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

Институт Космических и Информационных технологий
Кафедра «Информатика»

УТВЕРЖДАЮ
Заведующий кафедрой
_____ А. С. Кузнецов
подпись инициалы, фамилия
«____» _____ 20 __ г.

БАКАЛАВРСКАЯ РАБОТА

09.03.04 «Программная инженерия»

Разработка программного обеспечения для решения задачи идентификации при построении систем управления динамическими объектами

Руководитель	_____	_____	А. С. Михалев
	подпись, дата	должность, ученая степень	инициалы, фамилия
Выпускник	_____		Н. М. Луговая
	подпись, дата		инициалы, фамилия
Консультант	_____	_____	А. И. Рубан
	подпись, дата	должность, ученая степень	инициалы, фамилия
Нормоконтролер	_____		О. А. Антамошкин
	подпись, дата		инициалы, фамилия

Красноярск 2019

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

Институт Космических и Информационных технологий
Кафедра «Информатика»

УТВЕРЖДАЮ
Заведующий кафедрой
_____ А. С. Кузнецов
подпись инициалы, фамилия
«____» _____ 20 __ г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работе

Студенту Луговой Нине Михайловне

Группа КИ15-16Б Направление (специальность) 09.03.04 Программная инженерия

Тема выпускной квалификационной работы Разработка программного обеспечения для решения задачи идентификации при построении систем управления динамическими объектами

Утверждена приказом по университету № _____ от _____

Руководитель ВКР А. С. Михалев, старший преподаватель кафедры «Информатика»

Исходные данные для ВКР Проектирование, Разработка программного обеспечения для решения задачи идентификации при построении систем управления динамическими объектами

Перечень разделов ВКР Введение, Анализ предметной области, Проектирование архитектуры, Описание программной реализации, Анализ результатов работы, Заключение, Список использованных источников

Перечень графического материала Презентационные слайды PowerPoint

Руководитель ВКР _____ _____
подпись инициалы и фамилия

Консультант _____ _____
подпись инициалы и фамилия

Задание принял к исполнению _____
подпись, инициалы и фамилия студента

« ___ » _____ 20__ г.

РЕФЕРАТ

Выпускная квалификационная работа по теме «Разработка программного обеспечения для решения задачи идентификации при построении систем управления динамическими объектами» содержит 66 страниц текстового документа, 28 использованных источников, 31 таблицу, 40 иллюстраций.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, ИДЕНТИФИКАЦИЯ, МЕТОДЫ ИДЕНТИФИКАЦИИ, СТАТИЧЕСКИЕ И ДИНАМИЧЕСКИЕ ОБЪЕКТЫ, ПОСТРОЕНИЕ МОДЕЛИ, ПАРАМЕТРЫ, ИССЛЕДОВАНИЕ.

Целью данной выпускной квалификационной работы является разработка программного обеспечения для решения задачи идентификации при построении систем управления динамическими объектами.

Для достижения поставленной цели были решены следующие задачи:

- изучить и проанализировать предметную область;
- спроектировать архитектуру;
- разработать программное обеспечение, предназначенное для решения задач идентификации;

Также в рамках данной работы было проведено исследование метода последовательной линеаризации с использованием аналогов функций чувствительности.

В результате выполнения работы было разработано программное обеспечение для решения задач идентификации.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В настоящее время понятие модели используется во многих областях науки и техники, занимающихся решением сложных задач технологии, экономики, социологии, живой природы и прочих. Эти задачи возникают при изучении свойств и особенностей объектов с целью последующего управления, при создании адаптивных систем, в которых на основе построенной модели объекта вырабатываются оптимальные управляющие воздействия [1].

Основное содержание науки идентификации составляет построение математических моделей того или иного типа на основе результатов наблюдений за поведением объектов и исследование их свойств [2].

Различные типы моделей рассматриваемых объектов, систем или процессов используются на стадии создания систем управления этими объектами и на стадии их эксплуатации. Это обуславливает актуальность проблемы построения эффективных моделей объектов технических, технологических, экономических или социальных процессов.

Целью данной выпускной квалификационной работы является проектирование и разработка программного обеспечения для решения задачи идентификации при построении систем управления динамическими объектами.

Для достижения поставленной цели необходимо решить следующие задачи:

– изучить и проанализировать предметную область;

– спроектировать архитектуру;

– разработать программное обеспечение, предназначенное для решения задач идентификации.

Также в рамках данной работы было проведено исследование метода последовательной линеаризации с использованием аналогов функций чувствительности.

1 Анализ предметной области

1.1 Постановка задачи идентификации

Процесс идентификации складывается из двух взаимосвязанных этапов: идентификации структуры моделей и идентификации параметров в моделях выбранной структуры. При построении структуры модели (или набора конкурирующих либо взаимодополняющих структур) используется априорная информация об объекте. Для каждого класса объектов формируются банки структур с сопутствующей информацией [1].

Задача идентификации сводится, в общем случае, к определению оператора модели, преобразующего входные воздействия объекта в выходные величины. Оператор объекта является математической моделью объекта, и может быть определен в соответствующих пространствах функций. Задача идентификации объекта может иметь различные постановки, поскольку операторы могут характеризоваться разными структурой и характеристиками [2].

В большинстве реальных ситуаций взаимодействие объекта с окружающей средой соответствует следующей стандартной схеме (рисунок 1), где:

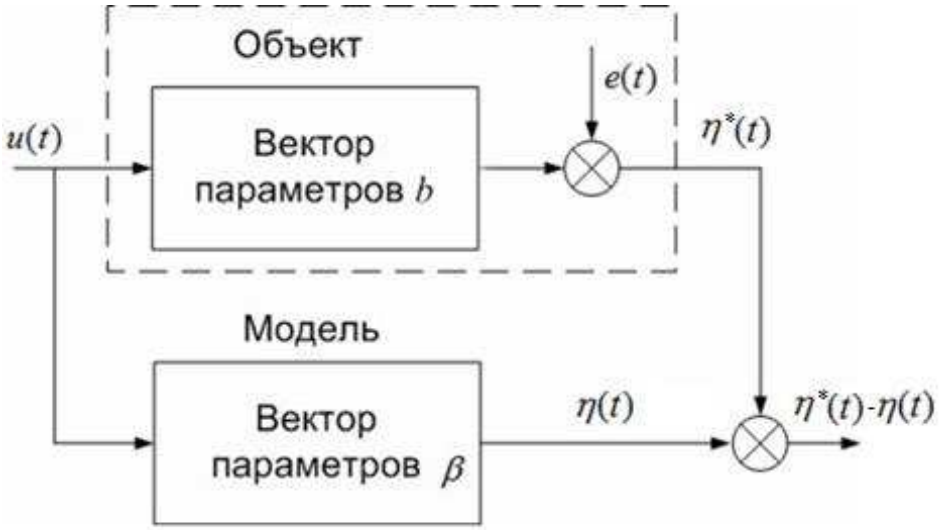

Рисунок 1 – Типовая схема наблюдения при идентификации объекта

– $u(t)$ – входное воздействие;

– $e(t)$ – неконтролируемое случайное воздействие;

– $\eta^*(t)$ – выходное воздействие объекта;

– $\eta(t)$ – выходное воздействие модели;

– $\eta^*(t) - \eta(t)$ – разность (невязка) между выходами объекта и модели;

– b – вектор параметров объекта;

– β – вектор параметров модели.

1.2 Математические модели систем

Существует огромное количество типов и классов моделей. Однако ни один из способов классификации не отражает всех свойств используемых моделей, т. к. характеризует только ее отдельные признаки.

Основными типами моделей, разделяющимися по основным системным признакам, являются:

– физические и математические;

– одномерные и многомерные;

– статические и динамические;

– детерминированные и стохастические;

– линейные и нелинейные;

– дискретные и непрерывные;

– стационарные и нестационарные;

– сосредоточенные и распределенные;

– характеристики типа «вход-выход» и описание в пространстве состояний;

– структурированные и агрегированные;

– параметрические и непараметрические [3].

Физическими являются модели, в которых свойства реального объекта представляются характеристиками вещественного объекта той же или

аналогичной природы. К математическим моделям относятся те, в которых для описания характеристик объекта используются математические конструкции. В дальнейшем будем рассматривать только математические модели.

Одномерными называют модели, имеющие один вход и один выход, многомерные (многосвязные) модели имеют несколько входов и несколько выходов.

Модель называется линейным, если для него справедлив принцип суперпозиции, т.е. реакция объекта на линейную комбинацию (суперпозицию) двух входных воздействий равна той же самой комбинации реакций данного объекта на каждое из воздействий:

$$f\big(\alpha u_1(t) + \beta u_2(t)\big) = \alpha f\big(u_1(t)\big) + \beta f(u_2(t)), \qquad (1.1)$$

где $u_1(t)$ и $u_2(t)$ – входные воздействия;

α и β – произвольные коэффициенты.

В противном случае модель считается нелинейным.

Параметрические модели описываются заданными в явной форме аналитическими зависимостями, содержащими параметры, подлежащие идентификации. Параметрами являются численные значения величин, определяющих выход модели (например, значения коэффициентов разностных уравнений). Непараметрические модели сводятся к описанию преобразований сигналов пространства входов в элементы пространства выходов [2].

В данной работе будут рассмотрены только параметрические модели.

Модели делятся на статические и динамические. Первые из них описывают объекты в стационарных режимах их работы [1].

Аналитически, в общем случае, уравнение модели статического объекта имеет вид нелинейной функции многих переменных. Во многих практических случаях общую нелинейную функцию удается параметризовать некоторым вектором a, тогда эта зависимость принимает вид

$$y = f(u, a), \tag{1.2}$$

где $u -$ вектор входных воздействий;

a – вектор неизвестных параметров.

