Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
Базовая кафедра интеллектуальных систем управления

УТВЕРЖДАЮ
Заведующий кафедрой
_____ Ю. Ю. Якунин
«13» июня 2018 г.

БАКАЛАВРСКАЯ РАБОТА

27.03.03 «Системный анализ и управление»

Анализ заполнения пропусков в данных ZET-алгоритмом при решении задачи идентификации

Руководитель _____ _____ А. А. Корнеева
подпись, дата должность, ученая степень

Выпускник _____ Е. А. Пермякова
подпись, дата

Красноярск 2018

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

При управлении различного рода процессами (технологическими, социальными, экономическими и т.д.) важную роль играет моделирование и идентификация. Модели позволяют выявить наиболее важные свойства и закономерности исследуемого процесса, что довольно сложно сделать на реальном объекте.

При моделировании большое значение имеет как априорная информация об объекте исследования, так и текущая. Текущая информация может содержать в себе ряд недостатков, в частности, пропуски (пробелы) в данных. Пробелы в данных приводят к понижению точности моделей исследуемого объекта.

Проблема пропущенных значений достаточно актуальна. Неполнота данных приводит к ошибкам работы программ, неполноте информации об исследуемом объекте. Такая неполнота может образоваться в связи с отказом датчика, ошибке респондента, ошибке в анкете и т.д.

При небольшом количестве пропусков строку, содержащую пробелы, можно не учитывать при решении задачи моделирования. Однако, когда пропусков достаточно много, данная процедура не может быть применена, поскольку большие объемы информации будут потеряны. Именно для таких случаев и предназначены методы восстановления данных с пропусками.

На сегодняшний день существует достаточно большое количество методов заполнения пропусков в данных (заполнение по среднему, разновидности ZET алгоритмов и других алгоритмов «Ванга», и другие). Существующие алгоритмы заполнения пропусков зачастую предполагают знания основных статистических характеристик выборки и достаточно большого объема априорной информации. В связи с этим предлагается непараметрическая методика заполнения пропусков в данных. Пропуски при этом могут располагаться хаотично, вплоть до нескольких пропусков в одной

строке. Учитывая выше сказанное, данная проблема является достаточно актуальной.

Целью данной дипломной работы является сравнение предлагаемого непараметрического алгоритма с ZET алгоритмом и их анализ. Для этого необходимо решить следующие задачи:

- реализовать алгоритмы обработки неполных данных на языке программирования C#;
- провести исследование каждого алгоритма;
- сравнить результаты исследования алгоритмов между собой;
- сделать выводы о проделанной работе.

В данной работе используются такие методы, как: математическое моделирование, теория идентификации, анализ данных и математическая статистика.

Объектом данной работы является решение задачи идентификации по выборкам, содержащим пропуски. Предметом работы являются алгоритмы обработки неполных данных.

1 Анализ данных при решении задач идентификации

1.1 Постановка задачи идентификации

Теория идентификации занимается построением математических моделей исследуемых процессов по результатам наблюдений за входными и выходными переменными. Работы в данной области ведутся с середины 20-го века и представлены в большом количестве работ [4, 5, 8, 26, 28 и др.].

Математическая модель объекта или процесса – это его количественная формализация [4], [8]. Под математической моделью понимают формальное описание объекта при помощи математических способов: дифференциальных, интегро-дифференциальных, интегральных, алгебраических, разностных уравнений и множеств, неравенств и т.д. Крайне существенный ответ на вопрос выбора оператора объекта, который превращает входные параметры, действующие на объект, в выходные параметры. Иногда представляется возможным использовать фундаментальные законы, лежащие в основе функционирования химических, физических, биологических и других процессов Законы, которые имеют место для механических, электромагнитных, электрических, термодинамических, электрохимических, гидравлических, биофизических и иных процессов. К сожалению, мы довольно часто сталкиваемся с существенными трудностями в этом направлении, в частности в ходе изучения социальных, технологических и других процессов, иными словами сталкиваемся с неполной информацией об исследуемом процессе. И степень неполноты информации может быть очень различной. По этому вопросу Л. Бриллюэном было отмечено, что «всякая плодотворная гипотеза кладет начало удивительному извержению потока непредвиденных открытий», но если посмотреть с противоположной стороны «физические модели отличаются от мира так же, как географическая карта от поверхности Земли».
Общая схема процесса идентификации объекта представлена на рисунке 1.1.1:

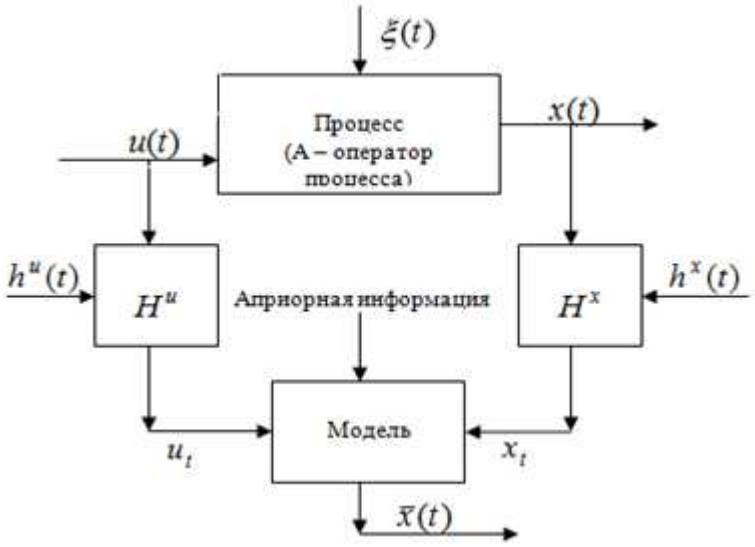

Рисунок 1.1.1 – Априорная информация

Приняты обозначения: A – оператор процесса, H – каналы измерения соответствующих переменных процесса, $u(t)$ – k-мерный вектор контролируемых входных переменных, $x(t)$ – n-мерный вектор выходных переменных, $h^u(t), h^x(t)$ – случайные помехи при измерении соответствующих переменных, $u^h[t],\ x^h[t] -$ наблюдения переменных в дискретные моменты времени t через соответствующие интервалы времени $\Delta t, \xi(t)$ – случайные помехи, действующие на процесс, $\bar{x}(t) -$ выход модели. В блоке «Модель» определен принятый класс моделей на основании априорной информации об исследуемом процессе и анализе текущей информации $\{u^h[t], x^h[t], t = \overline{1,s}\}$, здесь же осуществляется настройка (обучение) модели принятого класса. В дальнейшем, обучающую выборку $\{u^h[t], x^h[t], t = \overline{1,s}\}$, из соображений простоты записи, будем обозначать $\{u[t], h[t], t = \overline{1,s}\}$.

Как мы выяснили, математическое представление процесса, который мы исследуем, может быть представлено в следующем виде

$$x(t) = A\langle u(t),\ \xi(t), t\rangle, \tag{1.1.1}$$

а математическая модель данного процесса в форме

$$\bar{x}[t+1] = \bar{A}\langle u[t+1], [t]\rangle, \qquad (1.1.2)$$

где $\bar{A}$ – оператор модели, кроме того все случайные факторы, действующие в каналах измерения и на процесс, имеют нулевые математические ожидания и ограниченные дисперсии.

Система может быть описана как безынерционная система с запаздыванием

$$x(t) = f\big(u(t-\tau), \xi(t)\big), \qquad (1.1.3)$$

где τ – запаздывание, которое может отличаться по различным каналам связи. Модель системы, описываемой (1.1.3), имеет вид

$$\hat{x}(t) = f\big(u(t-\tau)\big), \qquad (1.1.4)$$

Помимо перечисленного на функциональный блок «Модель» поступает также априорная информация о процессе. От полноты данной информации во многом зависит и точность решения задачи идентификации. Далее рассмотрим различия между априорной и текущей информацией.

Большую роль при изучении того или иного процесса играют средства контроля его «входных – выходных» переменных [16, 19]. Рассмотрим следующую схему на рисунке 1.1.2:

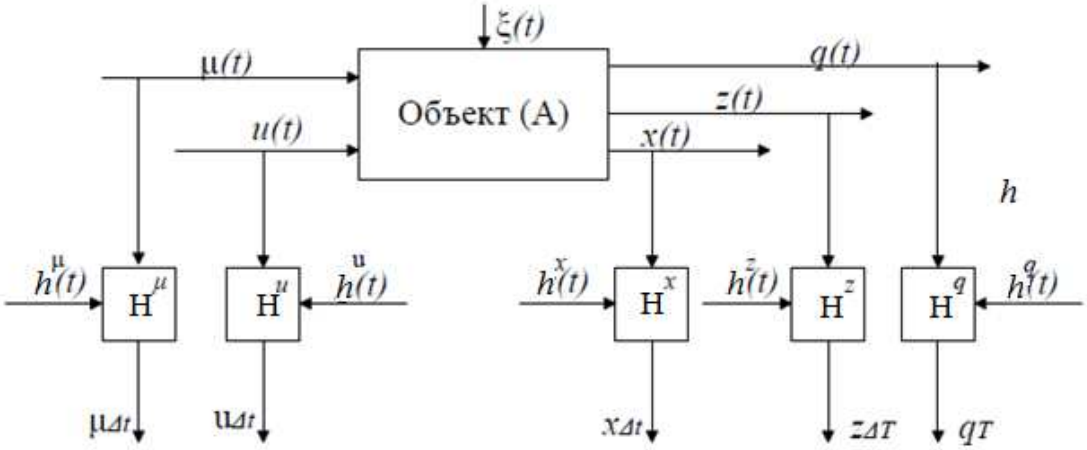

Рисунок 1.1.2 – Контроль «входных – выходных» переменных процесса

Здесь $\mu(t) = (\mu_1(t), \mu_2(t), \dots, \mu_r(t)) \in \Omega(\mu) \in R^r$ – это входная измеряемая, но неконтролируемая переменная процесса. К примеру, если u(t) – это загрузка подаваемого на вход материала, то $\mu(t)$ может быть физико-химической или технологической характеристикой этого материала, существенно влияющая на процесс, который протекает в объекте, то есть на выход. Переменные $z(t) = (z_1(t), z_2(t), \dots, z_r(t)) \in \Omega(z) \in R^l$ и $q(t) = (q_1(t), q(t), \dots, q_r(t)) \in \Omega(q) \in R^v$ – это векторные выходные переменные процесса. Отметим существенное различие между выходными переменными $x(t)$, $z(t)$ и $q(t)$. Выходные переменные данного процесса контролируются с различной дискретностью: переменная $x(t)$ измеряется через интервал времени Δt, переменная $z(t)$ измеряется через больший интервал времени ΔT, а переменная $q(t)$ – через интервал времени T, причем выполняется следующее условие $\Delta t << \Delta T << T$. Различие в дискретности измерения переменных процесса в данном случае обусловлено отличием в способах их контроля. К примеру, одни величины могут быть измерены электрическими средствами. Подобный контроль не требует больших временных затрат. Здесь мы сами можем задать интересующую нас дискретность измерения. Измерения других переменных могут быть получены лишь с помощью лабораторного анализа, или же путем физико-механических, физико-химических и другого рода испытаний, что требует значительно большего времени. Следует отметить, что чаще всего переменная с наибольшей дискретностью контроля $q(t)$ является наиболее важной, то есть она определяет качество выпускаемой продукции. Но использовать ее в целях управления в режиме реального времени невозможно. Поэтому задача ее прогнозирования является актуальной.

В этом случае выходная переменная $x(t)$, как и ранее, может быть описана уравнением (1.1.4). Для описания переменных с большей дискретностью измерения $z(t)$ и $q(t)$ целесообразно использовать весь набор переменных, влияющих на их поведение

$$\hat{z}(t) = A\big(u(t-\tau), \mu(t), \hat{x}(t)\big), \tag{1.1.5}$$

$$\hat{q}(t) = A\big(u(t-\tau), \mu(t), \hat{x}(t), \hat{z}(t)\big). \tag{1.1.6}$$

В качестве примера статической системы с запаздыванием рассмотрим процесс помола клинкера в шаровых трехкамерных мельницах сухого помола [55]. Процесс измельчения какого-либо конкретного продукта является сравнительно типичным для многих отраслей промышленности. Клинкер – это промежуточный продукт при производстве цемента, представляет собой гранулы, полученные в результате обжига сырьевой смеси, измельчение которых приводит к получению цемента. Рассмотрим рисунок 1.1.3, на котором схематично представлена шаровая трехкамерная мельница сухого помола.

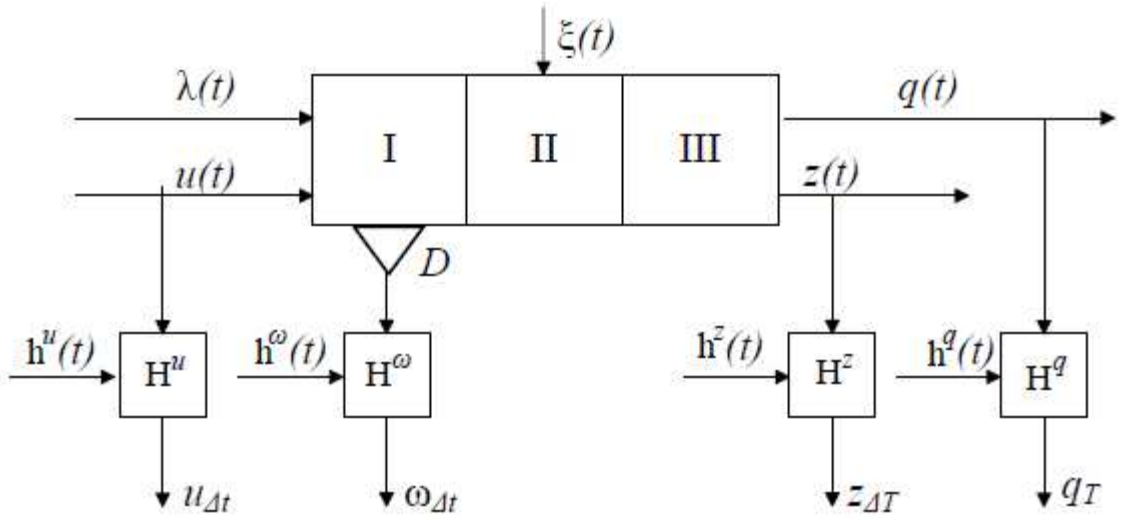

Рисунок 1.1.3 – Шаровая трехкамерная мельница сухого помола

Мельница сухого помола представляет собой цилиндрический вращающийся барабан, разделенный сеточными перегородками на три камеры, загруженными мелющими телами (в I камере достаточно крупные металлические шары, во II камере шары меньшего размера, в III камере цильберс – металлические цилиндры небольшого размера). Клинкер, поступающий в мельницу, измельчается в I, II, III камерах и превращается в цемент. Таким образом, с технологической точки зрения, входом мельницы

является загрузка клинкера, а выходом – цемент. Приняты следующие обозначения: $\lambda(t)-$ неконтролируемая входная переменная (размалываемость клинкера), $u(t)-$ контролируемая со случайной ошибкой входная переменная (загрузка/количество/клинкера); $w(t)-$ шум в первой камере, контролируемый индукционным датчиком D через интервал Δt, который в системах регулирования используется как выходной сигнал процесса измельчения; $z(t)-$ выход мельницы (тонкость измельчения), измеряемый через интервал времени $\Delta T \gg \Delta t;\ q(t)-$ основной показатель качества цемента (активность, прочность цементной балки при сжатии), контролируемый через $T \gg \Delta T \gg \Delta t$.

Постоянная времени объекта примерно 5-7 минут. Переменные u(t) и w(t) в локальных аналоговых системах регулирования контролируется непрерывно, а в цифровых системах регулирования дискретно через интервал Δt (Δt может измеряться через несколько секунд). Контроль выходных переменных $q(t), z(t)$ осуществляется в лаборатории по технологии, регламентируемой ГОСТом, причем $\Delta T = 2$ часа, а $T = 28$ суток. Как видно, это значительно превышает постоянную времени объекта. Отметим, что $q(t)-$ технологический показатель собственно процесса измельчения, а $z(t)-$ основной показатель качества (марки) цемента, который зависит не только от тонкости измельчения $q(t)$, но и от показателей работы предыдущих технологических переделов: приготовления сырьевой смеси, помол, обжиг.

