с помощью прямого метода:

$$\sigma(T) = C \cdot e^{2\mu T} \cdot N_{train}^{-2/d}, \quad (20)$$

где C — некоторая константа, μ — метрическая энтропия, d — информационная размерность аттрактора. Точность прогноза с использованием итерационного метода, как показывают оценки [7], несколько выше.

Эффективность (см. раздел 4) локальных линейных моделей будем оценивать по дальности прогноза (как для прямого, так и для итерационного методов) и по виду фазового портрета¹⁰.

4.3. Метод быстрого поиска ближайших соседей

На практике для применения изложенных методов существенно, что для осуществления прогноза наблюдаемого процесса каждый раз требуется находить в тренировочном ряде ближайших соседей текущего вектора состояния. Можно использовать прямолинейный подход для поиска соседей: рассчитать расстояния от каждого из векторов тренировочного ряда до текущего вектора и выбрать те векторы, для которых выполняется условие (19). Однако этот метод слишком нерационален. Если требуется осуществлять прогноз достаточно много раз, то он потребует значительных затрат машинного времени. Желательно

¹⁰ Будем сравнивать с оригиналом фазовый портрет модели, полученный только с помощью итерационного метода.

было бы не рассчитывать расстояния от текущего вектора до тех векторов тренировочного ряда, которые заведомо не могут быть его соседями.

Для этого используют, например, следующий подход. Векторы тренировочного ряда предварительно *сортируют*. Для сортировки используют первую и последнюю (наименее коррелированные) координаты векторов. Пусть v_{\min} и v_{\max} — минимальное и максимальное значение наблюдаемой, рассчитанные по тренировочному ряду. На плоскости (x_1, x_D) фазовая траектория лежит внутри квадрата со сторонами $x_1 = v_{\min}, x_1 = v_{\max}, x_D = v_{\min}, x_D = v_{\max}$ (см.

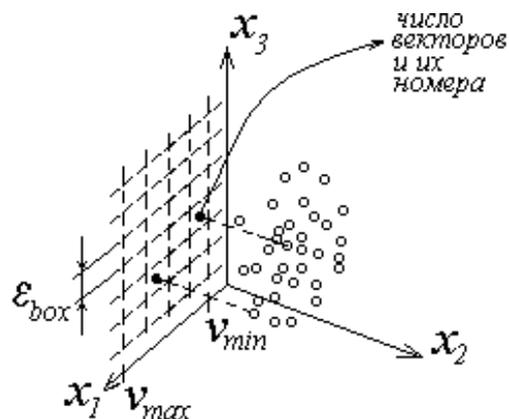


Рис.5. Векторы тренировочного ряда сортируются по значениям первой и последней координат. Для этого создается квадратный массив, значения элементов которого равны количеству векторов в соответствующей ячейке. В дополнительном массиве хранятся номера этих векторов.

рис.5). Этот квадрат разбивается на квадратные ячейки размером ε_{box} . Определяется, какой вектор в какую ячейку попадает. Создается квадратный массив, элементы которого соответствуют ячейкам разбиения. В элементы массива записывается количество векторов, попавших в соответствующие ячейки. Создается также линейный массив, в который записываются поочередно номера векторов, попавших в каждую ячейку. Таким образом, если теперь требуется отыскать ближайших соседей вектора x , то нужно определить, какой ячейке принадлежит этот вектор, и рассчитать расстояния от него только до тех векторов, которые принадлежат той же ячейке или смежным с ней (нужно проверить максимум 9 ячеек). Такой алгоритм существенно ускоряет процесс поиска ближайших соседей.

4.4. Практическое задание (работа № 2)

Цель работы. Применить на практике изложенный метод построения локальных линейных моделей:

- потренироваться в выборе оптимального (при данной длине тренировочного ряда) значения размера окрестности,
- проследить зависимость качества прогноза от количества данных и от уровня шума в системе.

Задача. Осуществить прогноз процесса, являющегося реализацией одного из эталонных отображений (заданного преподавателем).

1. С помощью программы **MapSimulator** (приложение 1) получите и запишите исходный временной ряд: не зашумленную хаотическую временную реализацию одного из эталонных отображений (заданного преподавателем) длиной 20000 итераций. Например, для отображения Эно при $a = 1.5$, $b = 0.2$ (см. рис.П.1.1).
2. С помощью программы **LLMap** (приложение 3) осуществите прогноз наблюдаемого процесса. Используйте тренировочный ряд длиной 100 итераций. Сразу выберите «правильную» размерность модели. Используйте для построения модели различный размер окрестности ε_{box} . Постройте график зависимости $\tau_{pred}(\varepsilon_{box})$ как для прямого, так и для итерационного методов. Для того чтобы понять причину эффективности или неэффективности прогноза просматривайте количество найденных соседей (файл neighbors.txt). Объясните вид полученной зависимости. Определите оптимальный размер окрестности ε_{opt} .
3. Используйте для построения локальной модели тренировочные ряды различной длины (250, 500, 1000, 2500, 5000, 10000, 15000 итераций). Выполните для них задание 2. Постройте графики для оптимального размера окрестности $\varepsilon_{opt}(N_{train})$ и соответствующей дальности прогноза $\tau_{pred}(N_{train})$. Сравните точность прогноза, полученного с помощью итерационного и прямого методов.