В следствие чего задача идентификации сводится к определению неизвестных параметров a. Частным случаем параметрических моделей являются модели, линейные относительно оцениваемых параметров:

$$y = a_0 + \sum_i a_i f_i(u), \tag{1.3}$$

где $f_i(u) = f_i(u_1, u_2, \dots, u_n)$ – заданная система векторных линейно независимых функций.

Моделью статического линейного многомерного объекта с n входами и m выходами является система линейных алгебраических уравнений

$$\begin{cases} y_1 = a_{10} + a_{11}u_1 + a_{12}u_2 + \cdots + a_{1n}u_n \\ \dots \\ y_m = a_{m0} + a_{m1}u_1 + a_{m2}u_2 + \cdots + a_{mn}u_n \end{cases}, \tag{1.4}$$

где a_{ij} – неизвестные параметры модели, подлежащие определению.

Динамические модели описывают переходные процессы в объектах, например, возникающие при переходе с одного стационарного режима работы объекта на другой [1].

Линейные динамические модели могут принимать следующий вид:

1) Разностные уравнения

Универсальной характеристикой для дискретных моделей является разностное уравнение n-го порядка, где используется понятие разности как аналога понятию производной для непрерывных моделей:

$$a_0 y(k) + a_1 y(k-1) + \cdots + a_n y(k-n) = b_0 u(k) + b_1 u(k-1) + \cdots + \\ + b_m u(k-m), \quad (1.5)$$

где $y(k), u(k)$ – значения выходной и входной величин в k-ый момент времени при $k = 1,2\ldots$

в) Уравнения в пространстве состояний

Используя для описания динамики дискретного модели дискретные переменные состояния, образующие n-мерный вектор состояния $x(k) = [x_1(k), x_2(k), \ldots, x_n(k)]^T$, получают описание объекта в пространстве состояний в следующей векторно-матричной форме:

$$x(k) = [x_1(k), x_2(k), \ldots, x_n(k)]^T,\ x[k+1] = A[k]x[k] + B[k]u[k], \\ x[k_0] = x_0,\ k \geq k_0,\ y[k] = C[k]x[k] + D[k]u[k], \quad (1.6)$$

где $u[k] \in R^m$, $y[k] \in R^p$.

Матрицы $A[k], B[k], C[k], D[k]$ имеют размерности $[n \times n], [n \times m], [p \times n]$ и $[p \times m]$ соответственно.

б) Авторегрессионные модели со скользящим средним

При анализе стохастических систем исходные данные являются результатом цифровой обработки отдельных реализаций случайного процесса. В соответствии с этим, в современной теории цифровых систем получили широкое распространение цифровые параметрические стохастические модели авторегрессии и скользящего среднего (АРСС-модели). Эти модели используются для изучения временных рядов, определения статистических характеристик этих рядов и широко применяются в управлении, экономике, при обработке звуковых сигналов. Они достаточно просты, удобны в применении и обычно содержат небольшое число параметров, необходимых для процедуры оценивания.

в) Обыкновенные дифференциальные уравнения n-го порядка

$$a_n \frac{d^n y(t)}{dt^n} + a_{n-1} \frac{d^{n-1} y(t)}{dt^{n-1}} + \cdots + a_0 y(t) = b_m \frac{d^m u(t)}{dt^m} + \cdots + b_0 u(t), \qquad (1.7)$$

где $a_i,\ b_j,\ i = 0,1,\ldots n;\ j = 0,1,\ldots m$ – параметры модели, подлежащие идентификации.

Для большинства реальных физически реализуемых систем управления $m \leq n$.

б) Передаточные функции

Если к дифференциальному уравнению (1.7) задать нулевые начальные условия, то, применяя преобразование Лапласа, получают передаточную функцию линейного объекта в следующем виде:

$$W(p) = \frac{y(p)}{u(p)} = \frac{\sum_{i=0}^{m} b_i p^i}{\sum_{i=0}^{n} a_i p^i}, \qquad (1.8)$$

где p – комплексная переменная (параметр преобразования Лапласа).

В нелинейных динамических системах протекают многообразные явления и процессы, поэтому данный класс систем, по сравнению с линейными, значительно шире.

Если рассматривать математическую модель как существенно нелинейную, наиболее распространенными видами моделей являются следующие:

а) Нелинейные дифференциальные уравнения.

Для непрерывного одномерного объекта управления связь между входным и выходным сигналами записывается, в общем виде, следующим выражением:

$$F\left(y, \dot{y}, \ddot{y}, \ldots y^{(n)}, u, \dot{u}, \ddot{u}, \ldots u^{(m)}\right) = 0, \qquad (1.9)$$

где F – некоторый нелинейный оператор $(n + m + 1)$ аргумента, который требуется идентифицировать.

Если возможно, то проводится параметризация нелинейной модели (1.9) на основе структурирования F с введением некоторого вектора параметров β_i :

$$F\left(y, \dot{y}, \ddot{y}, \dots y^{(n)}, u, \dot{u}, \ddot{u}, \dots, u^{(m)}, \beta_1, \beta_2, \dots \beta_l\right) = 0, \quad (1.10)$$

где $\beta_i, i = 1 \dots l$ – параметры модели.

В этом случае задача идентификации сводится к определению оператора F и к оцениванию его вектора параметров $\beta_i, i = 1 \dots l$.

Для нелинейного дискретного объекта строятся аналогичные нелинейные разностные уравнения.

б) Модели Гаммерштейна.

Такие модели нелинейных инерционных объектов строятся в предположении, что нелинейность и инерционность объекта можно разделить и представить объект в виде последовательной комбинации двух звеньев: нелинейного безынерционного и динамического линейного. Модели «вход-выход» для таких объектов в одномерном стационарном случае:

$$y(t) = \int_0^\infty w(\tau) F[u(t - \tau)] d\tau, \quad (1.11)$$

где $w(t)$ – импульсная переходная функция линейного звена;

$F(u)$ – статическая характеристика нелинейного звена.

в) Разложение Вольтерра.

При данном способе описания зависимость между входом $u(t)$ и выходом $y(t)$ представляется рядом

$$y(t) = \int_0^t w_1(\tau)u(t-\tau)d\tau + \int_0^t \int_0^t w_2(\tau_1,\tau_2)u(t-\tau_1)u(t-\tau_2)d\tau_1 d\tau_2 + \cdots, \quad (1.12)$$

где $w_1(\tau), w_2(\tau_1,\tau_2), w_3(\tau_1,\tau_2,\tau_3)$ – обобщенные весовые функции (ядра) i-го порядка.

Такой ряд (1.12) носит название ряда Вольтерра. Разложение в ряд Вольтерра является непосредственным обобщением линейной модели в форме интеграла свертки на нелинейные объекты. Задача идентификации при этом состоит в определении обобщенных весовых функций $w_i(\tau_1,\tau_2,\dots,\tau_i), i = 1,2,\dots$. Для нестационарного объекта ядра будут зависеть от t.

г) Описание в пространстве состояний.

В общем случае, уравнения состояния для конечномерных непрерывных систем записываются в следующем виде:

$$\frac{dx}{dt} = f(x(t),u(t),t); x(t_0) = x_0; t \geq t_0; y(t) = g(x(t),u(t),t). \quad (1.13)$$

Первое уравнение – уравнение состояния – описывает изменение состояния системы во времени в зависимости от начального условия в момент времени t_0 и входного воздействия $u(t)$. Второе уравнение – уравнение выхода – устанавливает связь между текущими значениями состояния и входа, с одной стороны – и выхода $y(t)$ – с другой [2].

В данной работе рассматриваются линейные и нелинейные статические и динамические объекты. Для описания этих объектов будут использованы разностные уравнения. Стоит также отметить, что данный тип записи был выбран в качестве основного для программной реализации, поскольку его преимущество заключается в простоте структуры уравнений.

1.3 Описание реализуемых алгоритмов

В процессе разработки программного обеспечения было реализовано четыре метода идентификации. Рассмотрим каждый из них.

1) Метод наименьших квадратов при линейной параметризации модели.

Модель объекта задана в виде линейной комбинации известных (базисных) функций $\varphi_1(u(t)), \dots, \varphi_m(u(t))$.

$$\eta(u(t), \alpha) = \sum_{j=1}^{m} \varphi_j(u(t))\, \alpha_j = \varphi^T(u(t))\alpha = \alpha^T \varphi(u(t)), \tag{1.14}$$

где α – матрица неизвестных параметров $\begin{pmatrix} \alpha_1 \\ \dots \\ \alpha_j \end{pmatrix}$,

$\varphi(u)$ – матрица базисных функций $\begin{pmatrix} \varphi_1(u(t)) \\ \dots \\ \varphi_j(u(t)) \end{pmatrix}$.

Вектор-столбец значений выхода модели (в моменты времени $i = 1, .., n$) имеет вид

$$H(\alpha) = \begin{pmatrix} \varphi^T(u_1(t))\alpha \\ \dots \\ \varphi^T(u_n(t))\alpha \end{pmatrix} = \begin{pmatrix} \varphi^T(u_1(t)) \\ \dots \\ \varphi^T(u_n(t)) \end{pmatrix} \alpha =$$

$$= \begin{pmatrix} \varphi_1(u_1(t)) & \varphi_2(u_1) & \dots & \varphi_m(u_1(t)) \\ & \dots & & \\ \varphi_1(u_n(t)) & \varphi_2(u_n(t)) & \dots & \varphi_m(u_n(t)) \end{pmatrix} \alpha = \Phi\alpha. \tag{1.15}$$

Параметры α находим по критерию наименьших квадратов:

$$I(\alpha) = (H^* - \Phi\alpha)^{-T} K^{-1} (H^* - \Phi\alpha) = \min_{\alpha}, \tag{1.16}$$

где K^{-1} – матрица коэффициентов корреляции.