Из сказанного становится ясно, что при управлении процессом помола учесть динамику процесса невозможно. Связано это с длительным контролем выходных переменных процесса. Подобные системы представляют как статические системы с запаздыванием.

1.2 Априорная и текущая информация

Априорная информация. Для математической формулировки задачи идентификации необходима априорная информация об объекте исследования, которая складывается из информации об его операторе, случайных помехах,

критерии оптимальности и ограничениях. Критерий оптимальности выражает собой те требования, которые должны быть наилучшим образом удовлетворены, а ограничения определяют наши возможности.

Априорная информация может базироваться на фундаментальных законах, лежащих в основе разнообразных физических, механических, электротехнических, химических, биофизических и других процессов и объектов или результатов предшествующих исследований интересующих нас объектов [9]. Априорные сведения об объекте не возобновляются, и со временем могут утратить свое значение (старение оборудования, различного рода случайные изменения и др.).

Следует отличать априорную информацию от текущей (апостериорной) информации. Текущая информация извлекается в результате наблюдений за ходом процесса или в результате экспериментов и представляет собой выборки наблюдений «входных – выходных» переменных процесса вида $\{u_i, x_i, i = \overline{1,s}\}$. Текущая информация обновляется в каждый момент времени. Она может использоваться для накопления соответствующей априорной информации, но наиболее важная ее роль – компенсация недостаточного объема априорной информации. Следует вспомнить высказывание Я.З.Цыпкина из [26]: «Априорная информация – это основа для формулировки проблемы оптимальности. Текущая информация – средство решения этой проблемы».

Полная априорная информация о процессе предполагает абсолютно точное знание, которого никогда нет. Кроме того, каждый процесс подвержен воздействию множества случайных факторов. Вследствие этого, все случаи, с которыми мы сталкиваемся, соответствуют неполной априорной информации.

Выделим некоторые уровни априорной информации:

- байесов уровень априорной информации. С точностью до параметров известны: параметрическая модель исследуемого объекта, законы распределения случайных помех и уравнения каналов связи. Необходимо оценить параметры параметрической модели объекта;

- уровень параметрической неопределенности. Параметрическая модель объекта исследования известна с точностью до параметров, которые необходимо оценить. Известны некоторые характеристики случайных помех. Решается задача идентификации в «узком» смысле;

- уровень непараметрической неопределенности. На этом уровне априорной информации отсутствует этап определения параметрической структуры исследуемого объекта, поэтому требования к уровню априорной информации ослабевают, но здесь требуется информация качественного характера (однозначность или неоднозначность характеристик, линейность процесса либо характер его нелинейности и др.). Для решения задачи идентификации в этом случае применяют методы непараметрической статистики [11];

- уровень параметрической и непараметрической неопределенности. Это случай, когда задача идентификации многосвязной системы формулируется в условиях и параметрической, и непараметрической априорной информации. Модели здесь представляют собой взаимосвязанную систему параметрических и непараметрических соотношений.

1.3 Методы параметрической идентификации

Задача параметрической идентификации заключается в поиске параметров и структуры системы исходя из наблюдений [89]. На основании наблюдаемых входных воздействий и выходных параметров объекта исследования выполняется поиск параметров налаживаемой модели, обеспечивающий экстремум определенного критерия, который характеризует непосредственно качество процедуры идентификации. Следует учесть предположения об известности структуры объекта исследования с точностью до всех необходимых входных параметров.

Исследуемому этапу, на котором непосредственно выполняется настройка параметров параметрической модельной структуры объекта, посвящено огромное количество научных исследований [5, 26, 28 и др.], но, к

сожалению, этап выбора самой структуры модели в значительной степени обделен таким вниманием. Но в случае параметрического подхода непосредственно от выбора способа математического описания исследуемого объекта в целом и зависит, что будет результатом решения задачи идентификации.

Общепринятая структурная схема процесса параметрической идентификации изображена на рисунке 1.3.1 [26]. В данном случае, аналогично вышеприведенным схемам, u_t и $x_t -$ значения выходных и входных переменных процесса, которые измеряются в дискретные промежутки времени t; $\xi_t -$ наблюдаемая случайная помеха; $x_{st} -$ выходные значения настраиваемой модели; $\alpha -$ вектор параметров настраиваемой модели; $I_s -$ вектор всех имеющихся к моменту времени t наблюдений; $\varepsilon_t = x_t - x_{st} -$ ошибка рассогласования; $Q(\varepsilon) -$ выпуклая функция потерь; $M -$ символ математического ожидания; $R(\alpha) -$ критерий идентификации.

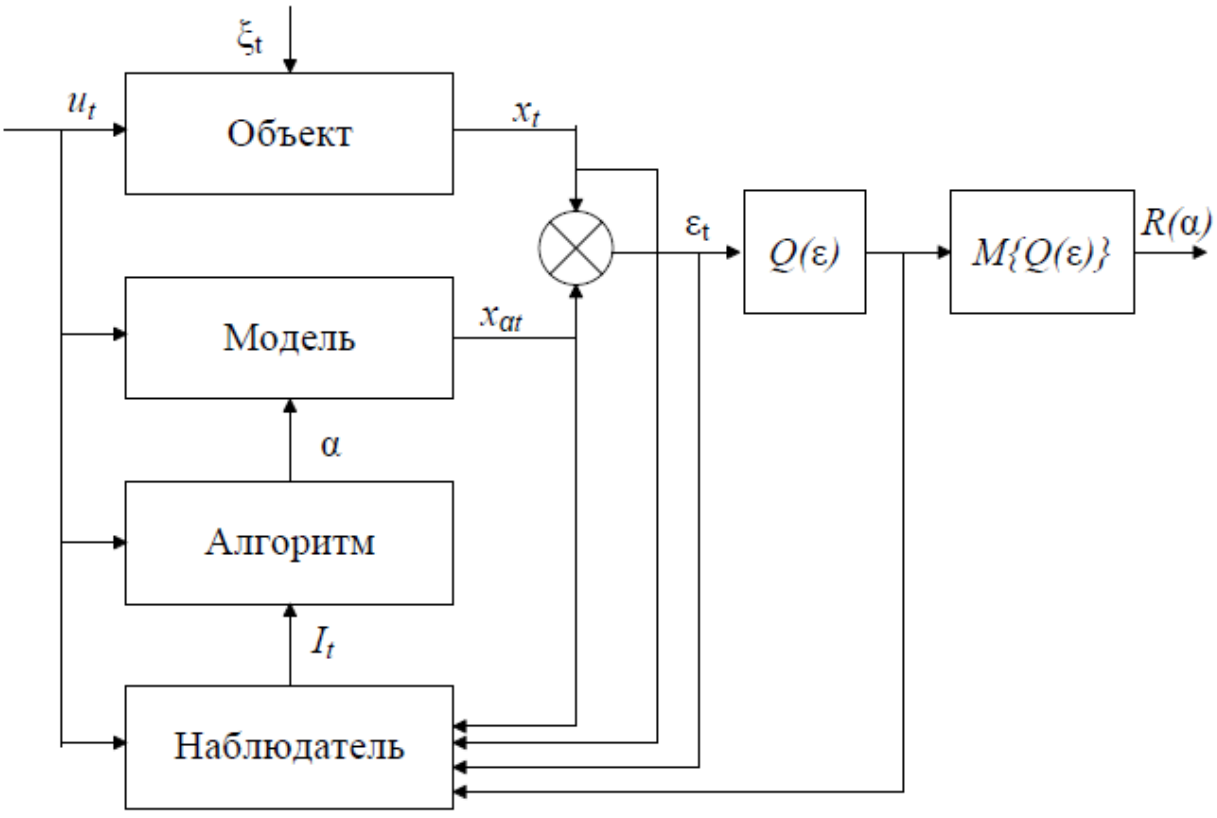

Рисунок 1.3.1 – Схема задачи идентификации

Идентификация по приведенной схеме производится с использованием настраиваемой модели, которая задается параметрической структурой (блок «Модель»). Параметры модели α изменяются в зависимости от поступающих в блок «Алгоритм» наблюдений I_s от блока «Наблюдатель».

Степень тождественности настраиваемой модели исследуемому объекту определяется критерием называемым критерий качества идентификации

$$R(\alpha) = M\{Q(\varepsilon(x_t, x_{st}))\} \quad (1.3.1)$$

Блок «Алгоритм» имеет в своем составе специальный алгоритм идентификации, позволяющий произвести оценку параметров модели α. Его единственной и прямой задачей является максимальная минимизация критерия качества идентификации

$$R(\alpha^*) = \min_{\alpha} R(\alpha) \quad (1.3.2)$$

С целью настройки имеющихся у модели параметров применяют разные рекуррентные или итеративные методы. В частном случае, когда функционал (1.3.2) дифференцируем, логично, что он достигает своего экстремума тогда и только тогда значения$\alpha = (\alpha_1, \dots, \alpha_k)$, и их частные k производных $\frac{\partial R}{\partial \alpha_j}, j = \overline{1,k}$ одномоментно равны нулю, т.е.

$$\frac{\partial R}{\partial \alpha_j} = 0, j = \overline{1,k} \quad (1.3.3)$$

Основной идей указанного метода решения (1.3.2) с использованием итеративных регулярных методов является следующее [25]. Представим уже рассматриваемое выше уравнение (1.3.3) в форме

$$\alpha = \alpha - \gamma \nabla R(\alpha), \quad (1.3.4)$$

где γ – некоторый множитель и начнем поиск оптимального вектора с использованием последовательных приближений

$$\alpha[n] = \alpha[n-1] - \gamma[n]\nabla_{\alpha}R(\alpha[n-1]) \qquad (1.3.5)$$

Методы, которые основаны на использовании этого алгоритма (1.3.5) для поиска α^* и получили название регулярных итеративных методов.

Помимо всего перечисленного, для поиска неизвестных параметров применяют методы наименьших квадратов, стохастических аппроксимаций и др.

1.4 Методы непараметрической идентификации

Как уже говорилось ранее, методам параметрической идентификации необходим огромный объем априорной информации, что определяется требованиями параметрической структуры модели исследуемого объекта. Но зачастую на практике собрать нужное количество априорной информации об исследуемом объекте весьма проблематично или вообще невозможно, из-за чего структуру модели исследуемого объекта не представляется возможным определить с необходимой точностью. Цитируя П.Эйкхоффа: «Значение структуры нельзя переоценить. Ее выбор определяется типом применения модели и может оказаться решающим фактором успеха или неудачи принятой схемы оценивания» [28]. В таких условиях, когда априорной информации недостаточно, абсолютно целесообразно применять методы непараметрической идентификации [15, 27].

Методы непараметрической идентификации не требовательны к наличию информации о параметрической структуре модели объекта исследования, но в данном случае необходимо решать значительное количество дополнительно возникающих задач. Таких, как задание класса моделей и выбор структуры системы, оценивание степени линейности и стационарности исследуемого

объекта, а также имеющихся переменных, выбор информативных переменных, оценка формы и степени влияния входных переменных на выходные и др. [28].

Для построения моделей в условиях непараметрической неопределенности [2, 14, 17] используется непараметрическая оценка кривой регрессии, которая в многомерном случае будет иметь вид [24]

$$x_s(u) = \frac{\sum_{i=1}^{s} x_i \prod_{j=1}^{m} \Phi(\frac{u^j - u_i^j}{c_s})}{\sum_{i=1}^{s} \prod_{j=1}^{m} \Phi(\frac{u^j - u_i^j}{c_s})}, \qquad (1.4.1)$$

где $u = (u_1, u_2, \dots, u_m) - m$-мерный вектор входных воздействий объекта;

$x -$ выходная величина;

$\Phi\left(\frac{u^j - u_i^j}{c_s}\right) -$ ядерная колоколообразная функция;

$c_s -$ коэффициент размытости ядра.

Ядерная функция [29] и коэффициент размытости ядра c_s удовлетворяют следующим условиям сходимости [20]

$$c_s > 0; \qquad \Phi\left(\frac{u^j - u_i^j}{c_s}\right) < \infty;$$

$$\lim_{s\to\infty} c_s = 0; \qquad c_s^{-1} \int_{\Omega(u)} \Phi\left(\frac{u^j - u_i^j}{c_s}\right) dx = 1; \qquad (1.4.2)$$

$$\lim_{s\to\infty} s c_s^m = \infty; \qquad \lim_{s\to\infty} c_s^{-1} \Phi\left(\frac{u^j - u_i^j}{c_s}\right) = \delta(u - u_i);$$

где $\delta(u - u_i) -$ дельта – функция Дирака.

Могут быть использованы различные формы ядерных функций, приведем некоторые из них:

треугольное ядро

$$\Phi\left(\frac{x - x_i}{c_s}\right) = \begin{cases} 1 - |c_s^{-1}(x - x_i)|, |c_s^{-1}(x - x_i)| \le 1; \\ 0, |c_s^{-1}(x - x_i)| > 1; \end{cases} \qquad (1.4.3)$$

параболическое ядро

$$\Phi\left(\frac{x-x_i}{c_s}\right) = \begin{cases} 0.75(1-\left(c_s^{-1}(x-x_i)\right)^2), |c_s^{-1}(x-x_i)| \le 1; \\ 0, |c_s^{-1}(x-x_i)| > 1; \end{cases} \quad (1.4.4)$$

кубическое ядро

$$\Phi\left(\frac{x-x_i}{c_s}\right) = \begin{cases} (1+2|c_s^{-1}(x-x_i)|)(1-\left(c_s^{-1}(x-x_i)\right)^2), |c_s^{-1}(x-x_i)| \le 1; \\ 0, |c_s^{-1}(x-x_i)| > 1; \end{cases} \quad (1.4.5)$$

Графическая интерпретация этих ядер представлена на рисунке 1.4.1:

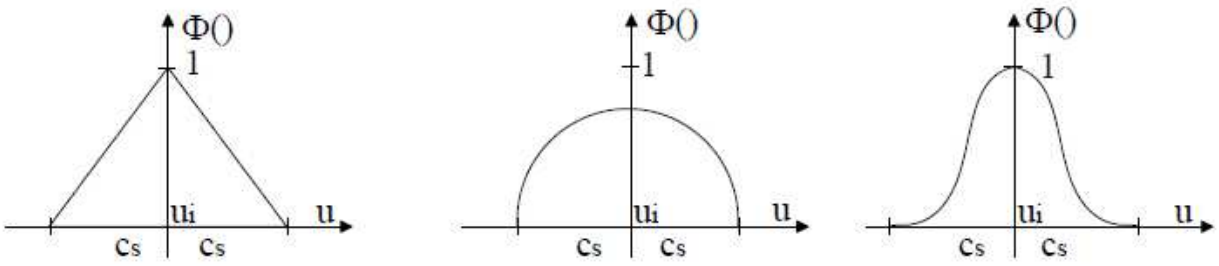

Рисунок 1.4.1 – Виды ядерных функций

В дальнейшем, для простоты, обозначается ядерная функция как $\Phi(\cdot)$. Выбор формы ядра не играет существенной роли, исследователь выбирает $\Phi(\cdot)$ исходя из практический соображений. Точность восстановления функции регрессии по наблюдениям с ошибками не существенно зависит от формы ядра и диктуется, чаще всего, практическими соображениями.

Треугольное, параболическое и кубическое ядро относятся к виду усеченных ядер. Также выбор ядерной функции может быть продиктован дополнительными условиями, к примеру, требованиями дифференцирования.

Большее влияние на функцию качества оказывает выбор коэффициента размытости ядра c_s. Коэффициент размытости ядра c_s – некоторое постоянное

число, от величины которого зависит степень «размытости» дельта – функции в окрестностях точки и соответственно степень гладкости полученной оценки.

Параметр размытости c_s здесь может быть определен путем решения задачи минимизации квадратичного показателя соответствия выхода объекта и выхода модели, основанного на «методе скользящего экзамена», когда в модели (1.4.1) исключается i-я переменная, предъявляемая для экзамена

$$R(c_s) = \sum_{k=1}^{s}\left(x_k - x_s(u_k, c_s)\right)^2 = \min_{c_s}, k \neq i. \qquad (1.4.6)$$

В случае если каждой компоненте вектора соответствует компонента вектораc_s, то во многих практических задачах c_s можно принять скалярной величиной, если предварительно привести компоненты вектора по выборке наблюдений, к одному и тому же интервалу, например, использовать операции центрирования и нормирования [30].

Эвристическая идея, лежащая в основе оценки (1.4.1), состоит в придании относительно большего веса наблюдениям, ближайшим к оцениваемой точке, в смысле расстояния, определяемого ядром [6].