4. Запишите тот же временной ряд (п.1) с добавлением шума различной интенсивности (отношение сигнал/шум в 80, 60, 40 дБ). Выполните для одного из зашумленных рядов задание 3.
5. Постройте график зависимости $\tau_{pred}(\sigma_{noise})$, используя фиксированное (достаточно большое) значение N_{train} и $\varepsilon_{box} = \varepsilon_{opt}$.¹¹

4.5. Контрольные вопросы

1. В чем смысл термина «локальная реконструкция»?
2. Опишите алгоритм построения локальной линейной модели.
3. Каким образом можно осуществить прогноз наблюдаемого процесса с помощью локальной линейной модели?
4. Какими факторами определяется эффективность локальной линейной модели? Как влияет на нее длина временного ряда? Играет ли существенную роль выбор размера окрестности (при фиксированной длине временного ряда)?
5. В чем состоит метод быстрого поиска ближайших соседей? Почему возникает необходимость в использовании подобного метода?

¹¹ Сравните результаты (дальность прогноза) с результатами работы № 1, если вы построили модель по временной реализации того же объекта, что и в работе № 1.

Приложение 1. Описание программы MapSimulator

Программа **MapSimulator** позволяет записать скалярную временную реализацию динамической переменной одного из эталонных отображений. Здесь можно

- выбрать нужное отображение и переменную, временная реализация которой будет записываться,
- задать значения параметров отображения и начальные условия, при которых будет получен временной ряд,
- добавить нормальный белый шум нужной интенсивности к записываемому ряду,
- выбрать файл для записи данных (или создать новый),
- задать установки для записи ряда (длительность переходного процесса N_{trans} — первые N_{trans} итераций отображения не записываются, и длину записываемого ряда N).

Имеется возможность также изменить все указанные величины в течение сеанса работы с программой и записать другой временной ряд в новый файл. Пример внешнего вида окна программы представлен на рис.П.1.1.

Запустите файл **MapSimulator.exe**. Откроется окно программы **MapSimulator** и в левой его части – окно ввода (Input Window).

1) В окне ввода появится пронумерованный список отображений, реализации которых можно записать: “You can select one of the following exemplar maps: 1 — Quadratic map, 2 — Henon map, 3 — Dissipatively coupled quadratic maps, ...” («Вы можете выбрать одно из следующих эталонных отображений: 1 – квадратичное отображение, 2 — отображение Эно, 3 — диссипативно связанные квадратичные отображения, ...»), и программа предложит выбрать одно из них: “Select a map: ” («Выберите отображение: »). Введите номер нужного отображения.

2) Далее будет приведен явный вид выбранного отображения (формулы), и программа запросит поочередно значения его параметров: “Enter values of