Необходимое условие минимума $\nabla I = 0$ (где ∇I – градиент от $I(\alpha)$ по α) приводит к системе линейных алгебраических уравнений

$$\Phi^T K^{-1}\Phi\alpha = \Phi^T K^{-1}H^*. \quad (1.17)$$

Система имеет единственное решение, если матрица $\Phi^T K^{-1}\Phi$ невырождена [4].

Запишем решение системы:

$$\alpha = (\Phi^T K^{-1}\Phi)^{-1}\Phi^T K^{-1}H^* = N_0 H^*. \quad (1.18)$$

Для частного случая, когда помеха ξ некоррелирована и измерения равноточные: $K = \sigma^2 E$ (где E – единичная матрица), $K^{-1} = \sigma^{-2}E$ – система уравнений (1.17) приобретает вид

$$\sigma^{-2}\Phi^T\Phi\alpha = \sigma^{-2}\Phi H^*, \quad (1.19)$$

где σ^{-2} – матрица дисперсий.

2) Простейший адаптивный алгоритм.

Данный алгоритм относится к самым первым применяемым в адаптации алгоритмам [5]. В силу его предельной простоты и часто неплохих свойств его применяют и в настоящее время либо в первозданном виде, либо с небольшими модификациями. Последовательно остановимся на подстройке параметров линейных и нелинейных моделей.

Для линейной параметризации модели $\eta(u(t), \alpha)$ на каждой итерации, например, n и $n-1$, параметры модели находим из условия равенства выходов модели и объекта:

$$\eta_n^* = \varphi^T(u_n(t))\alpha_n, \eta_{n-1}^* = \varphi^T(u_{n-1}(t))\alpha_{n-1}. \quad (1.20)$$

Алгоритм расчета α_n:

$$\alpha_n = \alpha_{n-1} + \frac{(\eta_n^* - \varphi^T(u_n(t))\alpha_{n-1})}{\varphi^T(u_n(t))\alpha_n}\varphi(u_n(t)) = \alpha_{n-1} + \left(\eta_n^* - \right.$$
$$\left. - \varphi^T(u_n(t))\alpha_{n-1}\right)\left(\varphi^T(u_n(t))\right)^+. \quad (1.21)$$

Здесь знак «+» – псевдообращения (обобщенного обращения) матрицы [8].

Если есть помеха, то из формулы (1.21) получаем, что

$$\alpha_n \xrightarrow[n\to\infty]{} a + \frac{\xi_n}{\varphi^T(u_n)}\varphi(u_n(t)), \quad (1.22)$$

где ξ_n – помеха.

Чувствительность к помехам можно уменьшить, вводя в алгоритм (1.21) дополнительный положительный параметр [5]. Его вводят двумя способами.

Первый способ:

$$\alpha_n = \alpha_{n-1} + \gamma\frac{(\eta_n^* - \varphi^T(u_n(t))\alpha_{n-1})}{\varphi^T(u_n(t))\alpha_n}\varphi(u_n(t)), \gamma > 0, n = 1, 2, ..., \quad (1.23)$$

где γ – дополнительный положительный параметр.

Второй способ:

$$\alpha_n = \alpha_{n-1} + \frac{(\eta_n^* - \varphi^T(u_n(t))\alpha_{n-1})}{\gamma + \varphi^T(u_n(t))\alpha_n}\varphi(u_n(t)), \gamma > 0, n = 1, 2, ... \quad (1.24)$$

Второй способ часто оказывается более предпочтительным, ибо параметр γ осуществляет регуляризацию алгоритма, когда $\varphi^T\varphi$ приближается к нулю.

3) Метод последовательной линеаризации (МПЛ).

Этот метод (метод минимизации функционала) включает в себя на каждой итерации две стадии [6]. На первой из них строим квадратичную аппроксимацию

(относительно траектории) минимизируемого функционала. На второй стадии в квадратичную аппроксимацию функционала (либо в сам функционал, если он является квадратичным, например, $I(\alpha)$ в критерии наименьших квадратов) подставляем линейную аппроксимацию выхода модели по искомым параметрам (либо по их приращениям). После этих двух этапов на каждой из итераций в явном виде вычисляем приращения параметров. Затем по ним находим параметры на следующей итерации из условия монотонной сходимости минимизируемого функционала. Эти итерации повторяются последовательно, и получаемая последовательность параметров сходится к искомому решению.

Построим итерационную процедуру расчета параметров α модели в соответствии с критерием наименьших квадратов [1].

Выход модели при значениях параметров α^l на l-ой итерации обозначаем через $\eta^l(t)$. Квадратичная аппроксимация функционала равна

$$I(\alpha) \approx \sum_{t\in T}\left[F\left(e^l(t)\right)\left|e^l(t)\right|^{-2}\right]\left(e^l(t)\right)^2 \equiv \sum_{t\in T} k^l(t)\left(e(t)\right)^2,$$
$$k^l(t) = \begin{cases} 1, \left|e^l(t)\right| < \Delta \\ \left|e^l(t)\right|^{-1}\Delta, \Delta \leq \left|e^l(t)\right| \end{cases}, \qquad (1.25)$$

где $e^l(t)$ – разность выхода объекта и выхода модели $(\eta^*(t) - \eta^l)$;

Δ – заданное бесконечно малое число.

Подставляем линейную аппроксимацию выхода модели в правую часть уравнения в квадратичный функционал:

$$\eta(\mathrm{t}) \approx \eta^l(t) + \mathrm{W}^l(t)\Delta\alpha^{l+1} = \eta^l(t) + (\frac{\partial\eta^l}{\partial y}V^l(t) + \frac{\partial\eta^l}{\partial\alpha})\Delta\alpha^{l+1}, \qquad (1.26)$$

где $\mathrm{W}^l(t)$ – вектор-строка функций чувствительности (ФЧ) для скалярного выхода модели относительно параметров α;

$\mathrm{V}^l(t)$ – матрица ФЧ для вектора $y(t)$ относительно вектора параметров.

Далее решаем обычную задачу наименьших квадратов:

$$I(\alpha) \approx \bar{I}(\Delta\alpha^{l+1}) = \sum_{t\in T} k^l(t)\left(\mathrm{W}^l(t)\right)^T (\eta^*(t) - \eta^l(t) - \mathrm{W}^l(t)\Delta\alpha^{l+1})^2 =$$

$$= \min_{\Delta\alpha^{l+1}} \qquad (1.27)$$

относительно приращения параметров $\Delta\alpha^{l+1}$:

$$\left(\sum_{t\in T} k^l(t)\left(\mathrm{W}^l(t)\right)^T \mathrm{W}^l(t)\right)\Delta\alpha^{l+1} = \sum_{t\in T} k^l(t)\left(\mathrm{W}^l(t)\right)^T e^l(t). \qquad (1.28)$$

Если выход модели $\eta(t)$ векторный, то это $\mathrm{W}^l(t)$ матрица ФЧ.

По приращениям $\Delta\alpha^{l+1}$ находим следующее приближение параметров по формуле:

$$\alpha^{l+1} = \alpha^l + \gamma^l\alpha^{l+1}, l = 0,1,2,\dots,\gamma^l > 0, \qquad (1.29)$$

где γ^l – положительный коэффициент, являющийся членом убывающего ряда.

Выбором скалярной величины γ^l добиваемся монотонной сходимости по функционалу качества (2): $I^{l+1} \le I^l$.

4) Метод последовательной линеаризации с использованием аналогов функций чувствительности.

Трудности, с которыми сталкиваются при реализации МПЛ, состоят в следующем:

1. При большом числе подстраиваемых параметров и сложной структуре динамических уравнений рассматриваемых объектов затруднено аналитическое получение уравнений чувствительности (УЧ) с соответствующими начальными условиями и моделями переключения для функций чувствительности (ФЧ), а также численное решение УЧ.

2. Алгоритмы МПЛ обеспечивают только локальный спуск в минимум функционала качества.

3. Проблемы состоят и в выборе исходного приближения параметров, ибо оно должно находиться в окрестности глобального экстремума функционала качества.

Для преодоления указанных трудностей целесообразно использовать численные оценки ФЧ, получаемые обработкой результатов поисковых движений по искомым параметрам.

В данной работе вектор – строку $\mathrm{W}^l(t)$ (либо матрицу) ФЧ будем доопределять из условия наилучшей линейной аппроксимации (1.26) за счет обработки результатов пробных движений параметров относительно точки α^l . При этом не надо аналитически составлять большое число УЧ для $\mathrm{V}^l(t)$ и их численно решать. За счет этого меньше вносится ошибок в алгоритмы, а процедуры расчета становятся более универсальными. При малых амплитудах пробных отклонений по α матрица $\mathrm{W}^l(t)$ является оценкой матрицы ФЧ; при больших отклонениях она уже не является таковой и служит только формальным аналогом ее. Применение линейной аппроксимации (1.26) (при малых и больших пробных отклонениях, т. е., говоря сокращенно, в малом и в большом) снимает проблему корректности линейного приближения в теории управления [7], ибо всегда линейное приближение будет корректным.