Непараметрическую оценку функции регрессии (1.4.1) можно привести к следующему виду

$$x_s(u) = \sum_{i=1}^{s} x_i \varphi(u, u_i), \qquad (1.4.7)$$

где

$$\varphi(u, u_i) = \frac{\prod_{j=1}^{m} \Phi\left(\frac{u^j - u_i^j}{c_s}\right)}{\sum_{i=1}^{s}\prod_{j=1}^{m} \Phi\left(\frac{u^j - u_i^j}{c_s}\right)} \qquad (1.4.8)$$

Таким образом, непараметрическую оценку функции регрессии (1.4.1) можно привести к виду (1.4.7), аналогичному описанию нейросети.

1.5 Задача анализа данных

Анализ данных представляет собой направление, посвященное извлечению знаний из некоторого набора предоставленных данных [3, 10]. Данные могут быть представлены в виде матрицы наблюдений «входных – выходных» переменных процесса (таблица 1.5.1).

Таблица 1.5.1 – Матрица наблюдений «входных – выходных переменных процесса»

i	u				x
1	u_{11}	u_{21}	…	u_{m1}	x_1
2	u_{12}	u_{22}	…	u_{m2}	x_2
3	u_{13}	u_{23}	…	u_{m3}	x_3
4	u_{14}	u_{24}	…	u_{m4}	x_4
5	u_{15}	u_{25}	…	u_{m5}	x_5
6	u_{16}	u_{26}	…	u_{m6}	x_6
7	u_{17}	u_{27}	…	u_{m7}	x_7
8	u_{18}	u_{28}	…	u_{m8}	x_8
9	u_{19}	u_{29}	…	u_{m9}	x_9
…	…	…	…	…	…
s	u_{1s}	u_{2s}	…	u_{ms}	x_s

Здесь строки представляют собой наблюдения, а столбцы – переменные процесса. В данном случае мы имеем объект, на вход которого поступает переменная $u = (u_1, u_2, \dots, u_m)$, а на выходе скалярная переменная. В таблице отображаются результаты измерений «входных – выходных» переменных исследуемого объекта (i – номер наблюдения).

Анализ данных, представленных в матрицах наблюдений, включает в себя решение двух задач [10]:

- обнаружение закономерных связей между элементами таблицы;

- использование обнаруженных закономерностей для прогнозирования значений одних элементов таблицы по известным значениям других ее элементов.

Эти задачи могут решаться как по раздельности, так и совместно. Задачи анализа данных классифицируются в зависимости от расположения прогнозируемых элементов и от их количества. Кроме того, выделяют типы задач в соответствии со шкалами, в которых измеряются значения предсказываемых элементов (абсолютная шкала, шкала отношений, интервалов, порядка, наименований) [12]. Н.Г.Загоруйко в своей книге [10] дает следующую классификацию задачам анализ данных (рисунок 1.5.1):

Рисунок 1.5.1 – Классификация задач анализа данных

К задачам анализа данных относится и задача первичной обработки данных. При решении задачи идентификации мы оперируем с наблюдениями «входных – выходных» переменных процесса вида $\{u_i, x_i, i = \overline{1,s}\}$. От качества этих данных во многом зависит и качество решения поставленной задачи.

Поэтому важную роль приобретет этап первичной обработки данных, предшествующий процессу моделирования. Данный этап направлен на обработку некоторых особенностей, присутствующих в данных, которые негативно влияют на процесс идентификации. К таким особенностям можно отнести наличие в исходных выборках пропусков, наличие в измерениях выбросов (промахов) и др. Выбросом мы назовем аномальное значение измеряемого параметра, не являющееся особенностью исследуемого процесса, вызванное, к примеру, сбоями аппаратуры. Выбросы – это наблюдения, сильно отличающиеся от основной массы элементов выборки [3]. Также выброс может быть следствием неточных входных данных, погрешностей, вносимых на отдельных этапах измерений, погрешностей самих методов вычислений, сбоя оборудования, ошибки оператора и других причин. Наличие подобных особенностей значительно усложняет процесс моделирования. Большинство известных методов анализа данных не способны обработать подобную информацию.

Представим исходные данные об исследуемом процессе в виде матрицы наблюдений (таблица 1.5.2).

В таблице 1.5.2. приняты следующие обозначения: «–» – пропуск в матрице наблюдений, «*» – выброс. Данная таблица иллюстрирует общий случай, когда пропуски и выбросы располагаются в матрице наблюдений хаотично как по входным, так и по выходным переменным.

Для работы с данными, содержащими выбросы, применяют методы робастной статистики, которые позволяют сгладить влияние «промаха» на результаты моделирования. Также существуют приемы, позволяющие вовсе исключить выброс из исходной выборки.

Таблица 1.5.2 – Матрица наблюдений «входных – выходных» переменных исследуемого процесса в общем виде

i	***u***				***x***
1	–	u_{21}	…	u_{m1}	x_1
2	–	u_{22}	…	u_{m2}	x_2
3	u_{13}	u_{23}	…	u_{m3}	x_3
4	u_{14}	–	…	u_{m4}	*
5	u_{15}	u_{25}	…	u_{m5}	x_5
6	u_{16}	u_{26}	…	u_{m6}	x_6
7	u_{17}	*	…	u_{m7}	–
8	u_{18}	u_{28}	…	u_{m8}	x_8
9	u_{19}	–	…	u_{m9}	x_9
…	…	…	…	…	…
s	u_{1s}	u_{2s}	…	u_{ms}	x_s

Выводы по главе 1

В данной главе выполнена постановка задачи идентификации, приведено подробное определение таких понятий как априорная и текущая информация, рассмотрены существующие методы параметрической и непараметрической идентификации, а также поставлена задача анализа данных.

2 Задача заполнения пропусков в матрице наблюдений

2.1 Причины появления пропусков в данных и их классификация

Пропуски «входных – выходных» переменных исследуемого процесса могут быть вызваны причинами различного характера и располагаться в матрице наблюдений хаотично, как это и видно из таблицы 1.5.2. Они могут являться следствием причин как технического характера, например, неисправности измерительного прибора, так и быть вызваны ошибкой в работе оператора. Также, причиной может послужить различная дискретность контроля «входных – выходных» переменных процесса.

Рассмотрим объект, представленный на рисунке 2.1.1.

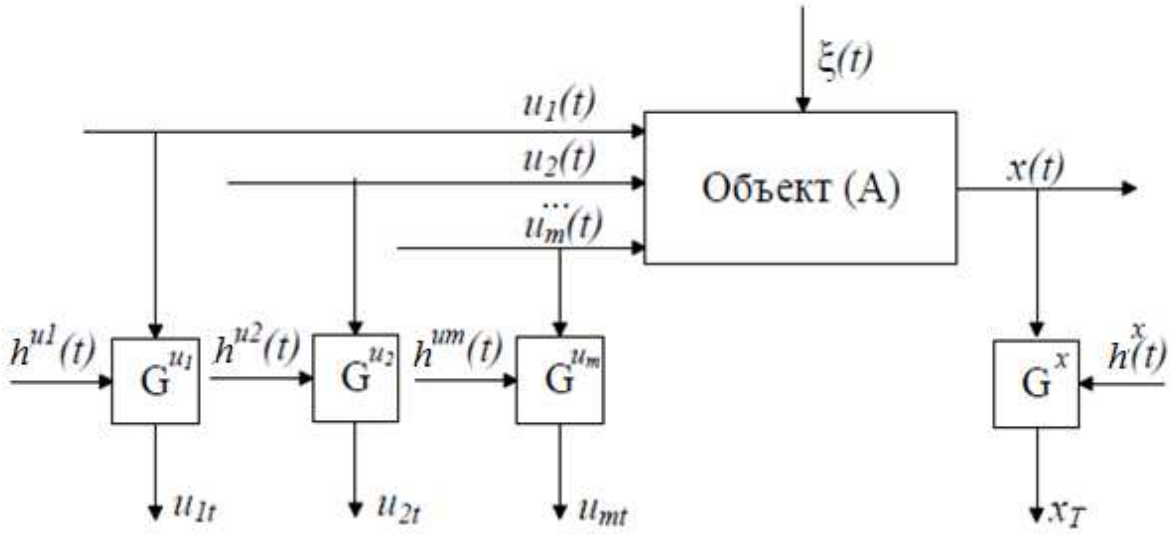

Рисунок 2.1.1 – Объект с различной дискретностью измерения переменных

Для рассматриваемого объекта векторная входная переменная измеряется с дискретностью $\Delta t'$, а выходная переменная $x(t) -$ с дискретностью $\Delta T'$, при этом $\Delta T = 3\Delta t$. Матрица наблюдений для данного объекта представлена в таблице 2.1.1.

Таблица 2.1.1 – Матрица наблюдений процесса с различной дискретностью контроля «входных – выходных» переменных

i	u				x
1	u_{11}	u_{21}	…	u_{m1}	x_1
2	u_{12}	u_{22}	…	u_{m2}	–
3	u_{13}	u_{23}	…	u_{m3}	–
4	u_{14}	u_{24}	…	u_{m4}	x_4
5	u_{15}	u_{25}	…	u_{m5}	–
6	u_{16}	u_{26}	…	u_{m6}	–
7	u_{17}	u_{27}	…	u_{m7}	x_7
8	u_{18}	u_{28}	…	u_{m8}	–
9	u_{19}	u_{29}	…	u_{m9}	–
…	…	…	…	…	…
s	u_{1s}	u_{2s}	…	u_{ms}	x_s

В данном случае пропуски располагаются по выходной переменной $x(t)$ и вызваны длительностью контроля переменной.

Наиболее простой и распространенный метод работы с данными, содержащими пропуски – исключение строки с пробелом целиком. Но тогда вместе с пропусками мы вычеркиваем из матрицы наблюдений и заполненные ячейки, а это не разумно с практической точки зрения. К тому же этим сокращается объём исходной выборки, а для решения задач идентификации предпочтительно иметь выборки большего объема.

Так как большинство существующих методов анализа данных не рассчитано на работу с матрицами наблюдений, содержащими пропуски, то предлагается заполнить недостающие значения переменных x в таблице 2.1.1 их оценками x_s [1].

Здесь $x_s -$ это оценки недостающих наблюдений. Различные методы обработки данных с пропусками, методы заполнения пропусков их оценочными значениями описываются Р.Дж. Литтлом и Д.Б.Рубином в [6].

Таблица 2.1.2 – Заполненная матрица наблюдений процесса с различной дискретностью контроля «входных – выходных» переменных

$\boldsymbol{i}$	$\boldsymbol{u}$				$\boldsymbol{x}$
1	u_{11}	u_{21}	…	u_{m1}	x_1
2	u_{12}	u_{22}	…	u_{m2}	$\boldsymbol{x_{s2}}$
3	u_{13}	u_{23}	…	u_{m3}	$\boldsymbol{x_{s3}}$
4	u_{14}	u_{24}	…	u_{m4}	x_4
5	u_{15}	u_{25}	…	u_{m5}	$\boldsymbol{x_{s5}}$
6	u_{16}	u_{26}	…	u_{m6}	$\boldsymbol{x_{s6}}$
7	u_{17}	u_{27}	…	u_{m7}	x_7
8	u_{18}	u_{28}	…	u_{m8}	$\boldsymbol{x_{s8}}$
9	u_{19}	u_{29}	…	u_{m9}	$\boldsymbol{x_{s9}}$
…	…	…	…	…	…
s	u_{1s}	u_{2s}	…	u_{ms}	x_s

Здесь x_s – это оценки недостающих наблюдений. Различные методы обработки данных с пропусками, методы заполнения пропусков их оценочными значениями описываются Р.Дж. Литтлом и Д.Б.Рубином в [6].

Приведем в качестве примера процесса с различной дискретностью контроля переменных процесс сжигания угля в котлоагрегате энергоблока.

Входными переменными технологического процесса, протекающего в котле, являются: температура аэросмеси, температура воздуха, температура питательной воды, расход топлива, расход питательной воды, химический состав питательной воды, влажность топлива, «тонина» помола, качество топлива.

Выходными параметрами являются: температура острого пара, давление острого пара, расход острого пара, содержание кислорода в уходящих газах, температура уходящих газов. Общая схема котлоагрегата энергоблока со всеми входными и выходными переменными представлена на рисунке 2.1.2.

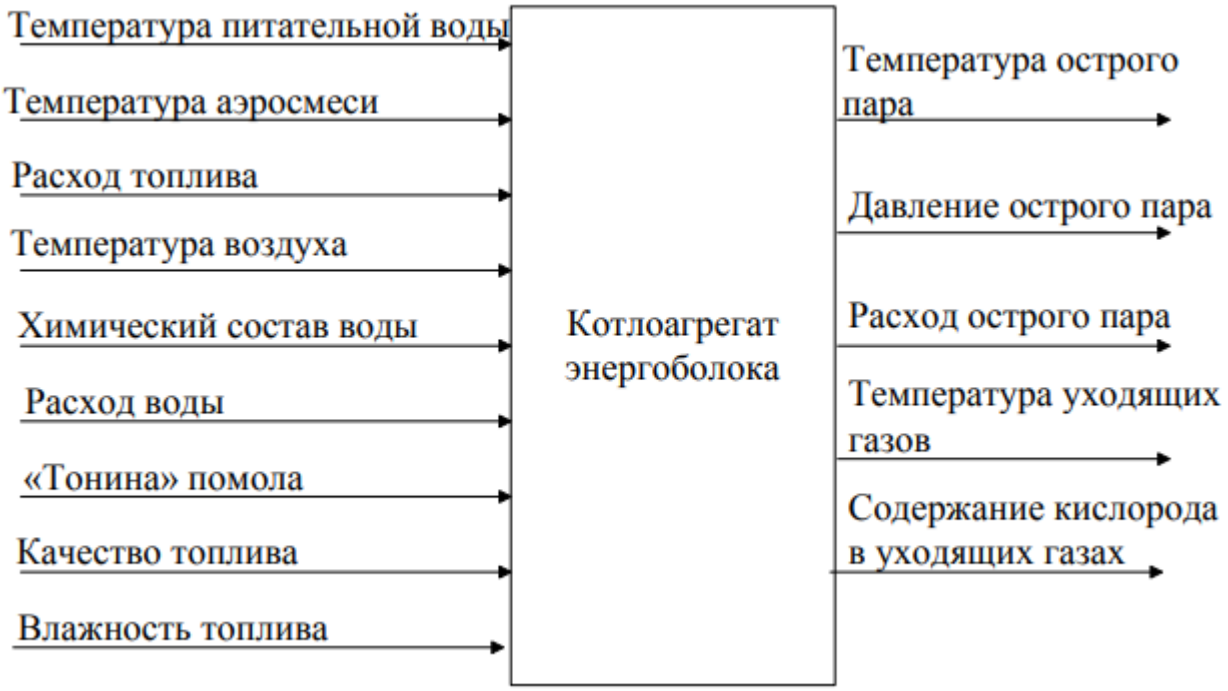

Рисунок 2.1.2 – Процесс сжигания угля в котлоагрегате энергоблока

Данные, снятые с объекта, хранятся в табличной форме и имеют следующий вид:

				входные переменные					выходные переменные				
Блок	Котел	Дата	Час	T питатель-ной воды	Расход воды	T воздуха (слева, справа)		P воздуха	G o.n.	P o.n.	T o.n.	T ух.газ.	O2 ух.газ.
9	A	01.12.2003	1	227	200				242	126	555	146	6,25
9	A	01.12.2003	2	227	210				231	124	549	146	6,25
9	A	01.12.2003	3	227	225				229	122	551	146	6,25
9	A	01.12.2003	4	227	225	270	275	130	237	123	555	146	6,25
9	A	01.12.2003	5	228	235				252	125	555	147	6,25
9	A	01.12.2003	6	230	250				270	129	555	148	6,25
9	A	01.12.2003	7	230	250				265	130	553	148	6,25
9	A	01.12.2003	8	229	250	270	275	130	264	127	554	148	6,25
9	A	01.12.2003	9	228	250				255	127	551	148	6,25
9	A	01.12.2003	10	227	250				242	125	549	146	6,25
9	A	01.12.2003	11	226	250				230	120	559	146	6,25
9	A	01.12.2003	12	227	250	270	275	130	265	125	555	146	6,25
9	A	01.12.2003	13	227	250				266	122	560	146	6,25
9	A	01.12.2003	14	229	230				286	125	552	146	6,25
9	A	01.12.2003	15	228	230				256	119	545	147	6,25
9	A	01.12.2003	16	228	230	280	280	130	283	119	550	146	6,25
9	A	01.12.2003	17	230	280				284	125	553	146	6,25
9	A	01.12.2003	18	230	280				297	127	558	147	6,25
9	A	01.12.2003	19	230	280				317	130	552	149	6,25
9	A	01.12.2003	20	230	245	280	280	130	279	127	555	150	6,25
9	A	01.12.2003	21	230	245				293	127	555	146	6,25
9	A	01.12.2003	22	230	245				276	127	552	146	6,25
9	A	01.12.2003	23	230	240				283	127	552	146	6,25
9	A	01.12.2003	24	230	270	280	280	130	290	128	558	147	6,25
9	A	02.12.2003	1	230	250				242	125	548	148	0

Рисунок 2.1.3 – Результаты измерений «входных – выходных» переменных процесса сжигания угля в котлоагрегате энергоблока

Из таблицы, представленной на рисунке 2.1.3, мы видим, что некоторые «входные – выходные» переменные процесса измеряются раз в час (температура питательной воды, расход воды, температура уходящих газов и др.). Другие переменные процесса (температура и расход воздуха) измеряются раз в четыре часа. В данном случае пропуски в матрице наблюдений процесса вызваны именно различной дискретностью контроля его переменных.