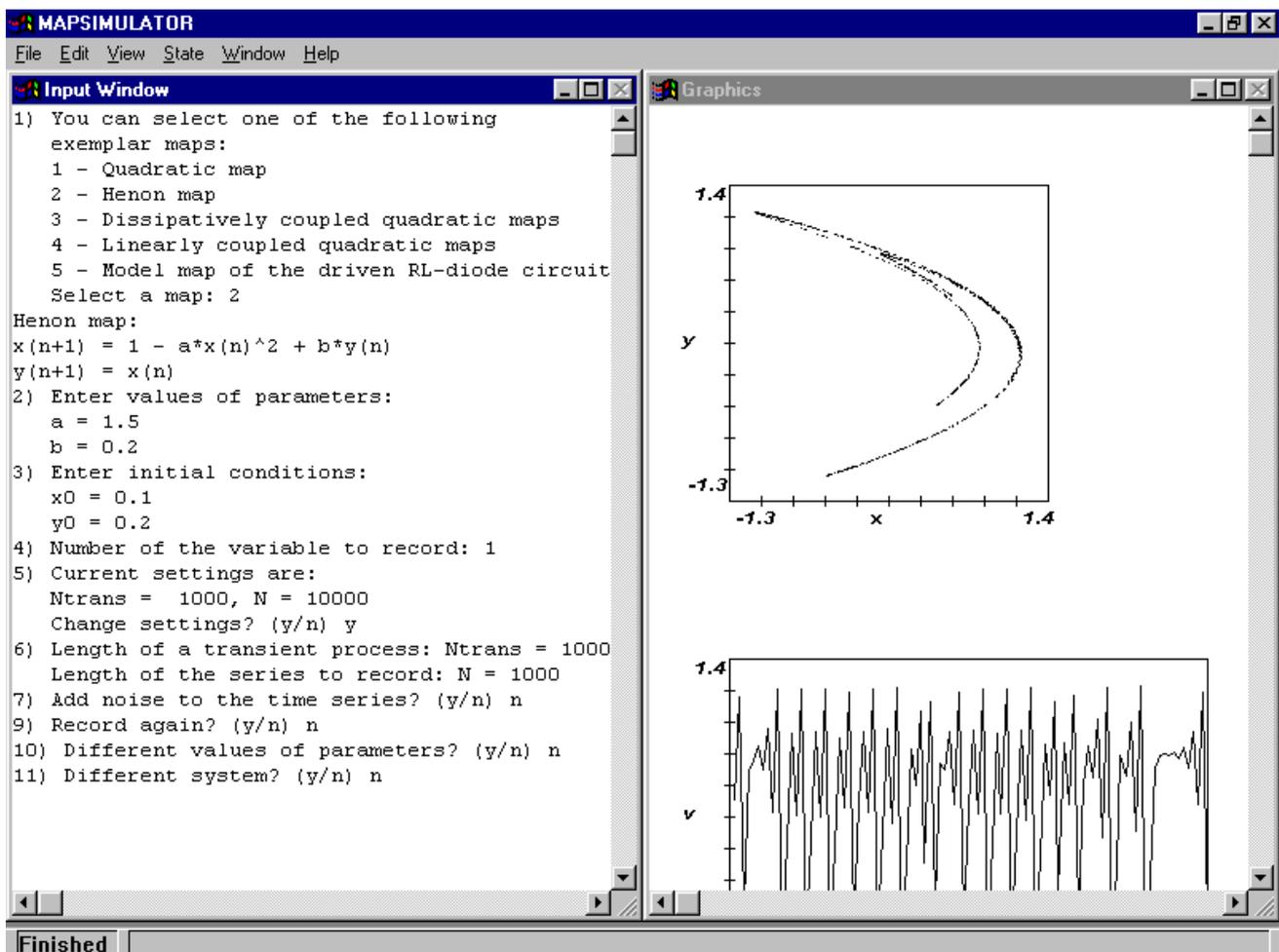


Рис.П.1.1. Внешний вид окна программы **MapSimulator** (данный сеанс работы с программой включал в себя запись временной реализации координаты x отображения Эно длиной 1000 итераций при значениях параметров $a = 1.5$, $b = 0.2$).

parameters: $a =$, $b =$ ” («Задайте значения параметров: $a =$, $b =$ ») и т.д. Введите нужные значения (обязательно вещественные числа – с разделяющей десятичной точкой).

3) Аналогично потребуется ввести начальные условия: “Enter initial conditions: $x_0 =$, $y_0 =$ ” («Введите начальные условия: $x_0 =$, $y_0 =$ »).

4) Далее (если отображение не одномерное) будет запрошен номер динамической переменной, временная реализация которой будет записана: “Number of the variable to record: ” («Номер переменной для записи: »). Введите соответствующее целое число: переменные нумеруются сверху вниз — по левой части формул отображения.

5) Появится сообщение: “Current settings are: $N_{trans} = \dots$, $N = \dots$ ” («Текущие установки: $N_{trans} = \dots$, $N = \dots$ ») и вопрос: “Change settings? (y/n) ” «Изме-

нить установки? (да/нет)». Если установки Вас не устраивают, то отвечайте положительно (введите символ “y”) и см. п. 6. Иначе отвечайте “n” и далее см. пункт 7.

6) Потребуется ввести длину переходного процесса N_{trans} : “Length of a transient process: Ntrain = ” («Длительность переходного процесса: Ntrans = ») и длину записываемого временного ряда N : “Length of the series to record: N = ” («Длительность временного ряда для записи: N = »). Введите нужные целые числа.

7) Следующий вопрос: “Add noise to the time series? (y/n)” («Добавить шум к временному ряду? (да/нет)»). Если хотите записать зашумленный ряд, ответьте положительно и см. пункт 8, иначе отвечайте “n” и см. пункт 9.

8) Потребуется ввести величину стандартного отклонения случайной величины: “Noise standard deviation: std = ” («Стандартное отклонение шума: std = »). При этом к значениям «чистой» временной реализации v_i при записи в файл прибавляются независимые случайные величины ξ_i , распределенные по нормальному закону с нулевым средним и заданным стандартным отклонением. В само отображение шумовая добавка не вводится, т.е. этот шум моделирует шум измерительной аппаратуры, а не флуктуации в системе.

9) Далее откроется окно диалога для выбора файла, в который будет записан временной ряд. Выберите нужный файл (или создайте новый). Временная реализация будет записываться в заданный файл, а в правой части окна программы появится окно “Graphics” («Графика»). В этом окне представлены фазовая траектория отображения и участок записанной временной реализации.

Появится вопрос: “Record again? (y/n)” («Записать снова? (да/нет)»). В случае положительного ответа — переходите к пункту 4: можно изменить настройки для записи временного ряда, выбрать шум другой интенсивности, другую динамическую переменную для записи, другой файл. Иначе см. пункт 10.

10) Следующий вопрос: “Different values of parameters? (y/n)” («Другие значения параметров? (да/нет)»). Отвечайте “y”, если нужно повторить запись

временного ряда при других значениях параметров и/или начальных условий — возвращайтесь к пункту 2. Иначе см. пункт 11.

11) Последний вопрос: “Different system? (y/n) ” («Другая система? (да/нет)»). Отвечайте положительно, если хотите выбрать другое отображение и см. пункт 1.

Иначе выполнение программы завершится и появится окно диалога с вопросом: “Exit Window?” «Закреть окно?». Можно ответить «нет», если нужно еще раз посмотреть последние графики. Окно программы можно закрыть позже. В случае положительного ответа окно программы закроется.