В каждый момент времени t из множества T в гиперпрямоугольной области $\alpha_v^l - \delta\alpha_v^l \le \alpha_v \le \alpha_v^l + \delta\alpha_v^l$, с центром в точке α^l равномерно (по случайному механизму, с помощью оптимальной ЛП τ-последовательности, по ортогональному плану первого порядка и др.) распределяем n пробных точек:

$$\alpha_v^{(i)} = \alpha_v^l + \delta\alpha_v^l u_v^{(i)}, v = \overline{1,m}, i = \overline{1,n}, \qquad (1.30)$$

где $u_v^{(i)}$ – распределенные (по выбранному закону) в интервале [-1; 1] числа;

m – количество дифференцируемых параметров.

В пробных точках вычисляем выходы динамической системы $\eta^{(i)}(t)$ и их уклонения от $\eta^{l}(t)$:

$$\delta\eta^{(i)}(t) = \eta^{(i)}(t) - \eta^{l}(t)\ . \tag{1.32}$$

На основе них строим n линейных уравнений:

$$\sum_{v=1}^{m} \delta\alpha_v^l u_v^{(i)} W_v^l\ (t) = \delta\eta^{(i)}(t), i = \overline{1,n}. \tag{1.33}$$

В матричном виде формула (1.33) выглядит как:

$$W^l(t)A = \delta\eta(t). \tag{1.34}$$

Система (1.33) служит для расчета вектор строки ФЧ $W^l(t)$ линейной аппроксимации (1.26). В (1.34) $\delta\eta(t)$ это вектор строка, включающая значения . При векторном выходе модели в системе уравнений (1.34) $\delta\eta(t)$ и $W^l(t)$ становятся матрицами. Решение переопределенной системы линейных уравнений находим по методу наименьших квадратов

$$W^l(t) = \delta z(t)A^{+}, \tag{1.35}$$

где A^{+} – псевдообратная матрица [8] по отношению к A.

Основной целью подстройки параметров является минимизация квадратичного функционала, поэтому пошаговую перестройку интервалов варьирования $\delta\alpha_v, v = \overline{1,m}$, проводим так же, как в экстремальных задачах [9] с использованием непараметрического сглаживания.

Для каждого t генерирование пробных движений (используемых при расчете W) проводим по одному и тому же алгоритму, т. е. последовательность $u_v^{(i)},\ i = \overline{1,n}, v = \overline{1,m}$ для каждого t одна и та же. При этом вычислительные

затраты будут минимальны, ибо значения выходов $\eta^{(i)}(t)$ используются при расчете $W(t)$ и по ним формируются n значений целевого функционала: $I^{(i)},\ \ i=\overline{1,n}$. В результате имеем [9, 10], что

$$\delta\alpha_v^{l+1}=\delta\alpha_v^l\gamma_p|\bar{u}_v|^p,\ \bar{u}_v=\sum_{i=1}^n\overline{K}^{(i)}\,u_v^{(i)},\ v=\overline{1,m},\ \gamma_p>1,\ p=1,2,\ldots$$

$$\overline{K}^{(i)}=\frac{K^{(i)}}{\sum_{j=1}^n K^{(j)}},\ K^{(i)}=K((I^*-I^{(i)})/(I_{max}-I_{min})), \qquad (1.36)$$

где $K(\cdot)$ – колоколообразные, симметричные относительно нуля, неотрицательные ядра;

$I_{max}=\max\{I^{(i)},i=\overline{1,n}\}$;

$I_{min}=\min\{I^{(i)},i=\overline{1,n}\}$;

$I^*=I_{min}$;

p – фиксированная целочисленная величина;

γ_p – величина, лежащая в диапазоне [1; 3] для $p=1,2$.

Заметим, что рабочее движение к экстремуму, особенно на первых шагах, с целью выхода в окрестность глобального экстремума, можно осуществлять на основе использования непараметрических оценок регрессии [9] между α и I:

$$\alpha_v^{l+1}=\alpha_v^l+\delta\alpha_v^l\bar{u}_v,,v=\overline{1,m}. \qquad (1.37)$$

Исходная область варьирования параметров (при $l=0$) должна включать в себя всю область возможных изменений этих параметров.

При выборе интервалов варьирования можно ориентироваться на величины рабочих шагов $\gamma^l\Delta\alpha^{l+1}$, получаемых по МПЛ.

2 Проектирование архитектуры

2.1 Используемые инструменты разработки

Выбор средств разработки программного обеспечения является важным этапом работы над проектом, так как использование тех или иных языков программирования, интегрированных средств разработки, а также готовых библиотек может оказать существенное влияние на развитие проекта.

Грамотный выбор инструментов разработки позволит избежать проблем в том случае, если при реализации функционала программного продукта будет обнаружено, что выбранная библиотека содержит в себе только часть необходимого функционала, а использование других библиотек не представляется возможным ввиду того, что они не поддерживаются используемыми языками программирования. В результате чего придется либо реализовать недостающий функционал самостоятельно, чему будут сопутствовать временные затраты, дополнительные накладки производительности, непредвиденные ошибки, дополнительное тестирование, либо заново производить выбор средств разработки и начинать с чистого листа.

Для разработки программного комплекса был выбран язык программирования C#, поскольку он обладает следующими преимуществами:

– Низкая сложность разработки и сопровождения.

– Наличие подробной документации языка;

– Большое количество открытых библиотек, исполняемых программ, литературы и технической помощи (MSDN [11] и другие источники);

– Высокая скорость работы при распределении процессов;

– Высокая скорость работы с данными;

– Большое количество доступных сред разработки.

Так как было необходимо разработать Desktop-приложение, в качестве платформы разработки была выбрана WPF. Windows Presentation Foundation (WPF) – это платформа пользовательского интерфейса для создания клиентских приложений для настольных систем. Платформа разработки WPF поддерживает

широкий набор компонентов для разработки приложений, включая модель приложения, ресурсы, элементы управления, графику, макет, привязки данных, документы и безопасность [12].

Поскольку разрабатываемая программная система должна учитывать добавление новых модулей и алгоритмов, основной чертой архитектуры является гибкость. Так как преимущество модульной архитектуры заключается в возможности обновления (замены) модуля, без необходимости изменения остальной системы, для разработки было выбрано модульное программирование с использованием платформы Prism.

Prism – это платформа для создания слабосвязанных, поддерживаемых и тестируемых приложений XAML в WPF, Windows 10, UWP и Xamarin Forms [13]. Prism предоставляет реализацию набора шаблонов проектирования, которые полезны при написании хорошо структурированных и поддерживаемых приложений XAML, включая MVVM, внедрение зависимостей, команды и прочее.

Для реализации модуля методов идентификации была выбрана библиотека Math.NET Numerics [14]. Целью Math.NET Numerics является предоставление методов и алгоритмов для численных расчетов в науке, технике и повседневном использовании. Данная библиотека использовалась для проведения операций с матрицами.

Для реализации парсера введенных выражений объекта была использована библиотека mXparser [15].

Для реализации графического интерфейса была использована библиотека Material Design In XAML[16]

В качестве интегрированной среды разработки для языка C# была выбрана бесплатная версия Visual Studio Community 2017 [17]. Данная IDE содержит весь необходимый функционал: средства анализа кода, графический отладчик, подсветку синтаксиса и ошибок, рефакторинг, удобную навигацию по проекту. В тоже время Visual Studio обладает простым графическим интерфейсом и не занимает большое количество аппаратных ресурсов компьютера.

2.2 Проектирование модульной архитектуры

Модульная архитектура в разработанном программном обеспечении представлена следующими модулями:

– ExportAndInport;

– IdentificationMethods;

– INavigationModule;

– NavigationModule;

– ObjectStructure;

– Core;

– CreateComponentsWindow;

– InputObject;

– StartResearchWindow;

– StartWindow;

– TaskResearchWindow;

– TaskSolutionWindow;

– NotifyPropertyChange.

Ниже представлена схема взаимодействия всех вышеописанных модулей (рисунок 2).

Модули Core, InputObject, StartResearchWindow, StartSolutionWindow, TaskResearchWindow, TaskSolutionWindow предназначены для отображения страниц приложения и являются библиотеками классов. Все перечисленные модули, кроме Core, соответствуют шаблону проектирования MVVM ((Model-View-ViewModel) [18].

Модуль Core является связующим и реализует базовую модель представления.