2.2 Алгоритмы заполнения пропусков в данных

На данный момент существует множество методик позволяющих эффективно восстанавливать данные с пропусками, но у каждой из них есть свои плюсы и минусы.

Данные методики в большинстве своем используются при незначительном количестве пропусков. Перечислим наиболее распространенные методы с указанием их основных особенностей:

Исключение строк с наличием пропусков. Данный метод легко реализуем, но необходимым условием его применения является следование данных требованию MCAR (missing completely at random), т.е. пропуски в данных по переменным должны быть полностью случайными. Кроме того, он обычно применяется лишь при незначительном количестве пропусков в таблице, иначе полученная на выходе таблица данных становится непредставительной. Главный недостаток такого подхода обусловлен потерей информации при исключении неполных данных.[5]

Заполнение пропусков средними по столбцу значениями. Данный метод также легко реализуем, но его применение имеет смысл только в случае удовлетворения данных условию MAR (missing at random), т.е. когда пропуски в данных по переменным являются случайными и сам механизм пропусков несущественен. К недостаткам метода относят вносимые искажения в распределения данных, уменьшение дисперсии.

Метод ближайших соседей. В основе метода лежит механизм поиска строк таблицы, которые по определенному критерию являются ближайшими к строке с пропусками. Для заполнения пропуска значения данной переменной (в фиксированном столбце) у соседних строк усредняются с определенными весовыми коэффициентами, обратно пропорциональными расстоянию к строке с пропуском. При большом количестве пропусков данный метод также практически неприменим, поскольку базируется на существовании связей между строками в таблице.

Метод сплайн-интерполяции. Для успешного применения необходимо, чтобы данные следовали условию MAR. Недостатки метода следуют из самой его идеи. Например, в случае восстановления группы пропусков, следующих подряд друг за другом, результат аппроксимации сплайном данной группы не всегда может дать оценки, приближающиеся с достаточной точностью к значениям, которые могли бы быть на месте пропусков.

Метод максимальной правдоподобности и EM-алгоритм. Метод требует проверки гипотез о распределении значений переменных. Применение осложняется при большом количестве пропущенных значений переменной. Особенность данного метода состоит в построении модели порождения пропусков с последующим получением выводов на основании функции правдоподобия, построенной при условии справедливости данной модели, с оцениванием параметров методами типа максимального правдоподобия. Отметим, что для данных методов возможно построение моделей, учитывающих конкретную специфику области, и, как следствие, возможна постановка более слабых условий к данным (слабее MAR).[34]

Алгоритмы ZET и ZetBraid. По сути, алгоритм ZET является детально проработанной и апробированной технологией верификации экспериментальных данных, основанной на гипотезе их избыточности. Главная идея алгоритма ZET заключается в подборе «компетентной матрицы», используя данные, из нее находят параметры зависимости, которая применяется для прогнозирования пропущенного значения. Субъективизм определения размерности «компетентной матрицы» приводит к учету неинформативных и шумовых факторов и смещению оценки неизвестного

значения. Основное отличие алгоритма ZetBraid состоит в определении оптимального размера «компетентной матрицы». Данные алгоритмы хорошо показали себя, но статистическая оценка неизвестного значения исключительно на основе корреляционно регрессионного анализа и необходимость задания ряда важных параметров приводит к необходимости убедиться в правдоподобности восстановленных значений.[35]
Рассмотрим подробнее методику алгоритма ZET.

2.3 Алгоритм ZET

В основе алгоритма ZET лежат три предположения. Первое (гипотеза избыточности) состоит в том, что реальные таблицы имеют избыточность, проявляющуюся в наличии похожих между собой объектов (строк) и зависящих друг от друга свойств (столбцов). Если же избыточность отсутствует (как, например, в таблице случайных чисел), то предпочесть один прогноз другому не возможно. Второе предположение (гипотеза локальной компактности) состоит в утверждении, что для предсказания пропущенного элемента b_{ij} нужно использовать не всю таблицу, а лишь ее «компетентную» часть, состоящую из элементов строк, похожих на строку i, и элементов столбцов, похожих на столбец j. Остальные строки и столбцы для данного элемента неинформативны. Их использование лишь разрушало бы локальную компактность подмножества компетентных элементов и ухудшало точность предсказания. Третье предположение (гипотеза линейных зависимостей) заключается в том, что из всех возможных видов зависимостей между столбцами (строками) в алгоритме ZET используются только линейные зависимости. Если зависимости носят более сложный характер, то для их надежного обнаружения требуется такой большой объем данных, который в реальных задачах встречается нечасто.

Для различных прикладных задач были сделаны многочисленные модификации базового алгоритма ZET, отличающиеся своим назначением и наборами разных режимов работы.

В основе алгоритма ZET лежат три предположения. Первое (гипотеза избыточности) состоит в том, что реальные таблицы имеют избыточность, проявляющуюся в наличии похожих между собой объектов (строк) и зависящих друг от друга свойств (столбцов). Если же избыточность отсутствует (как, например, в таблице случайных чисел), то предпочесть один прогноз другому не возможно.

Второе предположение (гипотеза локальной компактности) состоит в утверждении, что для предсказания пропущенного элемента b_{ij} нужно использовать не всю таблицу, а лишь ее «компетентную» часть, состоящую из элементов строк, похожих на строку i, и элементов столбцов, похожих на столбец j. Остальные строки и столбцы для данного элемента неинформативны. Их использование лишь разрушало бы локальную компактность подмножества компетентных элементов и ухудшало точность предсказания.

Третье предположение (гипотеза линейных зависимостей) заключается в том, что из всех возможных видов зависимостей между столбцами (строками) в алгоритме ZET используются только линейные зависимости. Если зависимости носят более сложный характер, то для их надежного обнаружения требуется такой большой объем данных, который в реальных задачах встречается нечасто. В работе алгоритма ZET можно выделить три этапа.

Основные этапы алгоритма ZET для обработки таблицы A с l пропусками:

1 Предварительная обработка начальных данных.

2 Прогнозирование пропуска - выполняется l раз:

2.1 Формирование компетентной матрицы.

2.2 Подбор параметров модели прогнозирования.

2.3 Прогнозирование пропуска.

Рассмотрим подробнее каждый этап.

1 Вначале столбцы матрицы A нормируются по дисперсиям для приведения различных свойств объектов к единой шкале

$$a_{ij} = \frac{a_{ij} - \overline{a_j}}{G_j} \quad (2.1.1)$$

2 Следующие этапы выполняют l раз. Пусть координаты текущего элемента с пропуском x,y.

2.1 Формирование компетентной матрицы

2.1.1 Задать размеры компетентной матрицы $s_{ij},\ i = \overline{1,p}, j = \overline{1,q},$ $2 < p < m,\ 2 < q < n.$

2.1.2 Выбрать (p-1) компетентных строк для строки с пропуском.

Компетентность L строки i по отношению к строке с пропуском y определяется по формуле

$$L_{iy} = \frac{t_{iy}}{r_{iy}} \quad (2.1.2)$$

где t_{iy} - комплектность, то есть число значений известных для обеих строк i и y;

r_{iy} - декартово расстояние между строками (элементы с пропусками не учитываются).

Компетентная строка не должна содержать пропуска на x-й позиции.

2.1.3 Выбрать (q-1) компетентных столбцов для столбца с пропуском.

Компетентность L столбца i по отношению к столбцу с пропуском x определяется по формуле

$$L_{iy} = |k_{ik}| * t_{ik} \quad (2.1.3)$$

где t_{ik} - комплектность столбцов i и x;

$k_{ix} -$ коэффициент корреляции между столбцами i и x.

При расчете k_{ix} используются только те значения столбцов, которые принадлежат к компетентным строкам. Компетентный столбец не должен содержать пропуск на y-й позиции.

2.2 Подбор параметров моделей прогнозирования a_r (по строкам) и a_c (по столбцам) – коэффициенты регулирующие влияние компетентности на результат предсказания.

2.2.1 Задаем пределы изменения коэффициентов a_r и a_c и шаг их изменения.

2.2.2 Находим оптимальные коэффициенты a_r и a_c для прогноза пропуска по строкам и по столбцам по следующему алгоритму (одинаков для строк и столбцов). Подавая значения коэффициента a ($a = a_r$ для строк, $a = a_c$ для столбцов) в указанных пределах и с указанным шагом минимизируем функцию

$$\sum_i |a_{ik} - b_{ik}| \to min \ , i \neq @ \qquad (2.1.4)$$

где a_{ik} - реальное значение элемента i строки (столбца) k с пропуском;

b_{ik} - прогноз этого элемента с помощью компетентных строк (столбцов);

b_{ik} рассчитываются по формуле

$$b_{ik} = \frac{\sum_{j=1}^{c-1} bl_{jk} * L_{ij}^a}{\sum_{j=1}^{c-1} L_{ij}^a} \qquad (2.1.5)$$

где c=p для строк и c=q для столбцов;

bl_{jk} - прогноз для известных значений строки (столбца) с пропуском k с помощью i-й строки (столбца), рассчитывается с помощью линейной регрессии вида y=ax+b по МНК.

2.3 Прогнозирование пропуска

2.3.1 Прогнозирование пропуска по столбцам выполняется по формуле

$$b_x = \frac{\sum_{i=1}^{q-1} bl_{ik} * L_{ik}^a}{\sum_{i=1}^{q-1} L_{ik}^a}. \qquad (2.1.6)$$

2.3.2 Прогнозирование пропуска по столбцам выполняется по формуле

$$b_y = \frac{\sum_{i=1}^{p-1} bl_{iy} * L_{iy}^a}{\sum_{i=1}^{p-1} L_{iy}^a}. \quad (2.1.7)$$

2.3.3 Общий прогноз получается усреднением прогнозов по строкам и столбцам

$$b_{yx} = \frac{b_y + b_x}{2}. \quad (2.1.8)$$

Программы заполнения пробелов могут работать в одном из следующих режимов:

1 Заполнение всех пробелов в таблице по указанному алгоритму.

2 Заполнение только тех пробелов, ожидаемая ошибка для которых не превышает заданной величины. Для определения ожидаемой ошибки предсказания вычисляется дисперсия значений подсказок bl_{ij}, получаемых от всех q столбцов и p строк компетентной подматрицы.

3 Заполнение пробелов только на базе информации, имеющейся в исходной таблице.

4 Заполнение каждого следующего пробела с использованием исходной информации и прогнозных значений ранее заполненных пробелов.

2.4 Модификации алгоритма ZET

Для различных прикладных задач были сделаны многочисленные модификации описанного выше базового алгоритма ZET, различающиеся своим назначением и наборами разных режимов работы. Программы заполнения пробелов могут работать в одном из следующих режимов:

- обнаружение грубых ошибок в указанной клетке или во всей таблице;
- заполнение всех пробелов;

- заполнение только тех пробелов, ожидаемая ошибка для которых не превышает заданной величины;

- заполнение пробелов только на базе информации, имеющейся в исходной таблице;

- заполнение каждого следующего пробела с использованием исходной информации и прогнозных значений ранее заполненных пробелов.

Для каждого из этих вариантов имеется несколько режимов выдачи промежуточных и окончательных результатов.

Ниже рассмотрим модификации алгоритмов ZET-R и ZET-D.

Алгоритм ZET-R используется для обнаружения грубых ошибок в исходной таблице данных (так называемый режим редактирования таблиц). Для этого программа по очереди предсказывает все элементы таблицы и сравнивает результаты предсказания с фактически имеющимися данными. Если предсказанное значение совпадает с исходным или мало отличается от него, то это означает, что элемент хорошо согласуется с закономерностями данной части таблицы данных. Если же обнаруживается большое расхождение, то выдается сигнал о необходимости проверки данного элемента. Если он отражает уникальный факт, выпадающий из общей закономерности, то его истинность нужно подтвердить. Если же он отражает ошибку, то ее нужно устранить. Таким путем удается обнаруживать грубые ошибки или умышленные искажения отдельных элементов таблицы данных.

Алгоритм ZET-D работает с таблицами типа «время – свойство». На рисунке 2.4.1, а более подробно представлена подробная таблица. T строк в этой таблице отражают значение n свойств $x_1, x_2, \dots, x_j, \dots, x_n$ некоторого объекта или процесса в последовательные моменты времени $t_1, t_2, \dots, t_\tau, \dots, t_T$. Таблицу можно переформировать, объединив в одну строку k – соседних по времени строк. Первая строка такой новой таблицы отражает данные из k – первых строк исходной таблицы $(t_1, t_2, \dots, t_{k-1}, t_k)$:

$b_{11}, b_{12}, \ldots, b_{1j}, \ldots, b_{1n},$

$b_{21}, b_{22}, \ldots, b_{2j}, \ldots, b_{2n},$

$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$

$b_{k1}, b_{k2}, \ldots, b_{kj}, \ldots, b_{kn}.$

Во вторую строку новой таблицы поместим k строк, начинающихся с момента времени t_2, в третью — с момента времени t_3 и т. д. до строки, начинающейся с момента t_{T-k+2}. В результате получим таблицу, состоящую из kn столбцов и T-k+2 строк (рисунок 2.4.1, б). Если строка t_T соответствовала, например, свойствам объекта в n-й год, то каждая строка новой таблицы будет соответствовать периоду в k лет.

	x_1	x_2	x_j	x_n
t_1	b_{11}	b_{12}	b_{1j}	b_{1n}
t_2	b_{21}	b_{22}	b_{2j}	b_{2n}
...	.....			
t_τ	$b_{\tau 1}$	$b_{\tau 2}$	$b_{\tau j}$	$b_{\tau n}$
...	.....			
t_T	b_{T1}	b_{T2}	b_{Tj}	b_{Tn}

а

	1	2	$k-1$	k
1	t_1	t_2	t_{k-1}	t_k
2	t_2	t_3	t_k	t_{k+1}
...	.....			
τ	t_τ	$t_{\tau+1}$	$t_{\tau+k-2}$	$t_{\tau+k-1}$
...	.....			
$T-k+2$	t_{T-k+2}	t_{T-k+1}	t_T	t_{T+1}

б

Рисунок 2.4.1 – Преобразование таблицы методом «змейки»

Все элементы новой таблицы известны, кроме последнего сегмента t_{T+1}, в котором должны быть отражены свойства изучаемого объекта или процесса в момент времени T+1, следующий за последним моментом из отраженных объектов в протоколе наблюдения.

Если каждую пустую j-ю клеточку последнего сегмента заполнить алгоритмом ZET, то получим прогноз свойств x_j в момент времени t=T+1. Описанный способ формирования длинных строк из сдвигаемых коротких и последующего прогнозирования элементов короткой строки в одной из недавних работ был назван методом «змейки».