Приложение 2. Описание программы GlobalMap

Программа **GlobalMap** позволяет построить по скалярному временному ряду (записанному в текстовом файле данных) модельные отображения. Можно выбрать необходимые значения *параметров алгоритма реконструкции*:

- количество значений в тренировочном временном ряду N_{train} ,
- размерность модельного отображения D ,
- вид аппроксимирующей функции (стандартный полином или дробно-рациональная функция),
- порядок полинома K (если выбрана полиномиальная аппроксимация) или порядки полиномов в числителе и знаменателе (если выбрана аппроксимация дробно-рациональной функцией),

и *установок* для проверки эффективности модели:

- длину участка тестового временного ряда L_{test} ,
- количество этих участков K_{test} ,
- величину сдвига между соседними участками S_{test} .

Значения всех этих параметров можно менять в течение сеанса работы с программой. Значения всех параметров и установок алгоритма и основные результаты построения моделей выводятся в файл отчета report.txt, который создается в текущем каталоге (если файл с таким именем уже существовал до за-

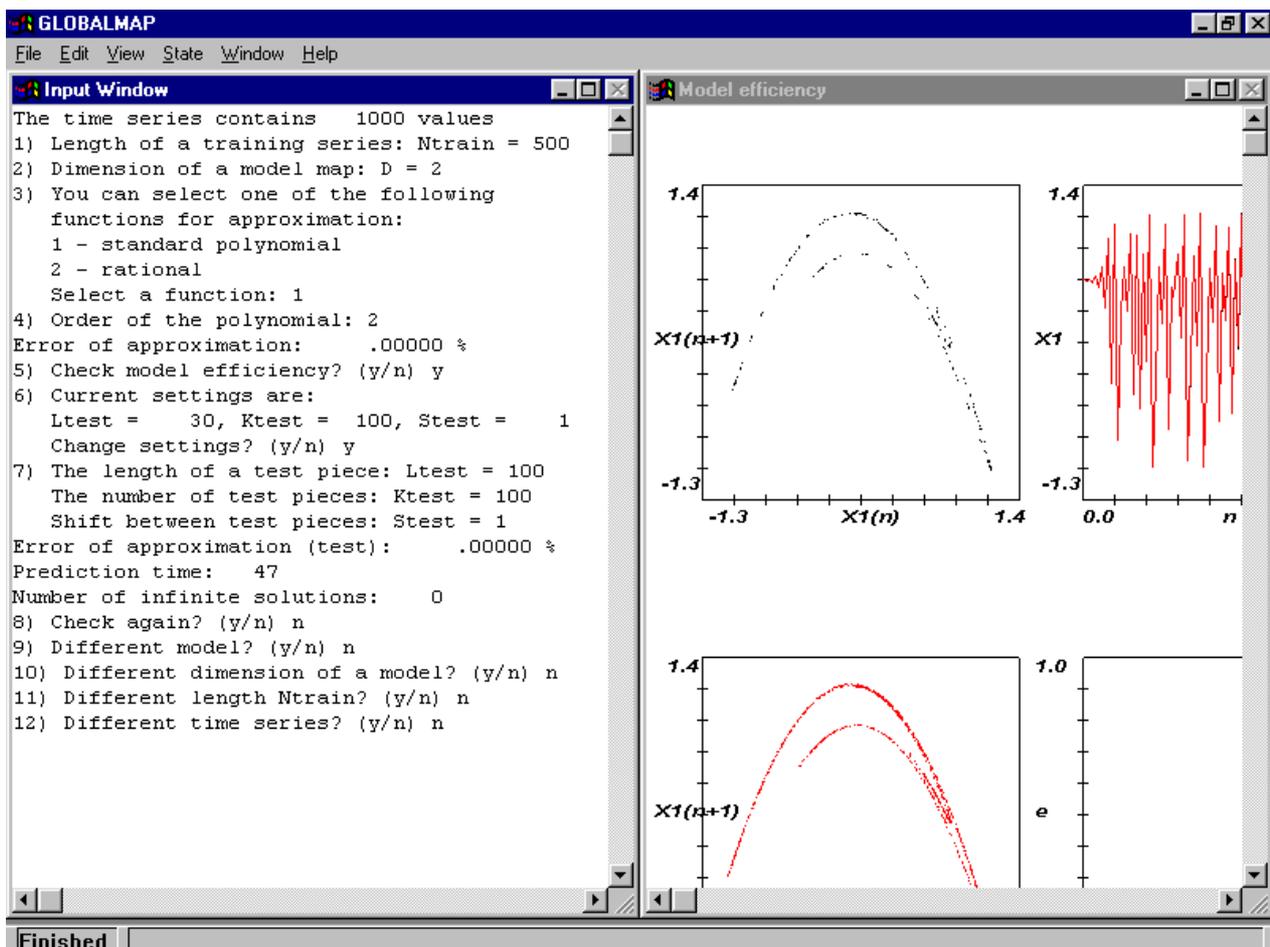


Рис.П.2.1. Внешний вид окна программы **GlobalMap** (данный сеанс работы с программой включал в себя восстановление модельных отображений по временной реализации координаты x отображения Эно при значениях параметров $a = 1.5$, $b = 0.2$).

пуска программы, то новые результаты добавляются к ранее записанной информации). Примеры внешнего вида окна программы и содержимого файла отчета представлены на рис.П.2.1 и рис.П.2.2.