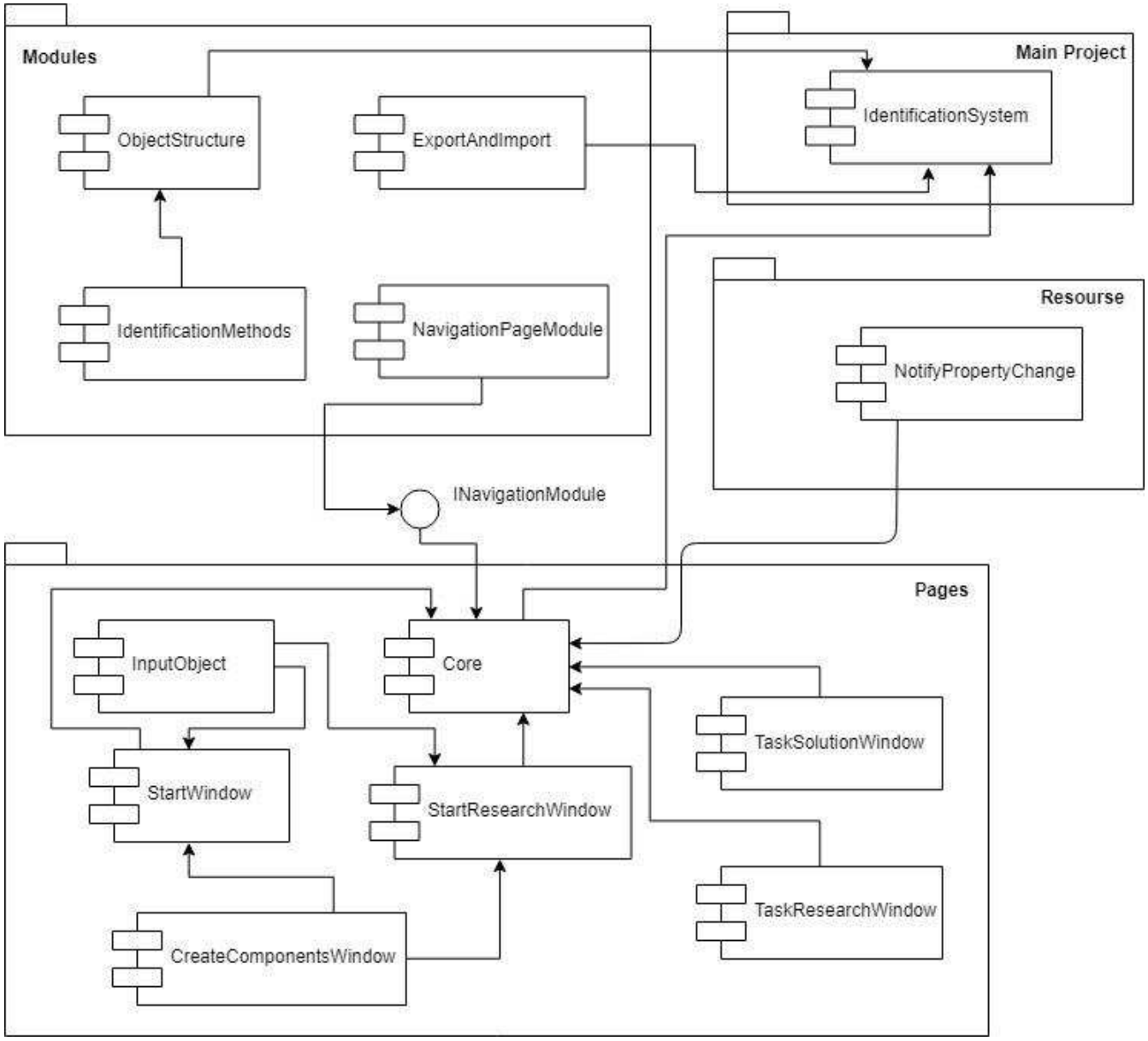

Рисунок 2 – Схема взаимодействия модулей

Модуль StartWindow реализует первую страницу приложения. Он отвечает за создание, открытие и сохранение задачи, выбор метода идентификации, формирование структуры объекта.

Модуль InputObject отвечает за парсинг введенного выражения и создания объекта.

Модуль CreateComponentsWindow предоставляет возможность создавать компоненты и компоновать их в нужном порядке.

Модуль TaskSolutionWindow представляет собой UserControl, отвечающий за ввод значений параметров объекта, начальных параметров модели, параметров метода, а также вывод полученного решения, как в табличном, так и в графическом виде.

Модули StartResearchWindow и TaskResearchWindow отвечают за отображение интерфейса исследовательской части ПО. По своей реализации они похожи на модули StartWindow и TaskSolutionWindow. Их отличие заключается в том, что данные модули предназначены для исследования и сравнения нескольких выбранных алгоритмов, а не изучения одного алгоритма в ходе решения поставленной задачи.

Модули ExportAndInport, IdentificationMethods, INavigationModule, NavigationModule, ObjectStructure являются библиотеками классов и используются как сервисы, необходимые для хранения промежуточных данных, работы с файлами, реализации методов идентификации и прочее.

Модуль NavigationModule реализует интерфейс из INavigationModule и отвечает за навигацию (содержит в себе инициализацию UserControls, добавление элементов в регион, метод NavigateTo и пр.).

Модуль ObjectStructure содержит в себе объектную структуру задачи и представляет собой набор классов, используемых во всех основных модулях. Подробная информация о структуре данного модуля находится в пункте 2.3.

Модуль ExportAndInport отвечает импорт и экспорт решаемой задачи в формате .xml.

Модуль IdentificationMethods содержит в себе методы идентификации, унаследованные от класса Method, а также параметры методов, родителем которых является класс MethodParameters.

Модуль NotifyPropertyChange является ресурсом и реализует интерфейс INotifyPropertyChange, предназначенный для уведомления клиента об изменении значения свойства.

2.3 Проектирование объектной архитектуры

Класс Task является основным и представляет собой отображение поставленной задачи в системе. Данный класс содержит в себе такие как свойства как Object (объект типа класса Object), Model (объект типа класса Model) и Method (объект типа класса Method). Такая структура обозначает, что у поставленной задачи имеется объект, на основе которого рассчитываются параметры модели выбранным методом идентификации. Классы Object и Model наследуются от класса Equation, так мы напрямую наследуем структуру объекта у модели. Данный класс содержит в себе как строковое, так и компонентное представление уравнения. Для компонентного представления уравнения используются класса ComponentParameter, ComponentInput, ComponentOutput.

Объект содержит в себе свойство класса Hindrance. Данный класс отвечает за генерацию помехи и содержит все необходимые свойства и поля для данного метода. Моменты времени являются общим свойством как для объекта, так и для модели. В каждый момент времени генерируются входы объекта и модели, рассчитываются выходы, меняются значения параметров модели.

Класс Method является родительским для всех методов идентификации, а класс MethodParameters – родительским для всех классов с параметрами метода.

Класс TaskForReseach содержит в себе список моделей, объект и список методов, выбранных для сравнения и исследований. Данный класс предназначен для сравнения методов идентификации.

На рисунке 3 представлена UML диаграмма классов [19].

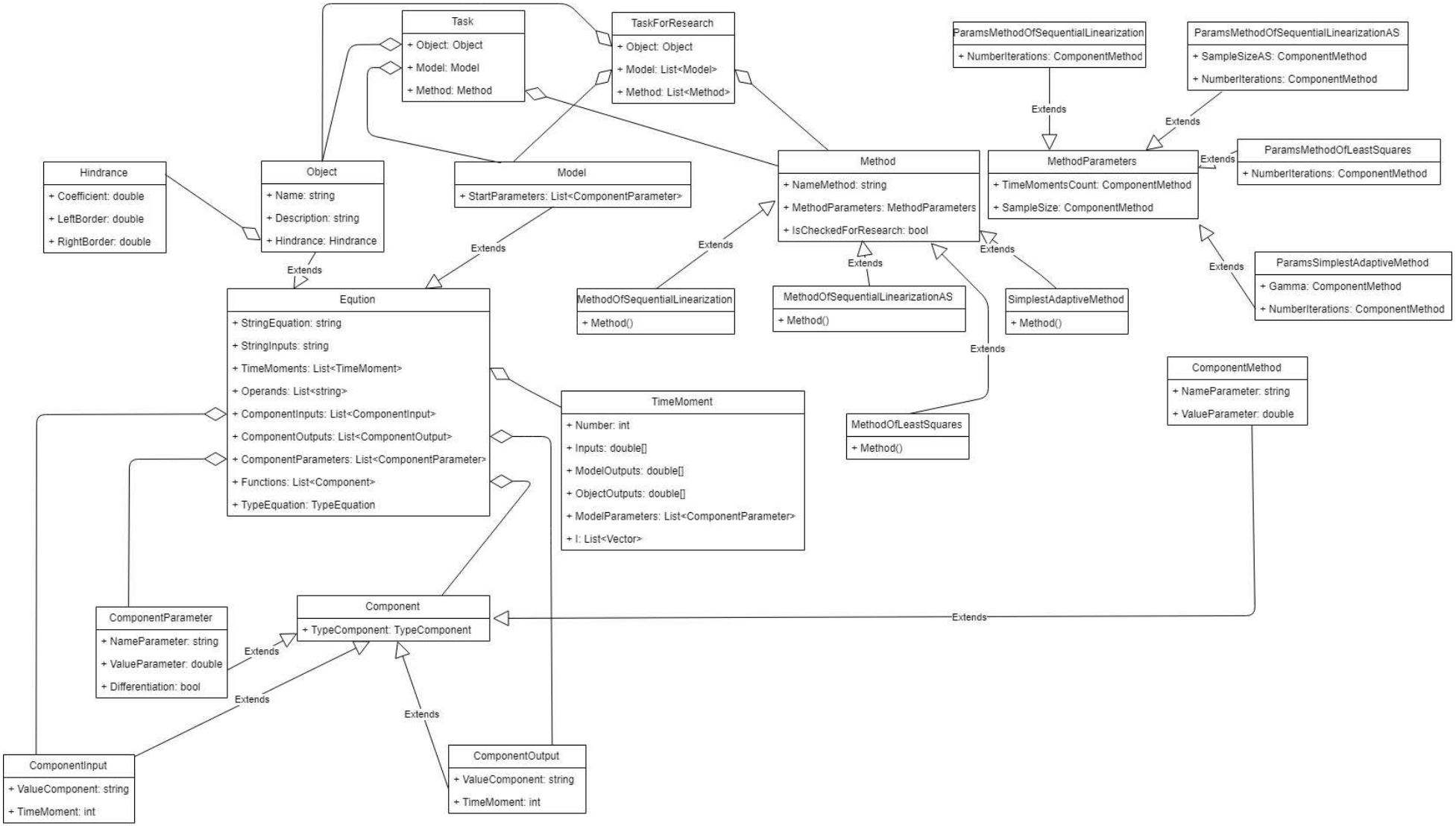

Рисунок 3 – UML-диаграмма классов

3 Описание программной реализации

Разработанный программный комплекс логически можно разделить на две части. Первая часть используется для решения задачи идентификации, вторая – для сравнения алгоритмов.