В [19] описано несколько вариантов этого алгоритма для исходных таблиц разного характера. Есть вариант (алгоритм ZETMC), ориентированный на таблицы с фиксированным порядком следования свойств x_j. Примером

такой таблицы может служить сводка ежемесячных показателей деятельности предприятия за T лет. Здесь роль свойств играют показатели в j-е месяцы, а -я строка — это данные за τ-й год. Прогнозирование делается не для всех месяцев года сразу, а последовательно для каждого следующего месяца. Начало годового цикла — вещь условная, цикл можно начинать с любого месяца. Пусть таблица содержит данные за период с 1970 по 1995 годы. Возьмем первый столбец (данные за январи) и поставим его за последним столбцом (за декабрями). Если его сдвинуть на одну строку вверх, то в первой строке окажутся данные за год, начинающийся в феврале 1970-го и заканчивающийся в январе 1971-го года. В последней строке будет цикл, который начинается в феврале 1995-го и заканчивается январем 1996-го года. Данные за январь 1996-го года нам не известны, и эту пустую клеточку таблицы, мы заполняем с помощью алгоритма ZET.

Затем, мы можем перенести с первой на последнюю позицию, столбец с данными за феврали. Годовые циклы будут начинаться с марта текущего года, и заканчиваться в феврале следующего года. Заполнив новую пустую клеточку, мы предскажем отсутствующее значение февраля 1996-го года. Эту процедуру поочередного переноса первых столбцов на последнее место и прогнозирования очередного неизвестного значения можно продолжать сколь угодно долго.

Однако ясно, что с удалением прогнозируемого момента времени от момента последнего наблюдения точность прогноза будет падать, причем скорость нарастания ошибок зависит от характера наблюдаемого процесса и заранее предсказана быть не может. Для каждой конкретной таблицы рекомендуется метод ретроспективного анализа: на прошлом материале делаются прогнозы известных данных, и фиксируется зависимость ошибок прогноза от длительности периодов упреждения. В результате можно предположительно говорить об ожидаемой ошибке прогноза при заданном периоде упреждения или о максимальном периоде упреждения при заданной допустимой величине ошибки прогноза.

Выводы по главе 2

В данной главе рассмотрен вопрос причины появления пропусков в массивах данных, а также основные существующие алгоритмы по их заполнению. Далее более детально описан алгоритм ZET и его наиболее популярные модификации ZET-R и ZET-D.

3 Вычислительные эксперименты

Исходные данные

Исходные данные представляют собой начальную таблицу, состоящую из 300 строк и 6 переменных, заполненную случайными числами от 1 до 50. Ошибка представляет собой сумму модулей расстояний между исходным значение оценки и полученным в результате восстановления алгоритмом ZET, деленную на сумму всех оценок.

Так же имеется таблица с хаотично разбросанными пропусками, как по входным, так и по выходным переменным, мы меняем только процент количества пропусков. Далее, заполняем пропущенные значения, созданные на предыдущем шаге.

3.1 Результаты исследования ZET – алгоритма

Средний результат среднеквадратичной погрешности при количестве тестов 40, диапазоне значений A_r и A_c от -20 до 20 и шагом 0.5.

Таблица 3.1.1 – Средний результат погрешности с ошибкой 5%

Количество переменных	Количество строк	Размер компетентной матрицы	Процент пропусков	Средний результат среднеквадратичной погрешности
6	300	4x4	5	0.065498
6	300	4x4	10	0.044475
6	300	4x4	15	0.035540
6	300	3x3	5	0.076104
6	300	5x5	5	0.053457
10	300	5x5	5	0.043978
15	300	5x5	5	0.081456
15	500	5x5	5	0.066915
15	800	5x5	5	0.040219

Таблица 3.1.2 – Средний результат погрешности с ошибкой 10%

Количество переменных	Количество строк	Размер компетентной матрицы	Процент пропусков	Средний результат среднеквадратичной погрешности
6	300	4x4	5	0.136129
6	300	4x4	10	0.091568
6	300	4x4	15	0.085540
6	300	3x3	5	0.147610
6	300	5x5	5	0.092130
10	300	5x5	5	0.092205
15	300	5x5	5	0.171916
15	500	5x5	5	0.126915
15	800	5x5	5	0.095219

Таблица 3.1.3 – Средний результат погрешности с ошибкой 15%

Количество переменных	Количество строк	Размер компетентной матрицы	Процент пропусков	Средний результат среднеквадратичной погрешности
6	300	4x4	5	0.136129
6	300	4x4	10	0.091568
6	300	4x4	15	0.085540
6	300	3x3	5	0.147610
6	300	5x5	5	0.092130
10	300	5x5	5	0.092205
15	300	5x5	5	0.171916
15	500	5x5	5	0.126915
15	800	5x5	5	0.040819

Для наглядности ниже приведены графики сравнения с рассчитанными и истинными значениями.

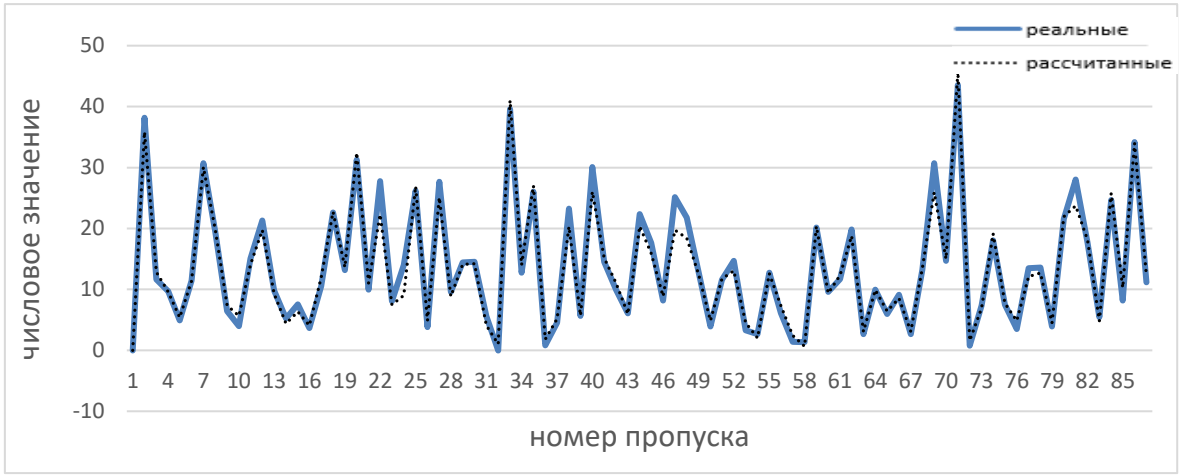

Рисунок 3.1.1 – Сравнение 5% пропусков (5% ошибка)

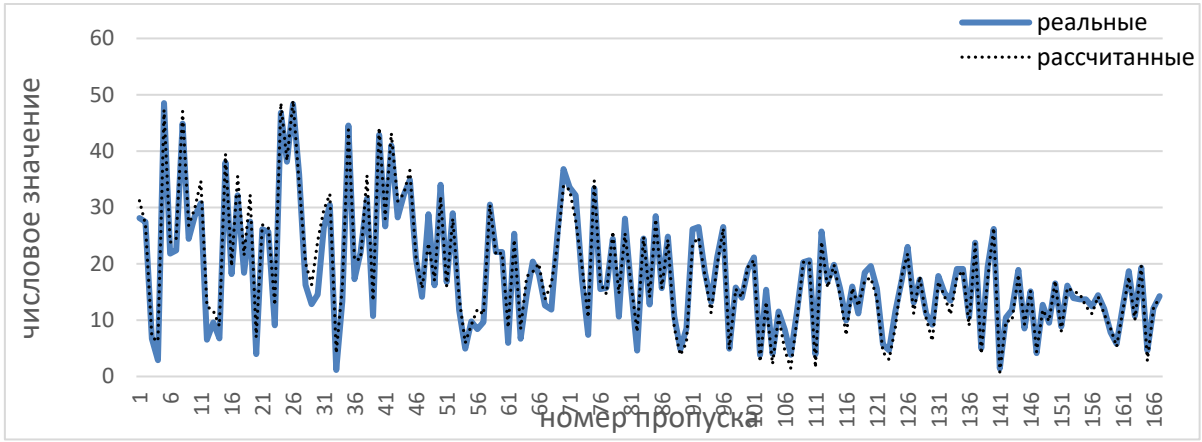

Рисунок 3.1.2 – Сравнение 10% пропусков (5% ошибка)

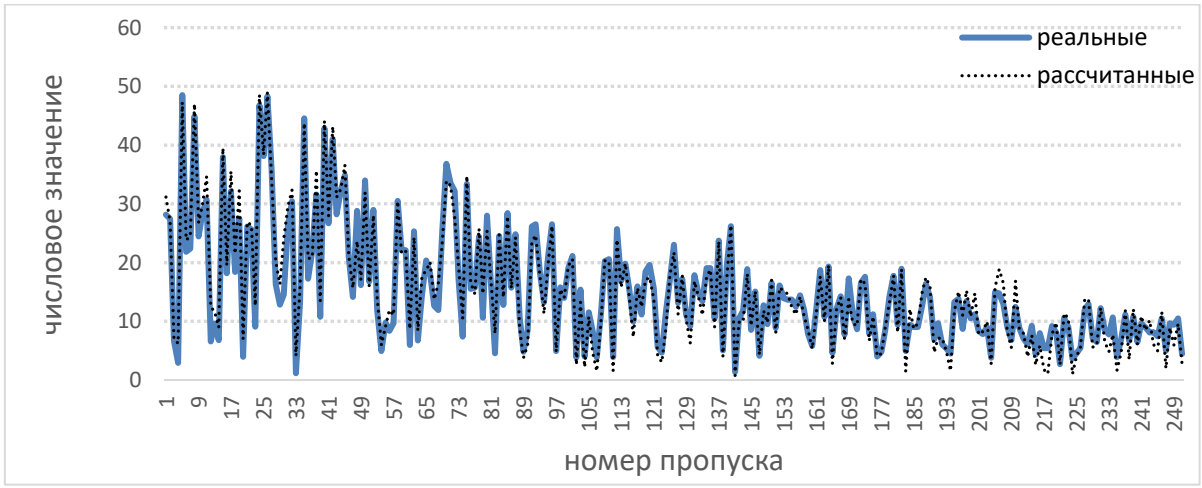

Рисунок 3.1.3 – 15% пропусков (5% ошибка)

Рисунок 3.1.4 – 5% пропусков (10% ошибка)

Рисунок 3.1.5 – 10% пропусков (10% ошибка)

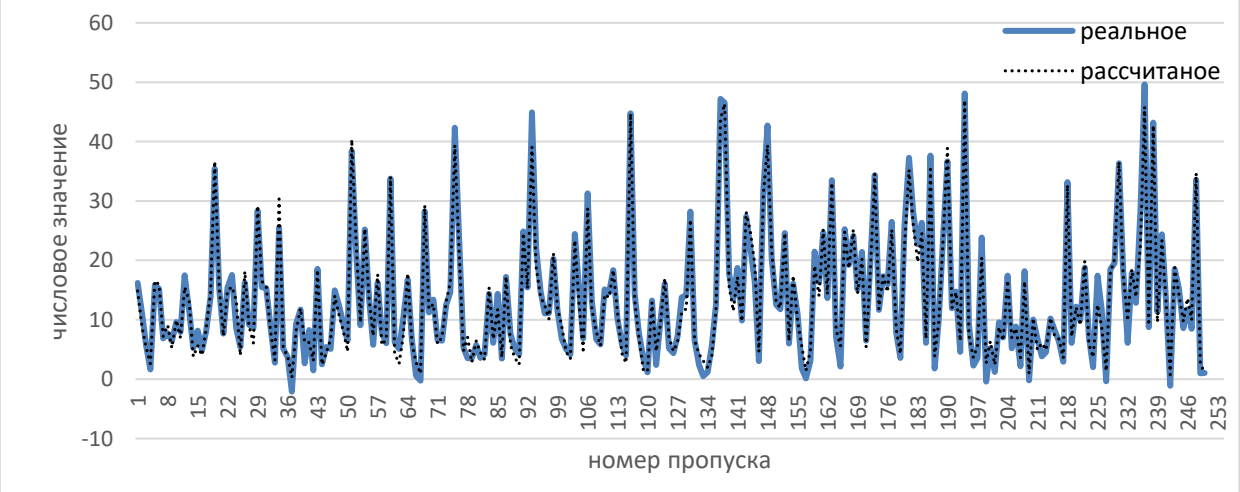

Рисунок 3.1.6 – 15% пропусков (10% ошибка)

Рисунок 3.1.7 – 5% пропусков (15% ошибка)

Рисунок 3.1.8 – 10% пропусков (15% ошибка)

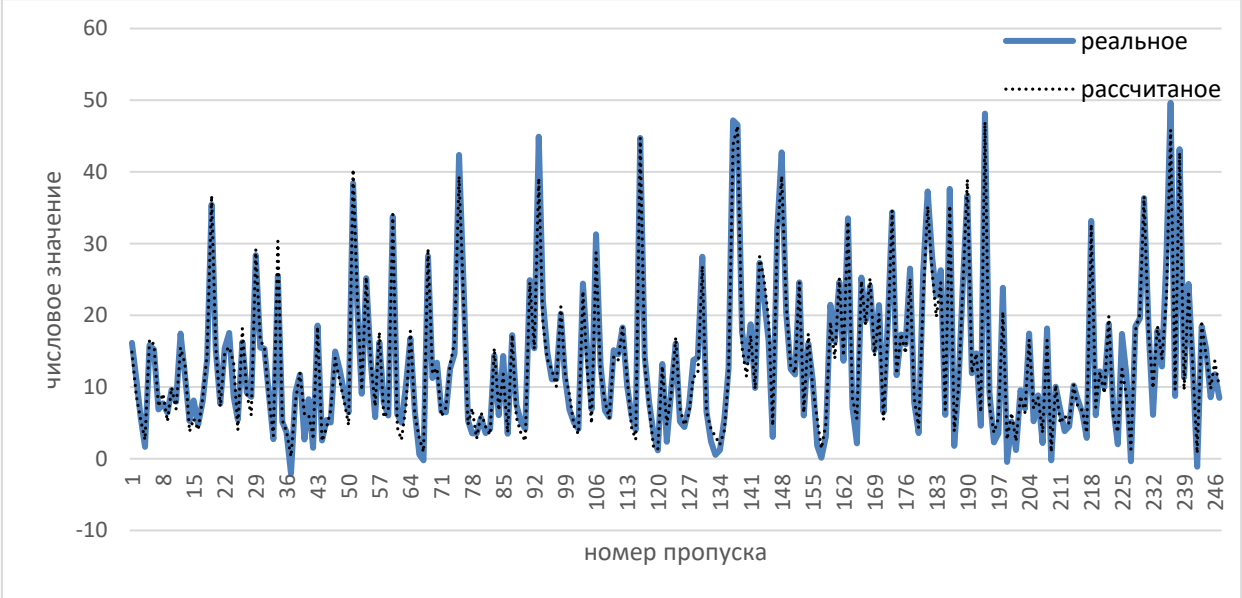

Рисунок 3.1.9 – 15% пропусков (15% ошибка)

Вывод: По рисункам видно, что реальные значения почти не отличаются от рассчитанных значений, но по пиковым значениям расчеты производились значительно хуже.

3.2 Результаты исследования непараметрической методики восстановления пропусков в данных

Средний результат среднеквадратичной погрешности при количестве тестов 40 и погрешности 5% приведен в таблице ниже.

Таблица 3.2.1 – Средний результат погрешности (5% ошибка)

Количество переменных	Количество строк	Коэффициент ядра	Процент пропусков	Средний результат среднеквадратичной погрешности
6	300	5	5	0.065186
6	300	5	10	0.050072
6	300	5	15	0.039435
6	300	2,27612306503 (оптимальное)	5	0.046671
6	300	2,501464837696 (оптимальное)	10	0.043547
6	300	5,430114764254 (оптимальное)	15	0.033517
10	300	3,324279803317 (оптимальное)	10	0.0350579
15	300	5,31347655644 (оптимальное)	10	0.029340
6	500	2,167820042651 (оптимальное)	10	0.032797
6	800	2,124723071139 (оптимальное)	10	0.026052

Лучший результат оказался при количестве переменных 10, количестве строк 800, проценте пропусков 15 и количестве тестов 40. Ниже приведен результат сочетания этих параметров.

Таблица 3.2.2 – Результат сочетания наилучших параметров

Количество переменных	Количество строк	Коэффициент ядра	Процент пропусков	Средний результат среднеквадратичной погрешности
10	800	3,1472396790 (оптимальное)	15	0.066204

При оптимальном значении ядра и выше процент не заполненных пропусков равен 0.