Запустите файл **GlobalMap.exe**. Откроется окно программы и окно диалога для выбора файла данных. Откройте файл, который содержит скалярный временной ряд. Это должен быть текстовый файл (в ASCII-формате), в котором в столбец записаны последовательные значения наблюдаемой — вещественные числа (обязательно с разделяющей десятичной точкой).

В левой части обрамляющего окна открывается окно ввода (Input Window), которое остается открытым в течение всего сеанса работы с программой — это окно диалога пользователя с программой. Сначала программа считает количество значений наблюдаемой N в данном временном ряде и

```

герopt.txt - Блокнот
Файл  Правка  Поиск  ?
The number of values in a training time series Ntrain = 500

Dimension of a model map D = 2
Function for approximation: standard polynomial
Polynomial order nu = 2
Coefficients of the model equation:
    1.0000000000
    .2000000000
    .0000000000
    .0000000000
    .0000000000
    -1.5000000000
Error of approximation:      .00000 %

Error of approximation (test series):      .00000 %
Length of a test piece: Ltest = 100
Number of test pieces: Ktest = 100
Shift between test pieces: Stest = 1
Prediction time: 47
Number of infinite solutions: 0

!*****!

```

Рис.П.2.2. Содержимое файла отчета для сеанса работы, проиллюстрированного на рис.П.2.1.

выдает сообщение: “The time series contains ... values” («Временной ряд содержит ... значений»).

1) Появится приглашение задать длину тренировочного временного ряда (количество значений): “Length of a training time series: Ntrain = ” («Длина тренировочного временного ряда Ntrain = »). Если нужно будет проверять эффективность построенных моделей, то задайте длину тренировочного ряда меньше длины всего ряда, чтобы осталось необходимое количество данных и для тестирования. Значение N_{train} — целое число, для которого справедливо $M + D \leq N_{train} \leq N$, где M — количество коэффициентов предполагаемой модели.

2) Программа запросит размерность модельного отображения D : “Dimension of a model map: D = ” «Размерность модельного отображения D = ». Введите нужное значение (это должно быть положительное целое число).

3) Появится сообщение: “You can select one of the following functions for approximation: 1 — standard polynomial, 2 — rational. Select a function: ” («Вы

можете выбрать один из следующих видов функций для аппроксимации: 1 — стандартный полином, 2 — дробно-рациональная функция. Выберите вид функции: »). Введите номер, соответствующий выбранному виду функции.

4) Далее нужно задать порядок полинома, если выбрана полиномиальная аппроксимация: “Order of the polynomial: ” («Порядок полинома: »). Задайте целое число $K \geq 0$.

Если выбрана рациональная аппроксимация, то потребуется задать поочередно порядок каждого полинома (в числителе и в знаменателе): “Order of the numerator polynomial: . Order of the denominator polynomial: ” («Порядок полинома в числителе: . Порядок полинома в знаменателе: »).

После этого рассчитываются коэффициенты модельного отображения. В окне ввода появляется сообщение: “Error of approximation: ... %” («Погрешность аппроксимации: ... %»). Значения коэффициентов выводятся в файл отчета¹².

5) Появится вопрос: “Check model efficiency? (y/n) ” («Проверить эффективность модели? (да/нет)»). Введите символ “y”, если Вы хотите протестировать модель, и далее см. пункт 6. Иначе отвечайте “n” и далее см. пункт 9.

6) Далее появится сообщение о текущих установках для проверки эффективности модели: “Current settings are: Ltest = ..., Ktest = ..., Stest = ...” («Текущие установки: Ltest = ..., Ktest = ..., Stest = ... »), и вопрос: “ Change settings? (y/n) ” («Изменить установки? (да/нет)»). Дайте положительный ответ, если установки Вас не устраивают, и см. пункт 7. В противном случае ответьте “n” и см. пункт 8.

7) Программа предложит задать установки для тестирования модели: “The length of a test piece: Ltest = ” («Длина тестового участка: Ltest = »), “The

¹² Значения коэффициентов записываются в столбец в следующем порядке. Сначала записывается свободный член полинома. Затем — коэффициенты при слагаемых в первой степени: x_1, \dots, x_D . Затем — при слагаемых во второй степени: $x_1^2, x_1x_2, \dots, x_1x_D, x_2^2, x_2x_3, \dots, x_D^2$. И так далее с повышением степени слагаемых. Если использовалась дробно-рациональная функция, то сначала выводятся коэффициенты полинома в числителе, затем — полинома в знаменателе, причем свободный член полинома в знаменателе, равный 1, не выводится.

number of test pieces: $K_{test} =$ ” («Количество тестовых участков: $K_{test} =$ »), “Shift between test pieces: $S_{test} =$ ” («Сдвиг между тестовыми участками: $S_{test} =$ »). Введите нужные значения (положительные целые числа), причем нужно следить, чтобы для тестового ряда хватило данных, т.е. $N_{train} + N_{test} \leq N$.

8) Проводится тестирование модели. В правой части обрамляющего окна появится окно “Model efficiency” («Эффективность модели»). В нем иллюстрируются результаты тестирования.

Слева: сверху показана фазовая траектория объекта, восстановленная по тестовому временному ряду, ниже — рассчитанная фазовая траектория модели.

Справа: сверху представлены временные реализации объекта (черным цветом) и модели (красным) при одинаковых начальных условиях, ниже приведен график зависимости ошибки прогноза от времени.