В обоих режимах пользователю предоставляется возможность создавать новую задачу, открывать уже существующую задачу, сохранять открытую или новую задачу (рисунок 4).

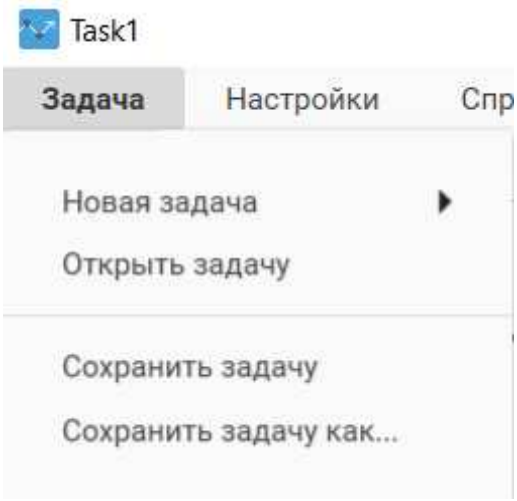

Рисунок 4 – Операции работы с задачей

Данные операции хранятся в главном проекте и взаимодействуют с методами модуля ExportAndImport. Модуль с помощью классов OpeningTask и SavingTask предоставляет возможность открывать и сохранять задачи в формате .xml (рисунок 5).

Операция «Новая задача» предназначена для создания задачи выбранного типа. При выборе одного из двух вариантов, представленных на рисунке 6, автоматически будет выбран нужный режим работы программы.

Если пользователь решит закрыть программу, не сохранив задачу, появится сообщение, представленное на рисунке 7.

```
<Task>
  <Object>
    <Name>Сложный объект</Name>
    <Description>Объект создан для тестирования метода №4</Description>
    <StringEquation>a*y+b*z+c*x+d*w+(l+f+g)*(h*u+i*u*u+j*u*u*u+k*u*u*u*u)</StringEquation>
    <StringInputs>u</StringInputs>
    <Functions>
      <ComponentParameter>
        <Name>a</Name>
        <Value>1</Value>
        <Differentiation>false</Differentiation>
      </ComponentParameter>
      <ComponentParameter>
        <Name>y</Name>
        <Value>-0.5</Value>
        <Differentiation>true</Differentiation>
      </ComponentParameter>
      <ComponentParameter>
        <Name>b</Name>
        <Value>2</Value>
        <Differentiation>false</Differentiation>
      </ComponentParameter>
      <ComponentParameter>
        <Name>z</Name>
        <Value>-0.35</Value>
        <Differentiation>true</Differentiation>
      </ComponentParameter>
      <ComponentParameter>
        <Name>c</Name>
        <Value>3</Value>
        <Differentiation>false</Differentiation>
      </ComponentParameter>
      <ComponentParameter>
        <Name>x</Name>
        <Value>0.6</Value>
        <Differentiation>true</Differentiation>
      </ComponentParameter>
      <ComponentParameter>
        <Name>d</Name>
        <Value>4</Value>
        <Differentiation>false</Differentiation>
      </ComponentParameter>
      <ComponentParameter>
        <Name>w</Name>
```

Рисунок 5 – Фрагмент файла сохраненной задачи в формате .xml

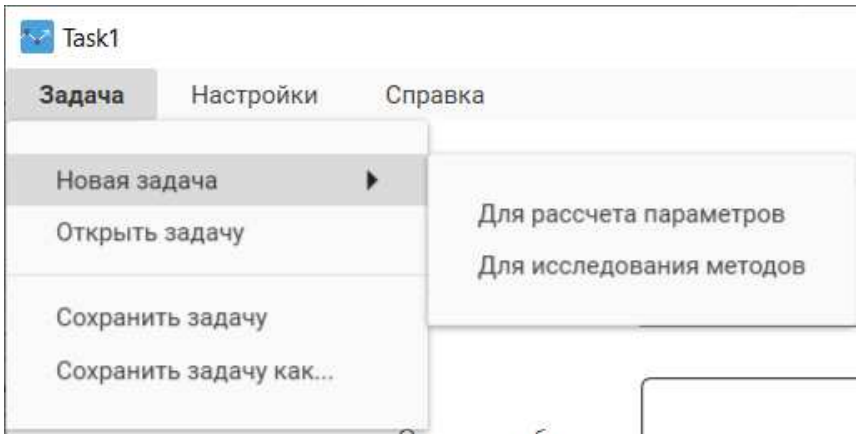

Рисунок 6 – Выбор типа задачи

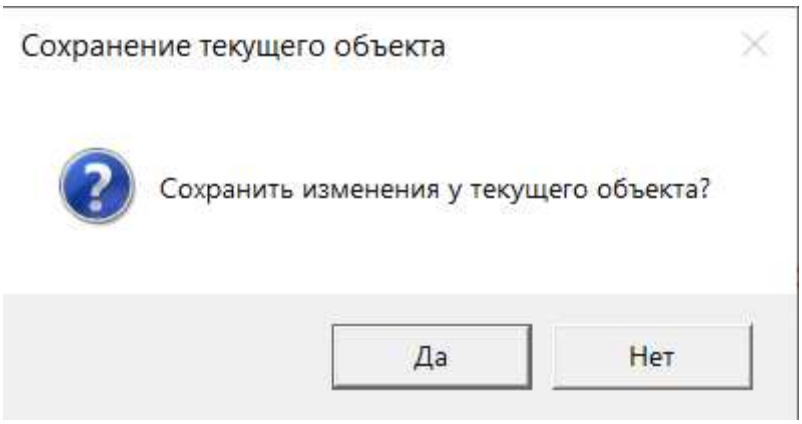

Рисунок 7 – Предложение сохранить задачу перед закрытием

Если при открытии задачи пользователь по ошибке выберет файл неподходящего формата или же файл нужного формата, но не содержащего задачу, появятся сообщения об ошибке (рисунки 8 и 9).

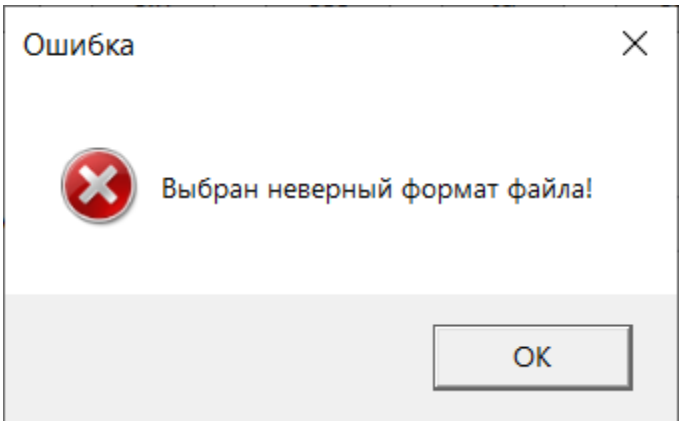

Рисунок 8 – Сообщение выборе файла неподходящего формата

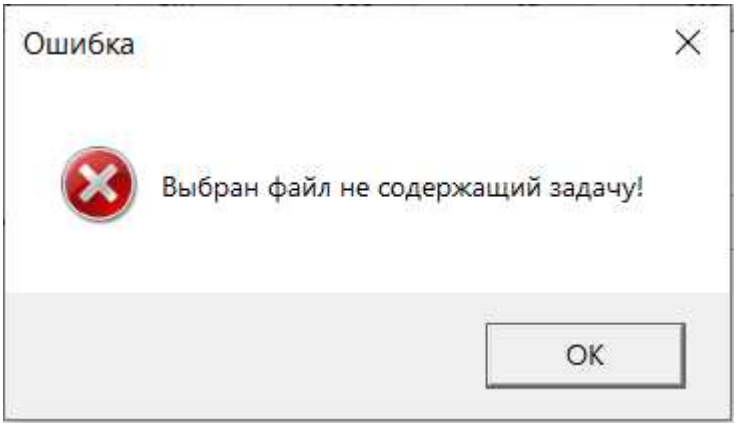

Рисунок 9 – Сообщение о выборе файла, не содержащего задачу

Если задача была ранее открыта, то операция «Сохранить задачу» сохранит все внесенные изменением, иначе пользователю будет необходимо выбрать место для сохранения (рисунок 10).