Таблица 3.2.3 – Средний результат погрешности (10% ошибка)

Количество переменных	Количество строк	Коэффициент ядра	Процент пропусков	Средний результат среднеквадратичной погрешности
6	300	5	5	0.119186
6	300	5	10	0.085072
6	300	5	15	0.073435
6	300	2,141336077591 (оптимальное)	5	0.117871
6	300	2,501464837696 (оптимальное)	10	0.082547
6	300	5,430114764254 (оптимальное)	15	0.065833
10	300	4,324279803317 (оптимальное)	10	0.058899
15	300	6,899965028045 (оптимальное)	10	0.051948
6	500	2,167820042651 (оптимальное)	10	0.059658
6	800	2,124723071139 (оптимальное)	10	0.049097

Таблица 3.2.4 – Средний результат погрешности (15% ошибка)

Количество переменных	Количество строк	Коэффициент ядра	Процент пропусков	Средний результат среднеквадратичной погрешности
6	300	5	5	0.167764
6	300	5	10	0.134072
6	300	5	15	0.104737
6	300	2,907714837836 (оптимальное)	5	0.154679
6	300	2,501464837696 (оптимальное)	10	0.111764
6	300	2,851524370629 (оптимальное)	15	0.104189
10	300	4,324279803317 (оптимальное)	10	0.078877
15	300	5,725000014808 (оптимальное)	10	0.071966
6	500	2,96097337533 (оптимальное)	10	0.092690
6	800	2,10000001802 (оптимальное)	10	0.064097

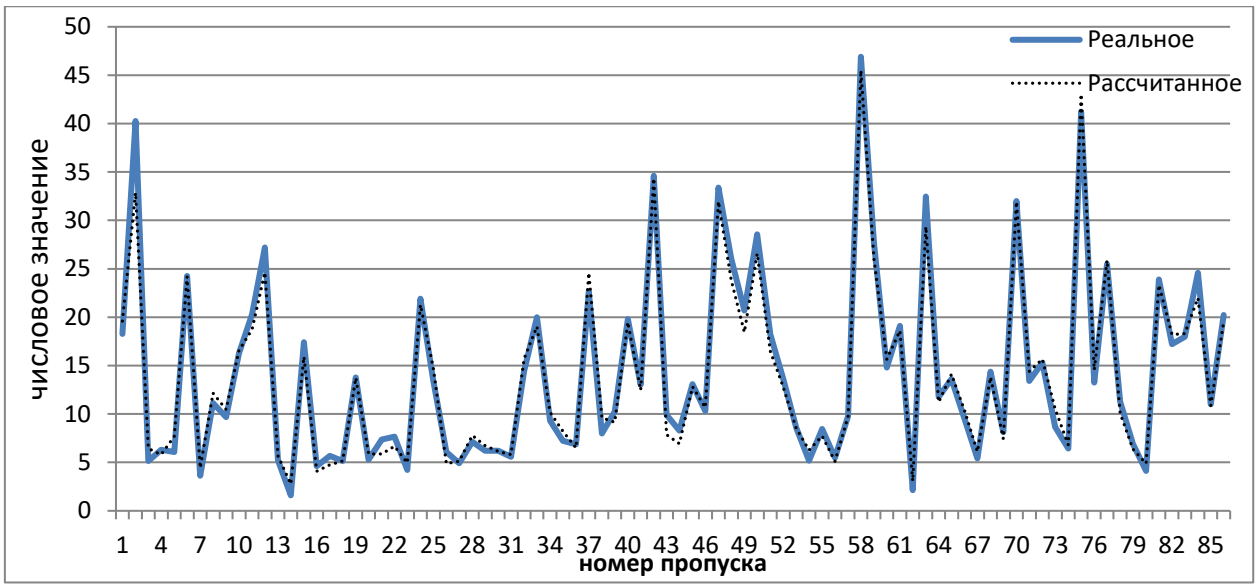

Рисунок 3.2.1 – 5% пропусков (5% ошибка)

График сравнения реальных значений и рассчитанных, при коэффициенте пропусков 10%, количестве переменных 6, количестве строк 300 и оптимальном ядре равном 2, 2348 изображен на рисунке 3.2.2.

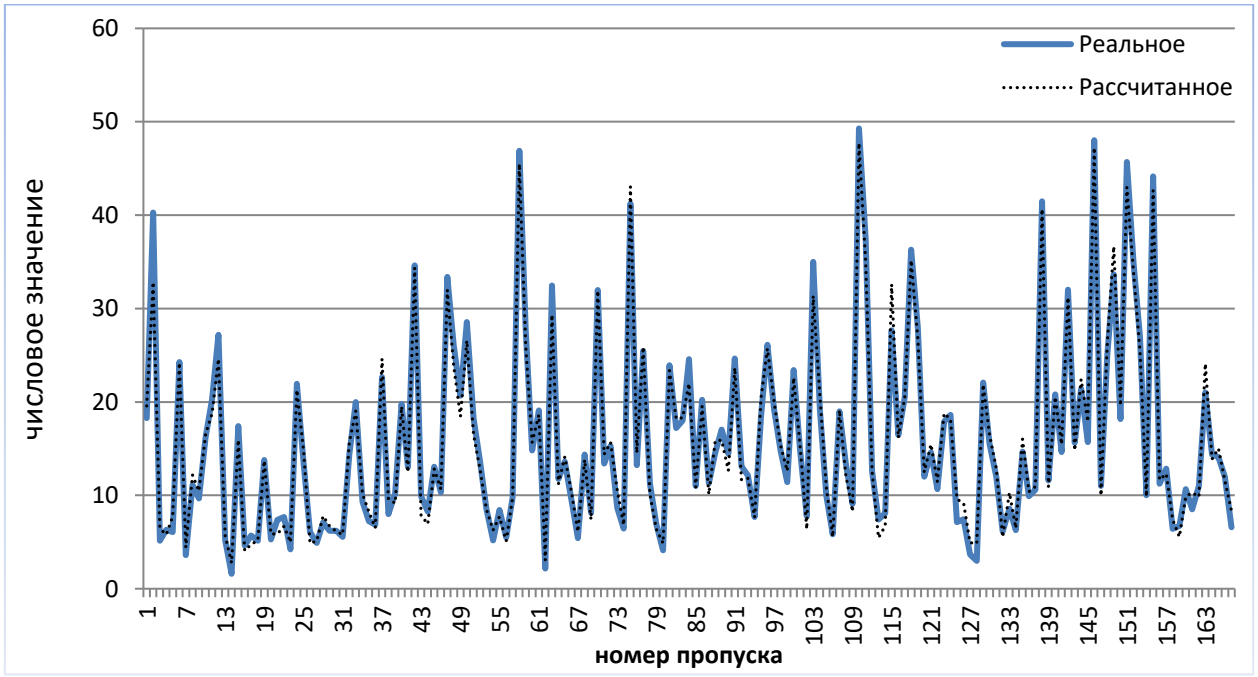

Рисунок 3.2.2 – 10% пропусков (5% ошибка)

График сравнения реальных значений и рассчитанных, при коэффициенте пропусков 10%, количестве переменных 6, количестве строк 300 и оптимальном ядре равном 3,1459 изображен на рисунке 3.2.3.

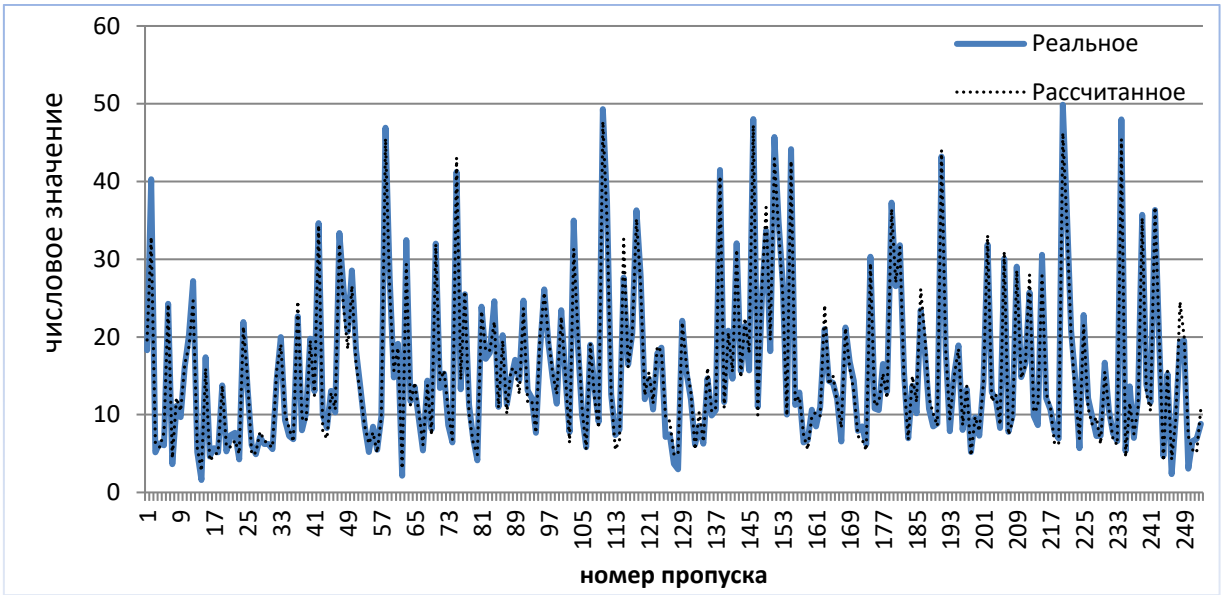

Рисунок 3.2.3 – 15% пропусков (5% ошибка)

Рисунок 3.2.4 – 5% пропусков (10% ошибка)

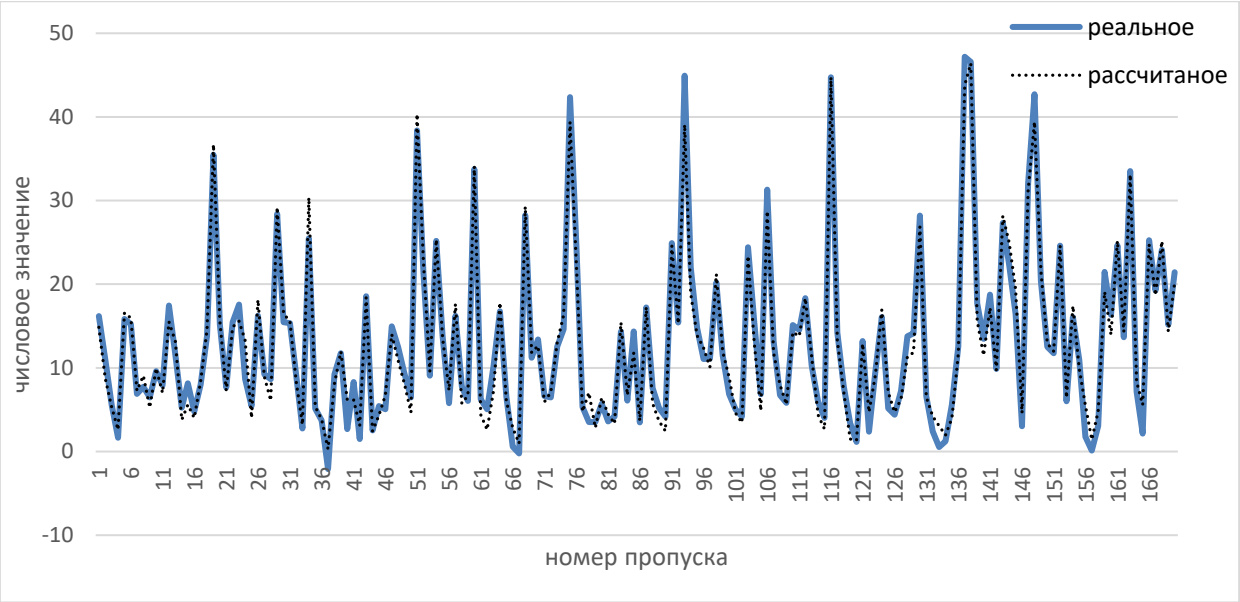

Рисунок 3.2.5 – 10% пропусков (10% ошибка)

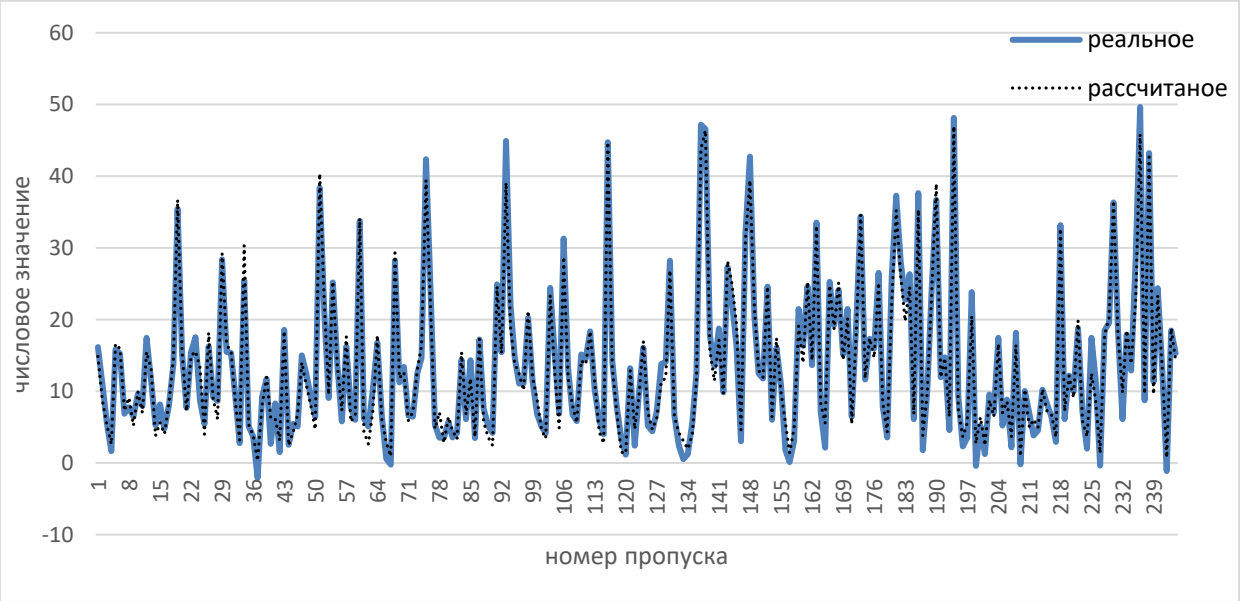

Рисунок 3.2.6 – 15% пропусков (10% ошибка)

Рисунок 3.2.7 – 5% пропусков (15% ошибка)

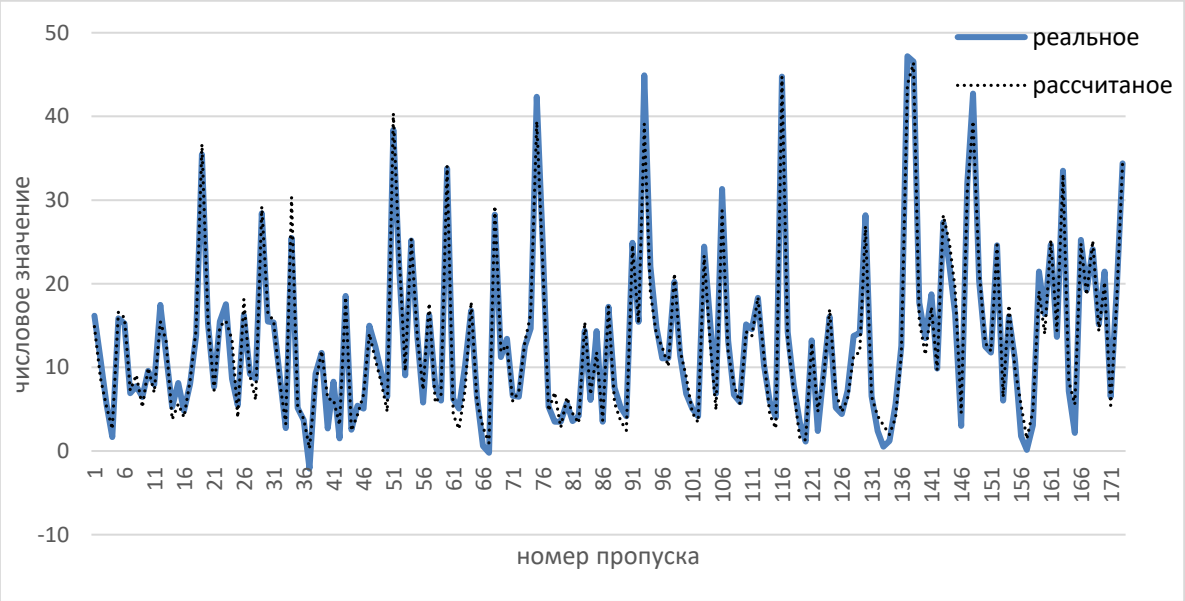

Рисунок 3.2.8 – 10% пропусков (15% ошибка)