В окне ввода распечатываются результаты тестирования — “Error of approximation (test): ... %”, “Prediction time: ...”, “Number of infinite solutions: ...” («Погрешность аппроксимации (по тестовому ряду): ... %», «Дальность прогноза: ...», «Количество неограниченных решений: ...»). Все результаты записываются также в файл отчета.

Появится вопрос: “Check again? (y/n)” («Проверить снова? (да/нет)»). Если Вы хотите проверить модель с другими установками, то отвечайте “y” и возвращайтесь к п. 7. Иначе отвечайте отрицательно и см. пункт 9.

9) Появится вопрос: “Different model? (y/n)” («Другую модель? (да/нет)»). Отвечайте положительно, если нужно построить модель, например, с другим порядком полинома, и см. п. 3. Иначе отвечайте “n” и см. пункт 10.

10) Появится очередной вопрос: “Different dimension of a model? (y/n)” («Другую величину размерности модели? (да/нет)»). В случае положительного ответа повторите моделирование при другом значении размерности модели — пункт 2, иначе — см. пункт 11.

11) Следующий вопрос: “Different length Ntrain? (y/n)” («Другую длину тренировочного ряда N_{train} ? (да/нет)»). Отвечайте “y”, если нужно провести

моделирование по тренировочному ряду большей или меньшей длины — см. пункт 1. Иначе переходите к пункту 12.

12) Далее последует вопрос: “ Different time series? (y/n) ” («Другой временной ряд? (да/нет)»). В случае положительного ответа Вам будет предложено выбрать новый файл данных для анализа (см. начало описания программы).

Иначе выполнение программы завершится и появится окно диалога с вопросом: “Exit Window?” («Закреть окна?»). Можно ответить «нет», если нужно еще раз просмотреть последние графики (окно программы можно закрыть позже). В случае положительного ответа окно программы закроется.

Приложение 3. Описание программы LLMar

Программа **LLMar** позволяет построить по скалярному временному ряду локальные линейные модели. Можно выбрать необходимые значения параметров алгоритма реконструкции:

- количество значений в тренировочном временном ряде,
- размерность модельного отображения,
- размер окрестности,

и установок для проверки эффективности модели:

- длина тестового участка L_{test} ,
- количество тестовых участков K_{test} .

Значения всех параметров и установок можно менять в течение сеанса работы с программой. Значения всех параметров и установок алгоритма и основные результаты построения моделей выводятся в файл отчета report.txt, который создается в текущем каталоге (если файл с таким именем уже существовал до запуска программы, то новые результаты добавляются к ранее записанной информации). Примеры внешнего вида окна программы и содержимого файла отчета представлены на рис.П3.1 и рис.П3.2.

Запустите файл **LLMar.exe**. Откроется окно программы и окно диалога для выбора файла данных. Откройте файл, который содержит скалярный вре-