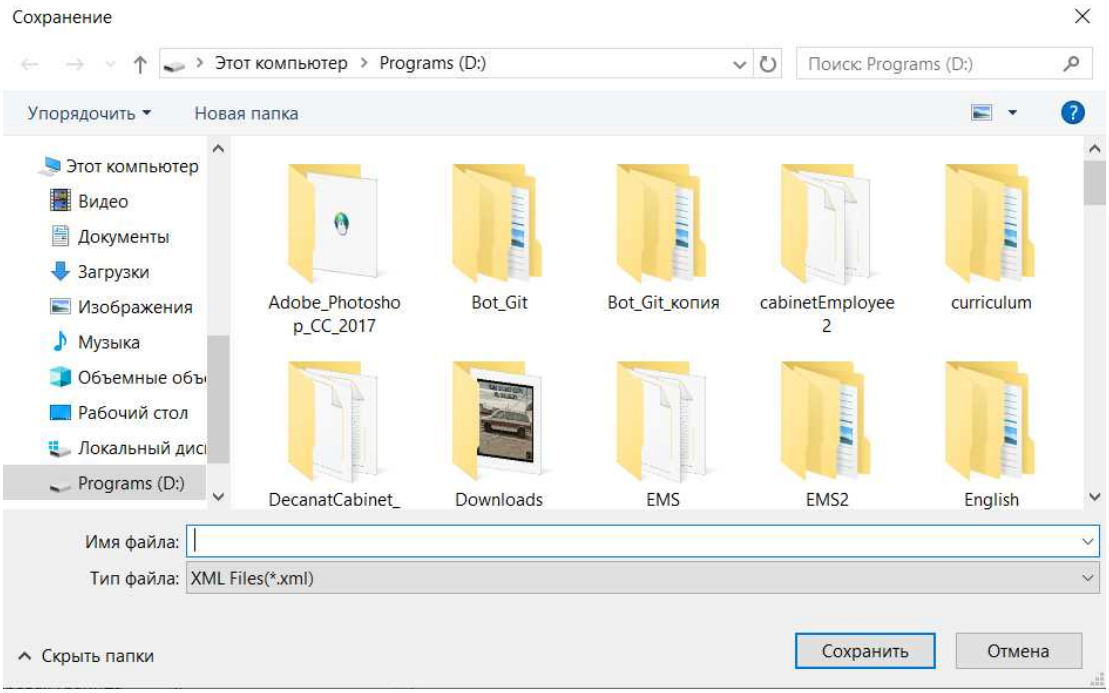

Рисунок 10 – Выбор места для сохранения задачи

Задача в формате файла является отражением класса Task и также хранит в себе объект, модель и метод.

Удобство данных операций очевидно, поскольку для исследования может быть выбран объект, содержащий большое количество параметров. Пользователю не придется вводить их значения каждый раз при запуске программы.

При запуске программного комплекса открывается новая задача Task1. По умолчанию будет запущен режим нахождения решения задачи идентификации. В настройках (рисунки 11 и 12) можно установить, какой из режимов будет запускаться по умолчанию.

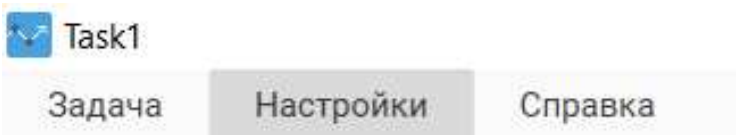

Рисунок 11 – Настройки

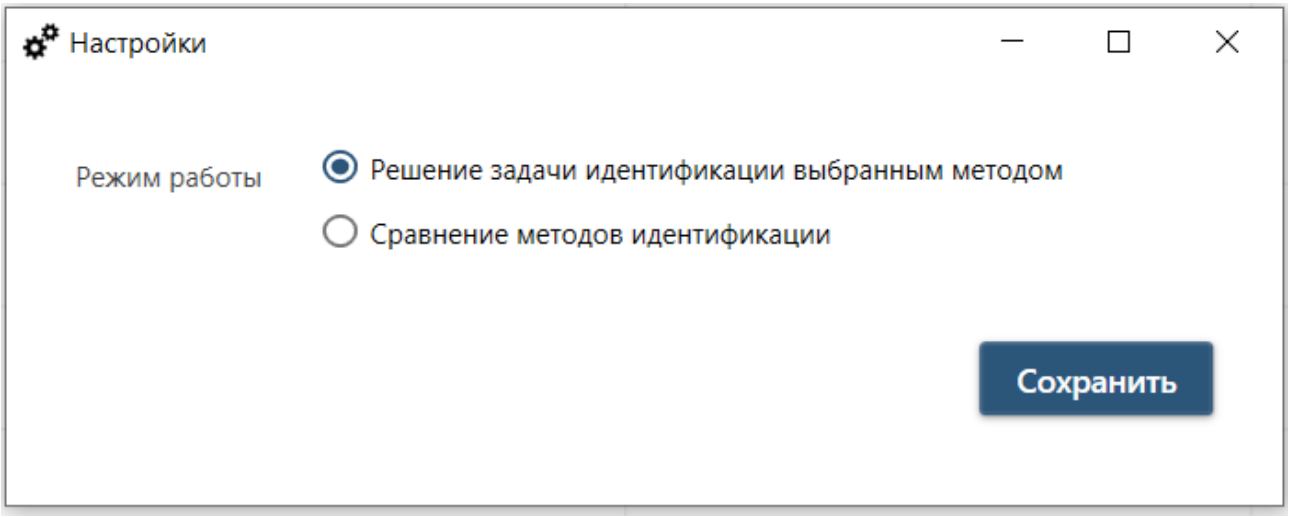

Рисунок 12 – Выбор режима

Просмотр справки осуществляется при нажатии на «Справка», далее «Просмотреть справку» (рисунок 13). Будет открыт документ с пользовательской документацией в браузере (рисунок 14).

Для просмотра сведений о программе необходимо нажать на «Справка», далее «О программе» (рисунок 15).

Рисунок 13 – Просмотр справки

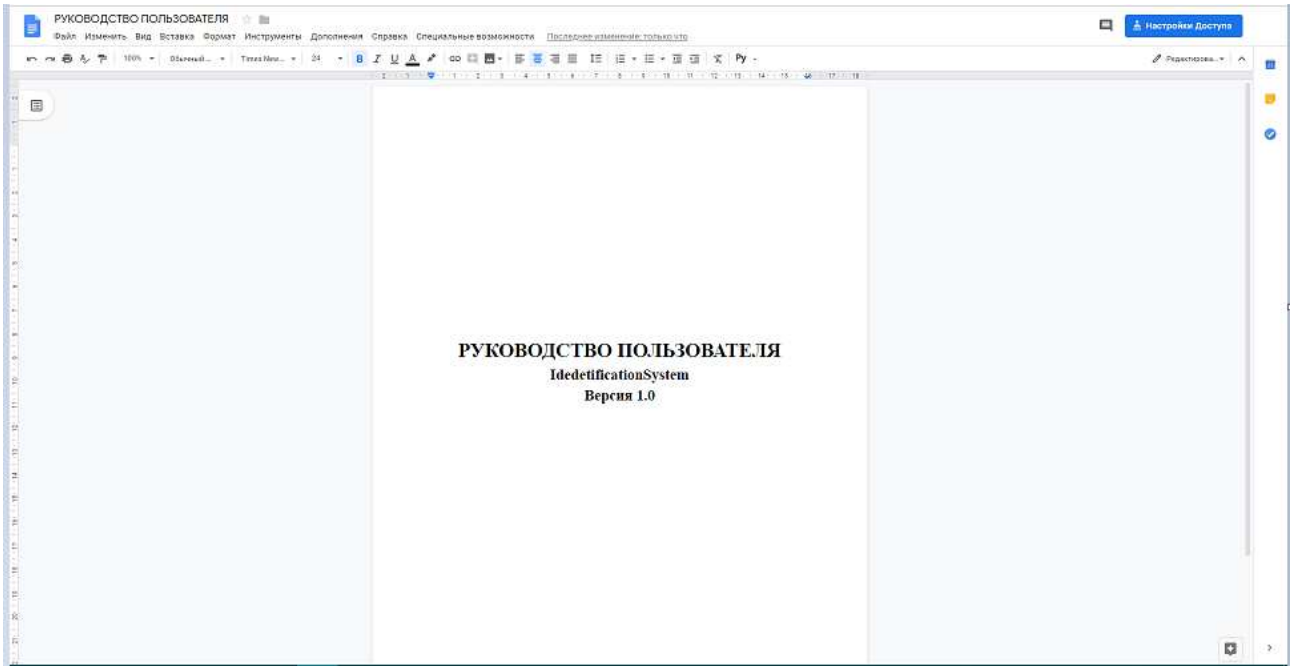

Рисунок 14 – Просмотр справки

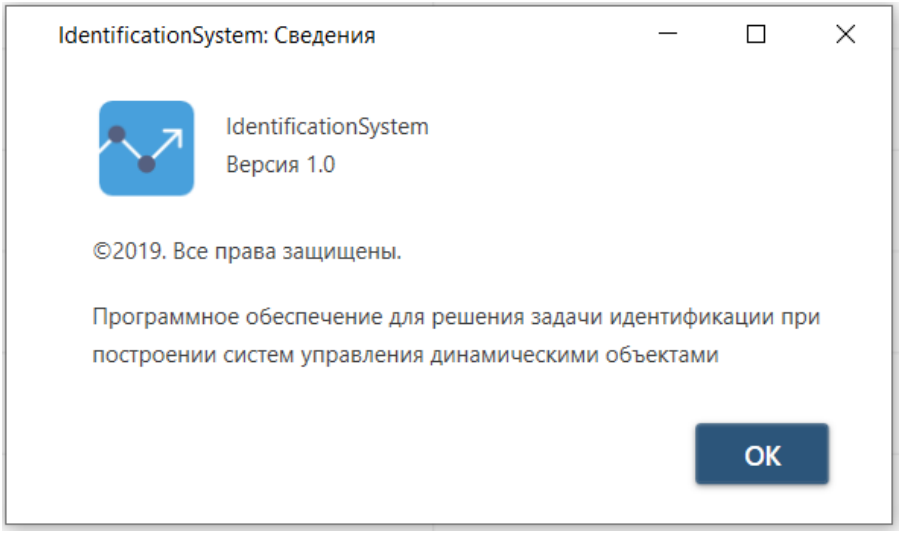

Рисунок 15 – «О программе»

Далее рассмотрим режим работы программного обеспечения, используемый для нахождения параметров модели выбранным методом идентификации.