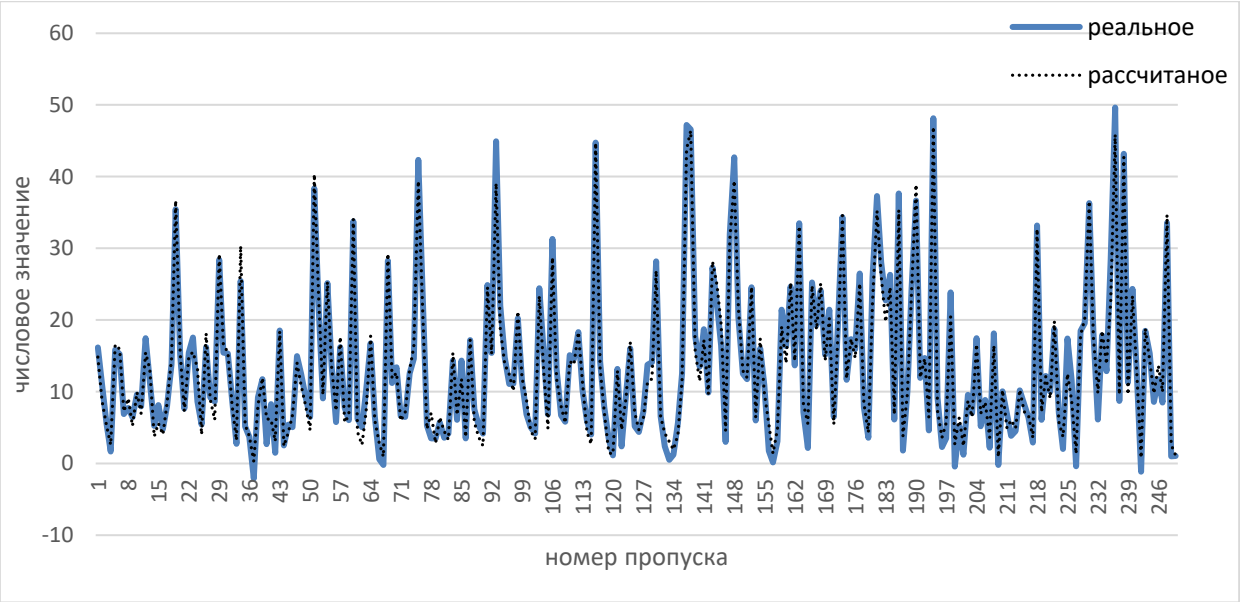

Рисунок 3.2.9 – 15% пропусков (15% ошибка)

Вывод: из рисунков видно, что оба графика во всех случаях практически совпадают, а так же алгоритм хуже всего рассчитывается в экстремальных значениях.

3.3 Сравнительная характеристика исследованных алгоритмов

Для сравнения алгоритмов, заполним пропуски одной и той же таблицы разными методами при 5% ошибке (процент пропусков и размеры таблицы будут варьироваться).

Таблица 3.3.1 – Сравнение результатов алгоритмов (5% ошибка)

К-во переменных	К-во строк	Процент пропусков	Погрешность	
			ZET	Непараметрический метод
6	300	5	0.222667	0.122514
6	300	10	0.125057	0.099978
6	300	15	0.128509	0.082236
10	300	5	0.196319	0.096305
15	300	5	0.255372	0.077828
5	500	5	0.147834	0.120144
5	800	5	0.119739	0.094995

Сравнение алгоритмов при 10% ошибке (процент пропусков и размеры таблицы будут варьироваться).

Таблица 3.3.2 – Сравнение результатов алгоритмов (10% ошибка)

К-во переменных	К-во строк	Процент пропусков	Погрешность	
			ZET	Непараметрический метод
6	300	5	0.179273	0.105311
6	300	10	0.123974	0.083745
6	300	15	0.067367	0.068995
10	300	5	0.086876	0.082154
15	300	5	0.154954	0.077854
5	500	5	0.124956	0.077416
5	800	5	0.109547	0.073748

Сравнение алгоритмов при 15% ошибке (процент пропусков и размеры таблицы будут варьироваться).

Таблица 3.3.3 – Сравнение результатов алгоритмов (15% ошибка)

К-во переменных	К-во строк	Процент пропусков	Погрешность	
			ZET	Непараметрический метод
6	300	5	0.173688	0.160725
6	300	10	0.126467	0.128466
6	300	15	0.130471	0.101137
10	300	5	0.142961	0.114611
15	300	5	0.168766	0.090821
5	500	5	0.135929	0.132322
5	800	5	0.112823	0.111645

Из таблиц видно, что погрешность уменьшается с возрастанием процента пропусков. Проанализируем, до какого значения эта зависимость будет действовать. Возьмем матрицу 6x300 с ошибкой 5%.

Таблица 3.3.4 – Зависимость погрешности от процента пропусков

Процент пропусков	Погрешность	
	ZET	Непараметрический метод
5	0.222667	0.122514
10	0.125057	0.099978
15	0.128509	0.082236
20	0.042356	0.036784
25	0.035056	0.035535
30	0.037316	0.037887

На 30% пропусков ZET алгоритм стал рассчитывать не все пустоты в таблице, а при непараметрическом методе увеличилась среднеквадратичная ошибка.

Из всех 3 таблиц видно, что ошибка непараметрического алгоритма заполнения пропусков либо приблизительно равна ошибке алгоритма ZET, либо меньше.

3.4 Моделирование

В данном пункте смоделированы значения некоторой функции и представлены в виде графиков. А так же показаны среднеквадратичные ошибки моделирования. Представлены два варианта:

1. с пропусками в столбце выхода;
2. с пропусками в столбце входа.

3.4.1 Вариант 1

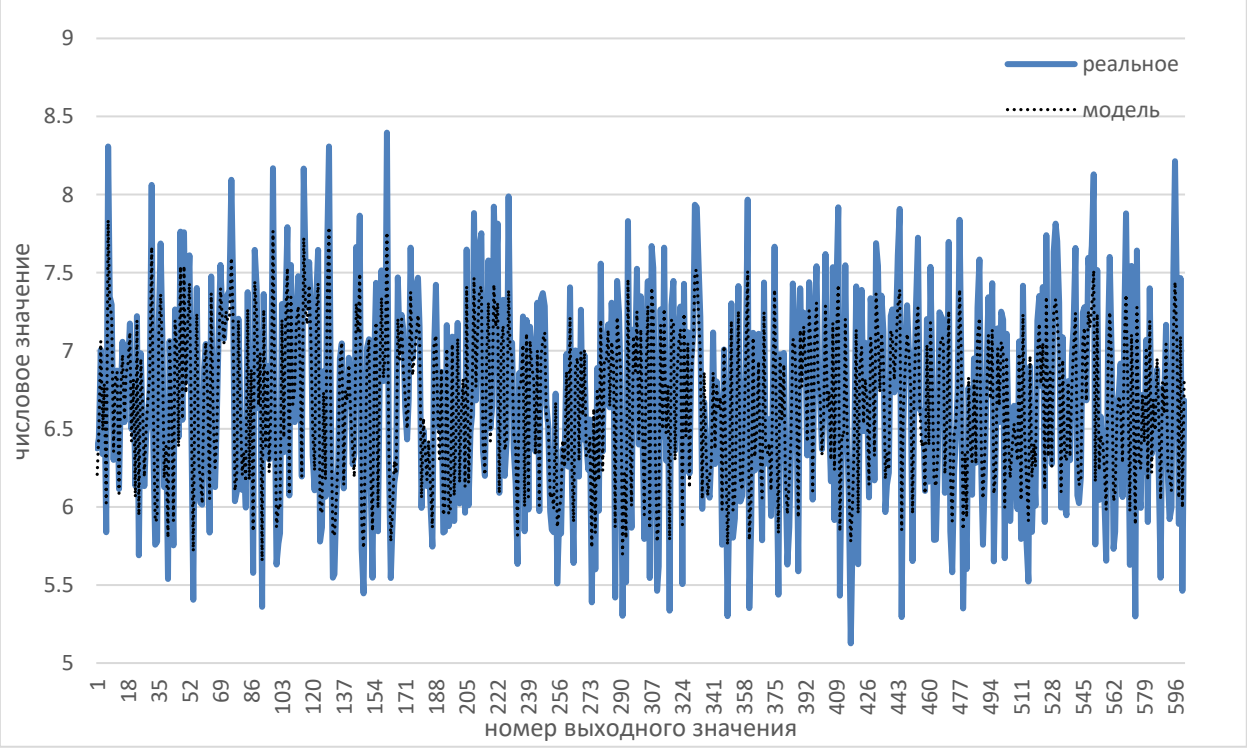

Рисунок 3.4.1.1 – Моделирование истинных значений выход

Построена модель по исходной выборке, среднеквадратичная ошибка моделирования при построении модели равна 0,00859.

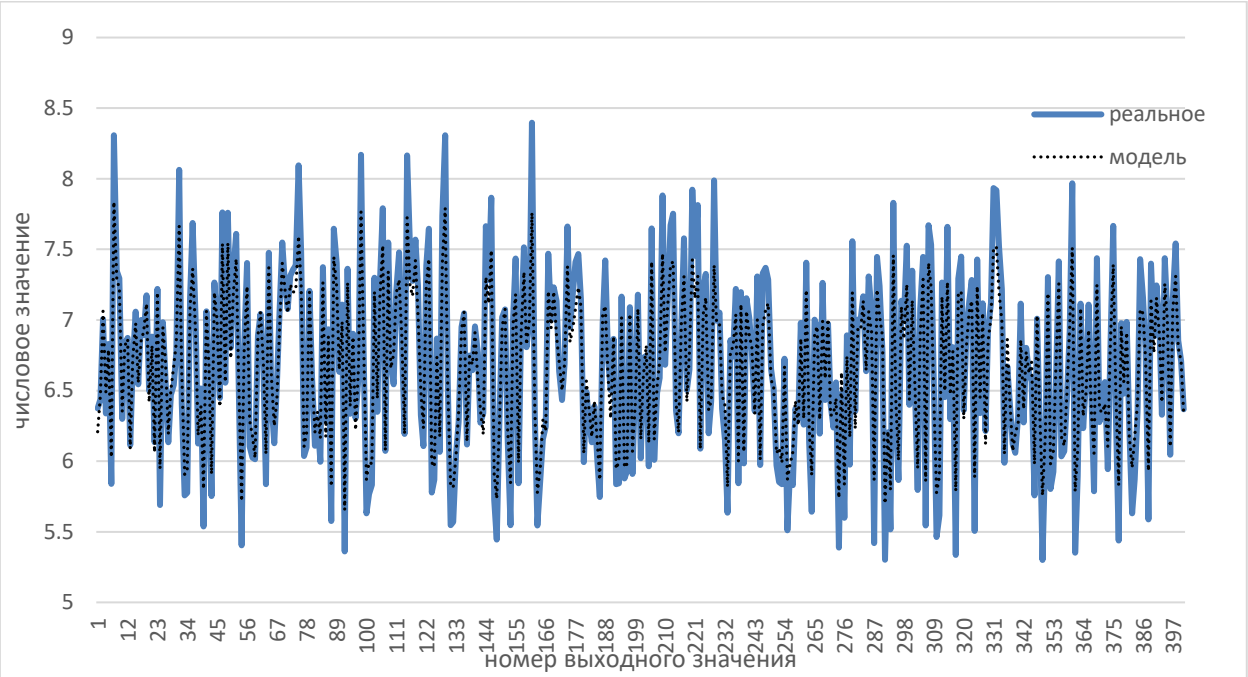

Рисунок 3.4.1.2 – Моделирование значение столбца выхода с незаполненными пропусками

Построена модель столбца выхода с незаполненными пропусками, среднеквадратичная погрешность моделирования 0.01021.

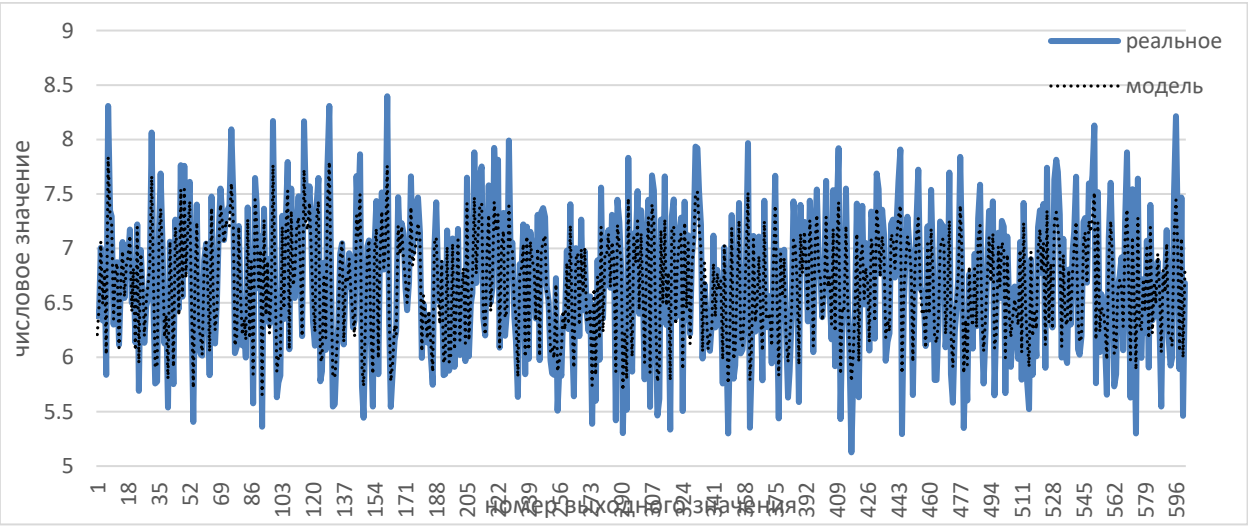

Рисунок 3.4.1.3 – Моделирование столбца с заполненными пропусками в столбце выхода

Построена модель столбца выхода с заполненными пропусками, среднеквадратичная погрешность моделирования 0.00881.

3.4.2. Вариант 2

Построим модель входных данных.

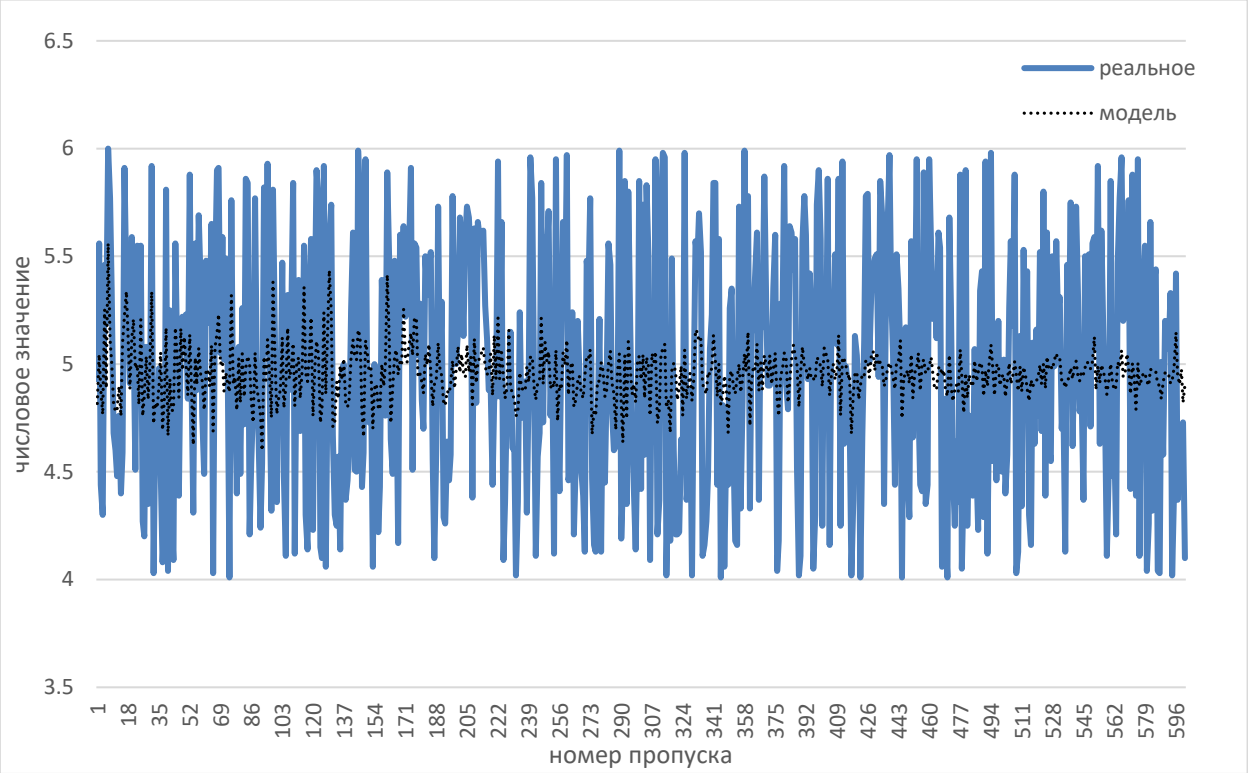

Рисунок 3.4.2.1 – Моделирование столбца с истинными значениями в столбце входа

Модель входа с истинными значениями оказалась менее точна, чем модель выхода. Среднеквадратичная погрешность моделирования 0.02.