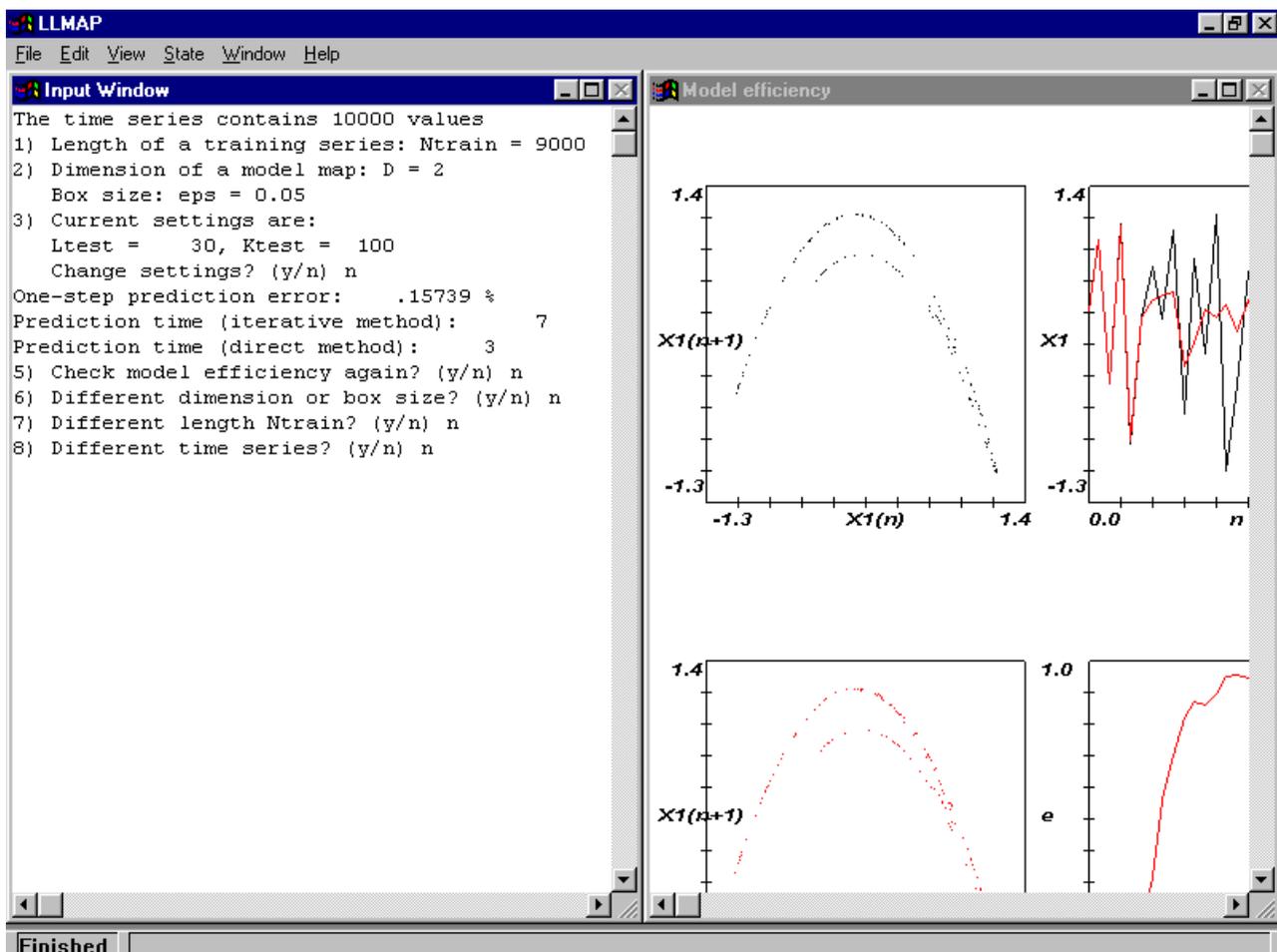


Рис.П.3.1. Внешний вид окна программы **LLMap** (данный сеанс работы с программой включал в себя восстановление модельных отображений по временной реализации координаты x отображения Эно при значениях параметров $a = 1.5$, $b = 0.2$).

менной ряд. Это должен быть текстовый файл (в ASCII-формате), в котором в столбец записаны последовательные значения наблюдаемой — обязательно вещественные числа (с разделяющей десятичной точкой).

В левой части обрамляющего окна открывается окно ввода (Input Window), которое остается открытым в течение всего сеанса работы с программой — это окно диалога пользователя с программой. Сначала программа считает количество значений наблюдаемой N в данном временном ряде и выдает сообщение: “The time series contains ... values” («Временной ряд содержит ... значений»).

1) Предлагается задать длину тренировочного временного ряда (количество значений): “Length of a training series: Ntrain = ” («Длина тренировочного ряда Ntrain = »). Задайте длину тренировочного ряда меньше длины всего ряда,

```

report.txt - Блокнот
Файл  Правка  Поиск  ?

The number of values in a training time series: Ntrain = 9000

The number of values in a training time series: Ntrain = 900

Dimension of a model map: D = 2
Box size: eps = .10000
The number of values in a training time series: Ntrain = 9000

Dimension of a model map: D = 2
Box size: eps = .05000
Length of a test piece: Ltest = 30
Number of test pieces: Ktest = 100
One-step prediction error: .15739 %
Prediction time (iterative method): 7
Prediction time (direct method): 3

!*****!

```

Рис.П.3.2. Содержимое файла отчета для сеанса работы, проиллюстрированного на рис.П.3.1.

чтобы осталось необходимое количество данных для тестирования. Значение N_{train} — целое число, для которого справедливо $1 \ll N_{train} \ll N - D$.

2) Программа запросит значения размерности модели D и размера окрестности ε_{box} : “Dimension of a model map: D = ” («Размерность модели D = »), “Box size: eps = ” («Размер окрестности: eps = »). Для размерности введите положительное целое число, а для размера окрестности — вещественное число (обязательно с разделяющей десятичной точкой).

Тренировочный временной ряд будет отсортирован.

3) Появится сообщение: “Current settings are: Ltest = ..., Ktest = ...” («Текущие установки: Ltest = ..., Ktest = ...») и вопрос: “Change settings? (y/n)” («Изменить установки? (да/нет)»). Если установки для тестирования модели Вас устраивают, то отвечайте отрицательно (введите символ “n”) и см. п.5. Иначе отвечайте “y” и см. п. 4.

4) Программа предложит задать установки: “The length of a test piece: Ltest = ”, “The number of test pieces: Ktest = ” («Длина тестового участка: Ltest = », «Число тестовых участков: Ktest = »).

«Количество тестовых участков: $K_{test} = \dots$ »¹³. Введите нужные значения: положительные целые числа; помните, что должно выполняться $N_{train} + N_{test} \leq N$.

5) Рассчитывается дальность прогноза модели (прямым и итерационным методами) и фазовый портрет модели (итерационным методом). Результаты представляются графически в окне “Model efficiency” («Эффективность модели»), которое открывается в правой части окна программы.

Слева: сверху показана фазовая траектория объекта (восстановленная по тестовому временному ряду), ниже — фазовая траектория модели.

В центре: сверху показана временная реализация объекта (черным цветом) и прогноз с помощью прямого метода (красным), ниже — ошибка прогноза с помощью прямого метода.

Справа: сверху представлена та же временная реализация объекта (черным цветом) и прогноз с помощью итерационного метода (красным), ниже — ошибка прогноза с помощью итерационного метода.

В окне ввода появляются сообщения: “One-step prediction error: ... %”, “Prediction time (iterative method): ...”, “Prediction time (direct method): ...” («Ошибка прогноза на один шаг вперед: ... %», «Дальность прогноза (прямой метод): ...», «Дальность прогноза (итерационный метод): ...»). Результаты выводятся также в файл отчета. Количество соседей, найденных для каждого из K_{test} векторов тестового временного ряда, выводится в столбец в файл neighbors.txt в текущем каталоге.

Появляется вопрос: “Check model efficiency again? (y/n)” («Проверить эффективность модели снова? (да/нет)»). Если Вы хотите проверить эффективность модели при других установках, отвечайте “y” и см. п.3. Иначе отвечайте “n” и см. п. 6.

6) Следующий вопрос: “Different dimension or box size? (y/n)” («Другие значения размерности модели или размера окрестности? (да/нет)»). Отвечайте

¹³ Сдвиг между тестовыми участками принимается равным $S_{test} = 1$.

положительно, если нужно построить модель другой размерности или с другим радиусом окрестности — см. пункт 2, иначе отвечайте “n” и см. пункт 7.

7) Очередной вопрос: “Different length Ntrain? (y/n)” («Другой размер тренировочного ряда Ntrain? (да/нет)»). Отвечайте «да», если требуется повторить моделирование при другой длине тренировочного временного ряда — см. пункт 1, иначе отвечайте отрицательно и см. пункт 8.

8) Вопрос: “Different time series? (y/n)” («Другой временной ряд? (да/нет)»). Отвечайте “y”, если нужно выбрать другой файл данных — вновь откроется окно диалога для выбора файла (возвращайтесь к началу описания программы).

В случае отрицательного ответа выполнение программы завершится и появится окно диалога с вопросом: “Exit Window?” «Закреть окно?». Можно ответить «нет», если нужно еще раз просмотреть последние графики. Окно программы можно закрыть позже. В случае положительного ответа окно программы закроется.

Литература

1. Бокс Дж., Дженкинс Т. “Анализ временных рядов. Прогноз и управление”, М.: Мир, 1974, 242 с.
2. Crutchfield J.P., McNamara B.S. “Equations of motion from a data series”, Complex Systems, 1987, Vol. 1, P. 417-452.
3. Cremers J., Hubler A. “Construction of differential equations from experimental data”, Z.Naturforschung A, 1987, Vol. 42, P. 797-802.
4. Farmer J.D., Sidorowich J.J. “Predicting chaotic time series”, Phys.Rev.Lett., 1987, Vol. 59, P. 845-848.
5. Packard N.H., Crutchfield J.P., Farmer J.D., Shaw R.S. “Geometry from a time series”, Phys.Rev.Lett., 1980, Vol. 45, № 9, P. 712-716.
6. Takens F. “Detecting strange attractors in turbulence”, in Dynamical Systems and Turbulence, Warwick, 1980, eds. D.Rang and L.S.Young, Lecture Notes in

Mathematics, Vol. 898, P. 366-381.

7. Casdagli M. "Nonlinear prediction of chaotic time series", *Physica D*, 1989, Vol. 35, P.335-356.
8. Breeden J.L., Hubler A. "Reconstructing equations of motion from experimental data with unobserved variables", *Phys.Rev. A*, 1990, Vol. 42, № 10, P. 5817-5826.
9. Judd K., Mees A. "On selecting models for nonlinear time series", *Physica D*, 1995, Vol. 82, P. 426-444.
10. Аносов О.Л., Бутковский О.Я., Кравцов Ю.А. "Минимаксная процедура идентификации хаотических систем по наблюдаемой временной последовательности", *Радиотехника и электроника*, 1997, Т. 42, В. 3, С. 313-319.
11. Brown R., Rulkov N.F., Tracy E.R. "Modeling and synchronizing chaotic systems from time-series data", *Phys.Rev. E*, 1994, Vol. 49, № 5, P. 3784-3800.
12. Янсон Н.Б., Анищенко В.С. "Моделирование динамических систем по экспериментальным данным", *Изв. ВУЗов. Прикладная нелинейная динамика*, 1995, Т. 3, № 3, С.112-121.
13. Грибков Д.А., Грибкова В.В., Кравцов Ю.А., Кузнецов Ю.И., Ржанов А.Г. "Восстановление структуры динамической системы по временным рядам", *Радиотехника и электроника*, 1994, Т. 39, В. 2, С. 269-277.
14. Кравцов Ю.А. "Случайность, детерминированность, предсказуемость", *Успехи физ. наук*, 1989, Т. 158, № 1, С. 93-115.

Учебно-методическое пособие

БЕЗРУЧКО Борис Петрович
СМИРНОВ Дмитрий Алексеевич

**Построение модельных отображений
по хаотическим временным рядам**

ГосУНЦ «Колледж», Лицензия ЛР №020773 от 15.05.98

Подписано к печати 13.04.2000. Формат 60x84 1/16.
Бумага Papirus Slim. Гарнитура Times.
Усл. печ. л. 1,39 (1,5). Уч.-изд. л. 1.1. Тираж 100 экз. Заказ 161.

Издательство ГосУНЦ «Колледж»
410026, Саратов, ул. Астраханская, 83.
Тел. (845-2) 523864



Отпечатано на ризографе издательства ГосУНЦ «Колледж»