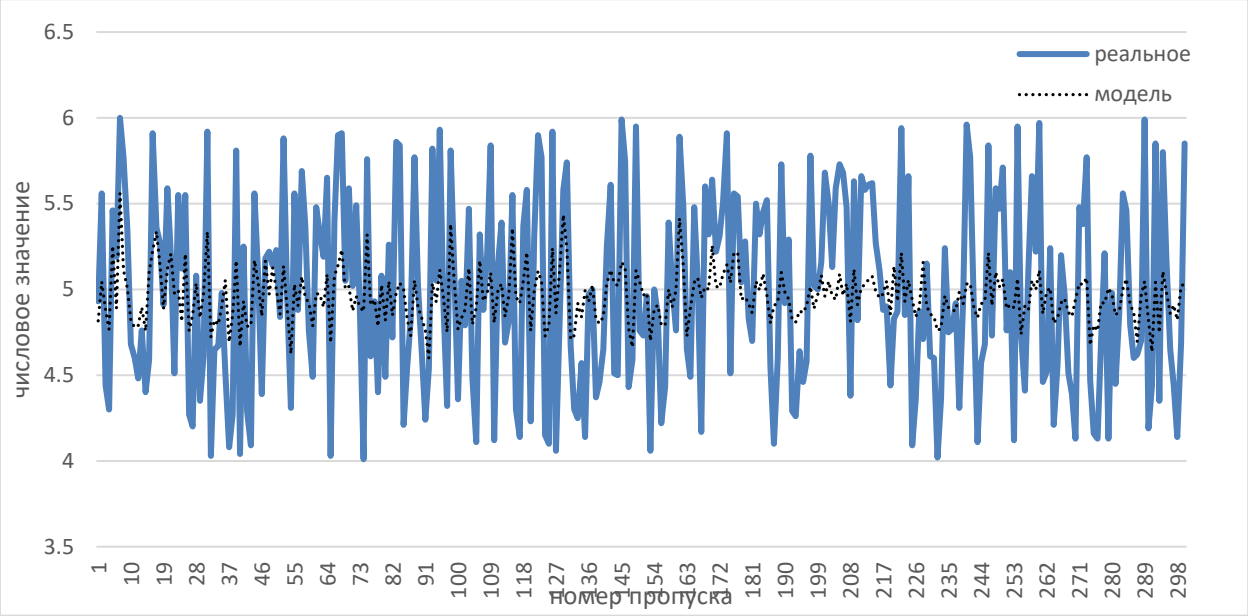

Рисунок 3.4.2.2 – Моделирование столбца входа с пропусками

Так же оказалась менее точна модель с незаполненными пропусками в столбце входа, по сравнению со значениями выхода. Среднеквадратичная погрешность равна 0.02846.

Погрешность 0.0205

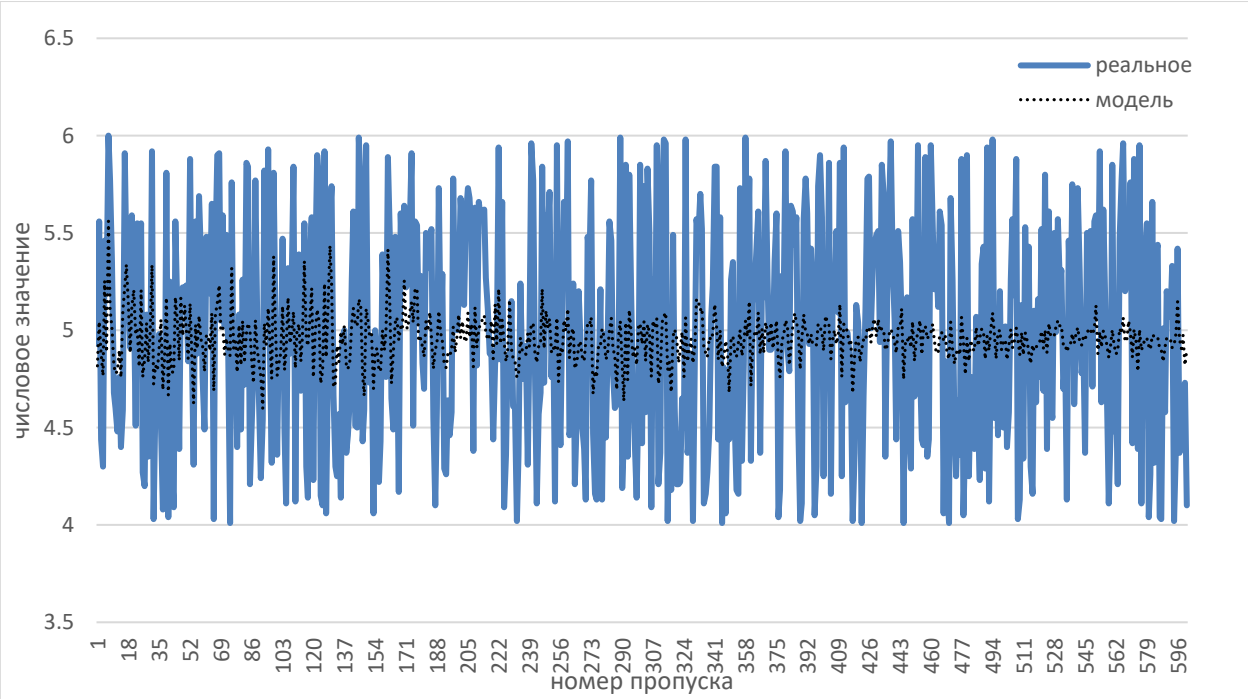

Рисунок 3.4.2.3 – Моделирование столбца входа с заполненными пропусками

Модель входных значений столбца с заполненными пропусками оказалась менее точная, чем модель выхода. Среднеквадратичная погрешность равна 0.0205.

Выводы по главе 3

Были рассчитаны среднеквадратичные погрешности матриц с пропусками, построены графики отклонения от истинного результата для методов ZET и непараметрического. Выявлено, что непараметрический метод рассчитывает значения пропусков с меньшей погрешностью, чем ZET алгоритм. Погрешности отличаются на сотые доли, а в некоторых случаях и на десятые. Выявлено, что погрешность уменьшается с возрастанием процента пропусков, но эта закономерность действует примерно до 30% пропусков, как для ZET алгоритма, так и для непараметрического метода заполнения пропусков. Так же выявлено, что оба алгоритма устойчивы к ошибке, это видно из таблиц. При возрастании количества переменных среднеквадратичная ошибка уменьшается у обоих методов.

Так же было выявлено, что погрешность моделирования построения модели, используя непараметрический алгоритм заполнения пропусков при заполненных пропусках, примерно равна погрешности модели истиной таблице.

ЗАКЛЮЧЕНИЕ

В данной выпускной квалификационной работе была поставлена цель решения задачи идентификации по выборке наблюдений с пропусками. В связи с этим рассматривается как уже существующий алгоритм заполнения пропусков, так и предлагаемый новый непараметрический алгоритм.

Для решения поставленных в работе задач был проведен обзор уже существующих алгоритмов восстановления пропусков (в данной работе ZET алгоритм). При исследовании эффективности работы алгоритма были рассчитаны среднеквадратичные погрешности при различных сочетания помех, процента пропусков, количества переменных и размера выборки. Выявлено, что ZET алгоритм сильно зависит от размера компетентной матрицы и количества пропусков в матрице наблюдений. Так же была решена задача идентификации по неполной выборке с помощью предлагаемого непараметрического алгоритма. Алгоритм позволил повысить точность ее решения.

Реализован новый алгоритм заполнения пропусков, проведено его исследование, а так же сравнительный анализ с ZET алгоритмом. Выявлено, что среднеквадратичная погрешность у ZET алгоритма выше, чем у предлагаемого алгоритма. Так же исследования показали, что задача идентификации по заполненной выборке решается точнее, чем по выборке с пропусками.

В итоге, основываясь на вышесказанном, предлагаемый алгоритм позволит повысить точность моделирования и может быть применен в таких областях как анализ и обработка данных, социология, промышленость, медицина и др.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Абдушукуров, А.А. Статистика неполных данных (асимптотическая теория оценивания для неклассических моделей) / А.А. Абдушукуров. – Ташкент : Университет, 2009. – 271 с.

2. Апраушева, Н.Н. Использование непараметрических оценок в регрессионном анализе / Н.Н. Апраушева, В.Д. Конаков // Заводск. лаб. – 1973. – № 5. – С. 556 – 569.

3. Афифи, А. Статистический анализ: подход с использованием ЭВМ / А. Афифи, С. Эйзен. – М.: Мир, 1982 – 488 с.

4. Бойко, Р. Непараметрические Н-модели процесса нагревания / Р. Бойко, Я. Демченко // Труды международной конференции «Applied methods of statistical analysis. Simulations and statistical inference». – 2011. – P.224 – 236.

5. Боровков, А.А. Математическая статистика. Оценка параметров. Проверка гипотез / А.А. Боровков. – М.: Наука, 1984. – 472 с.

6. Васильев, В.А. Непараметрическое оценивание функционалов от распределений стационарных последовательностей / В.А. Васильев, А.В. Добровидов, Г.М. Кошкин. – М.: Наука, 2004. – 508 с.

7. Вероятность и математическая статистика: Энциклопедия / Под.ред. Ю.В. Прохорова. – М.: Большая Российская энциклопедия, 2003. – 912 с.

8. Живоглядов, В.П. Непараметрические алгоритмы адаптации / В.П. Живоглядов, А.В. Медведев. – Фрунзе: Илим, 1974. – 133 с.

9. Заварин, А.Н. Использование априорной информации в непараметрических оценках функции регрессии / А.Н. Заварин // Автоматика и телемеханика. – 1985. – №5. – С.79 – 85.

10. Загоруйко, Н.Г. Прикладные методы анализа данных и знаний / Н.Г. Загоруйко. – Новосибирск: Издательство ИМ СО РАН, 1999. – 264 с.

11. Катковник, В.Я. Непараметрическая идентификация и сглаживание данных / В.Я. Катковник. – М.: Наука, 1985. – 336 с.

12. Лбов, Г.С. Методы обработки разнотипных экспериментальных данных / Г.С. Лбов. – Новосибирск: Наука, Сиб. отд-ние, 1981. – 160 с.

13. Литтл, Р.Дж.А. Статистический анализ данных с пропусками / Р.Дж.А. Литтл, Д.Б. Рубин. – М.: Финансы и статистика, 1990. – 336 с.

14. Медведев, А.В. Адаптация в условиях непараметрической неопределенности / А.В. Медведев // Адаптивные системы и их приложения. – Новосибирск: Наука, 1978. – С. 4 – 34.

15. Медведев, А.В. Непараметрические системы адаптации / А.В. Медведев. – Новосибирск: Наука, 1983. – 173с.

16. Медведев, А.В. Теория непараметрических систем. Процессы / А.В. Медведев // Вестник Сибирского государственного аэрокосмического университета имени академика М.Ф.Решетнева. – 2010. – №3 (29). – С. 4 – 9.

17. Медведев, А.В. Элементы теории непараметрических систем управления / А.В. Медведев // Актуальные проблемы информатики, прикладной математики и механики. Часть 3, Информатика. – Новосибирск-Красноярск: СО РАН, 1996. – С. 87 – 112.

18. Медведев, А.В. Теория непараметрических систем. Моделирование / А.В. Медведев // Вестник сибирского государственного аэрокосмического университета имени академика М.Ф. Решетнева. – 2010. – №4 (30). – С. 4 – 9.

19. Медведев, А.В. Теория непараметрических систем. Общий подход / А.В. Медведев // Вестник сибирского государственного аэрокосмического университета имени академика М.Ф. Решетнева. – 2008. – №4 (30). – С. 4 – 9.

20. Надарая, Э.А. Непараметрическое оценивание плотности вероятностей и кривой регрессии / Э.А. Надарая. – Город.: Издательство Тбилисского университета, 1983. – с.

21. Смоляк, С.А. Устойчивые методы оценивания: (Статистическая обработка неоднородных совокупностей) / С.А. Смоляк, Б.П. Титаренко. – М.: Статистика, 1980. – 208 с.

22. Тихонов, А.Н. Статистическая обработка результатов эксперимента / А.Н. Тихонов, М.В. Уфимцев. – М.: Изд-во Моск. ун-та, 1988. – 176 с.

23. Уилкс, С. математическая статистика / С. Уилкс. – М.: Наука, 1967. – 632 с.

24. Хардле, В. Прикладная непараметрическая регрессия / В. Хардле. – М.: Мир, 1993. – 349 с.

25. Цыпкин, Я.З. Адаптация и обучение в автоматических системах / Я.З. Цыпкин. – М.: Наука, 1968. – 400с.

26. Цыпкин, Я.З. Информационная теория идентификации / Я.З. Цыпкин. – М.: Наука. Физматлит, 1995. – 336 с.

27. Шуленин, В.П. Математическая статистика. Ч.2. Непараметрическая статистика: учебник / В.П. Шуленин – Томск: Изд-во НТЛ, 2012. – 388 с.

28. Эйкхофф, П. Основы идентификации систем управления / П. Эйкхофф. – М.: Мир, 1975. – 681 с.

29. Eddy, W.F. Optimum kernel estimators of the mode / W.F. Eddy // Ann. Math.Statist. – 1980. – V. 8. – P. 870 – 882.

30. Gasser, T. Kernel estimation of regression function / T. Gasser, H.G. Muller // Lect. Notes Math. – 1979. – V.757. – P. 23 – 68.

31. Marvin, L. Brown. Data mining and the impact of missing data / Marvin L. Brown, John F. Kros // Industrial Management & Data Systems. – 2003. – T.103. № 8. – P. 611 – 621.

32. Myrtyeit, I. Analyzing data sets with missing data: an empirical evaluation of imputation methods and likelihood-based methods / I. Myrtyeit, E. Stensrud, U.H. Olsson // IEEE Transactions on Software Engineering, 2001. – T. 27. № 11. – P. 999.

33. Литтл, Р.Дж.А., Рубин, Д.Б. (1991) Статистический анализ данных с пропусками. Финансы и статистика, Москва;

34. Эфрон, Б. (1988) Нетрадиционные методы многомерного статистического анализа. Финансы и статистика, Москва;

35. Дрейпер, Н., Смит, Г. (1988) Прикладной регрессионый анализ. Т.1,2. Машиностроение, Москва.

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
Базовая кафедра интеллектуальных систем управления

УТВЕРЖДАЮ
Заведующий кафедрой
_____ Ю. Ю. Якунин
«13» июня 2018 г.

БАКАЛАВРСКАЯ РАБОТА

27.03.03 «Системный анализ и управление»

Анализ заполнения пропусков в данных ZET-алгоритмом при решении задачи идентификации

Руководитель _____ доцент, к.т.н. А. А. Корнеева
подпись, дата — должность, ученая степень

Выпускник 08.06.2018 Е. А. Пермякова
подпись, дата

Красноярск 2018