При запуске программы открывается окно, в котором необходимо ввести уравнение объекта (рисунок 16) и выбрать метод идентификации (рисунок 17).

Вкладка объект содержит в себе UserControl из проекта InputObject.

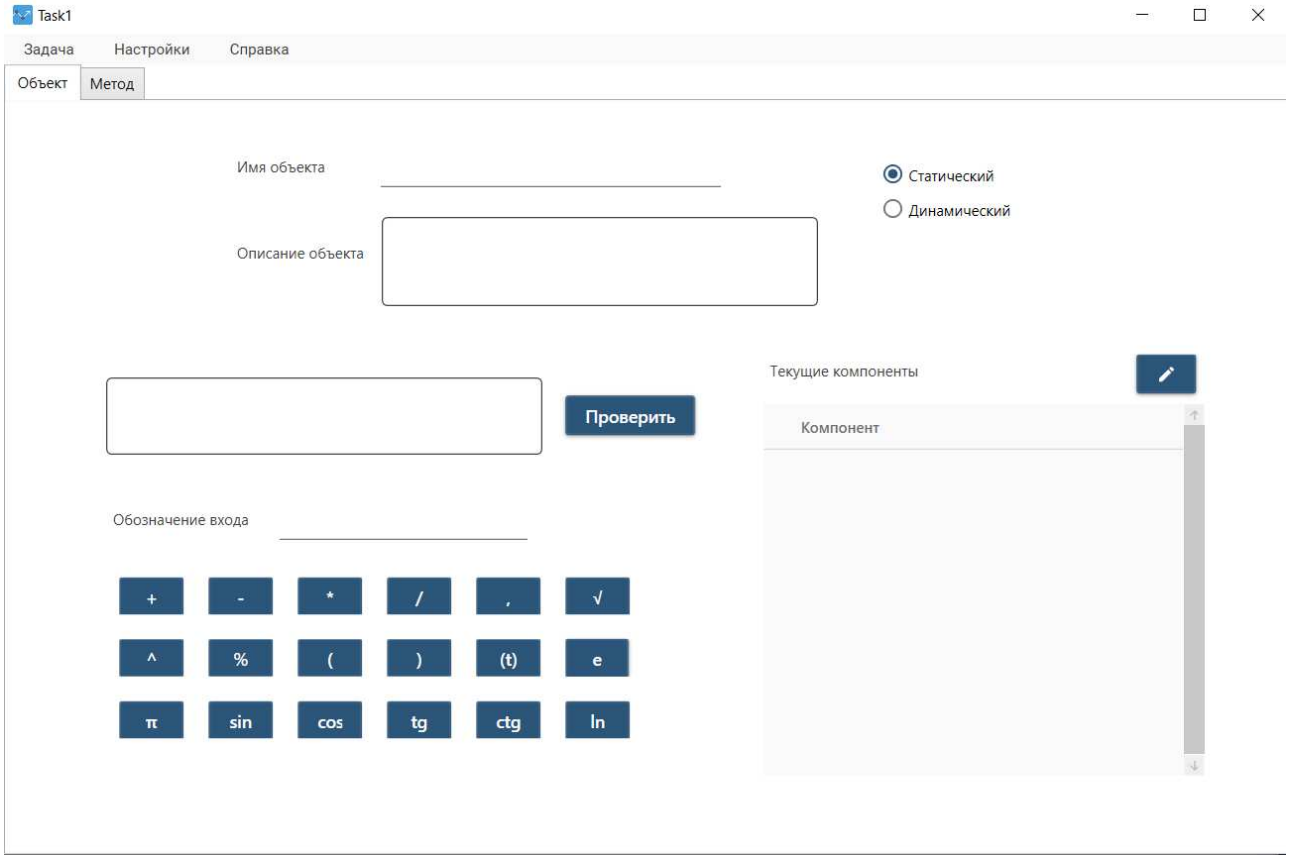

Рисунок 16 – Вкладка для ввода уравнения объекта

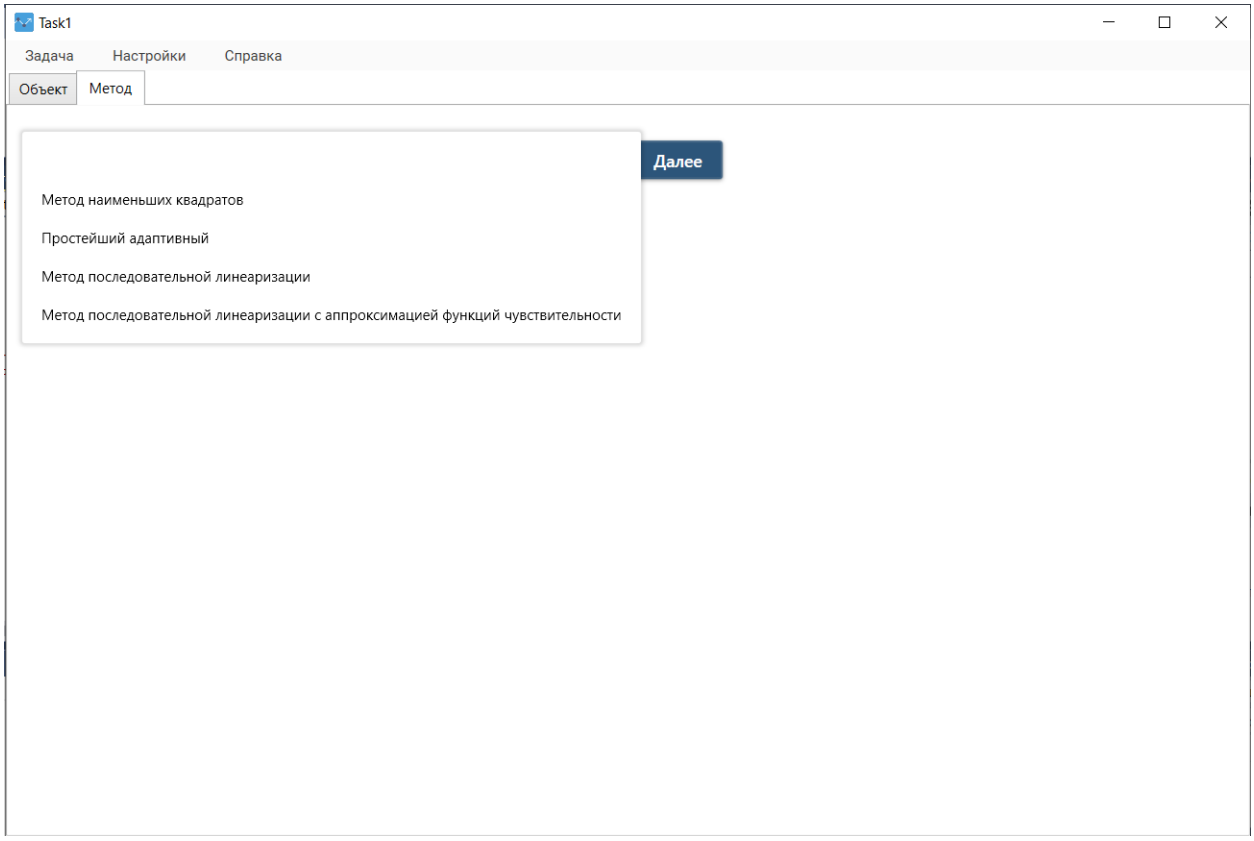

Рисунок 17 – Вкладка выбора метода идентификации

В первую очередь, необходимо ввести имя и описание объекта в соответствующие текстовые поля (рисунок г).

Далее нужно выбрать тип объекта – статический или динамический, так как это повлияет на ввод уравнения. При вводе уравнения статического объекта не нужно указывать моменты времени, например:

$$y = a + b \cdot \sin(s \cdot u) + c \cdot \cos(d \cdot u). \quad (3.1)$$

Однако при вводе уравнения динамического объекта пользователь обязан указать моменты времени у управляющих воздействий:

$$y(t) = a + b \cdot \sin\big(s \cdot u(t)\big) + c \cdot \cos(d \cdot u(t-1)). \quad (3.2)$$

По умолчанию будет выбран статический тип.

Также необходимо через запятую указать, какими буквами латинского алфавита будут обозначены управляющие воздействия. По умолчанию «u».

Далее необходимо ввести уравнение объекта в предложенное поле ввода (рисунок 18).

Рисунок 18 – Текстовое поле для ввода уравнения объекта

Также для ввода уравнения можно воспользоваться панелью, представленной на рисунке 19.

Рисунок 19 – Панель для ввода уравнения

Для проверки синтаксиса введенного выражения необходимо нажать кнопку «Проверить». В результате нажатия на экране появится одно из трех окон, изображенных на рисунках 20, 21, 22.

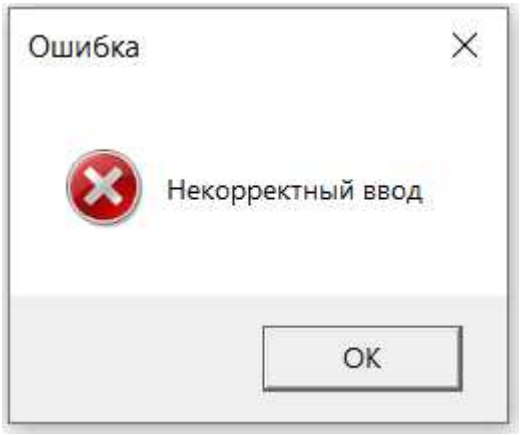

Рисунок 20 – Сообщения о некорректном вводе

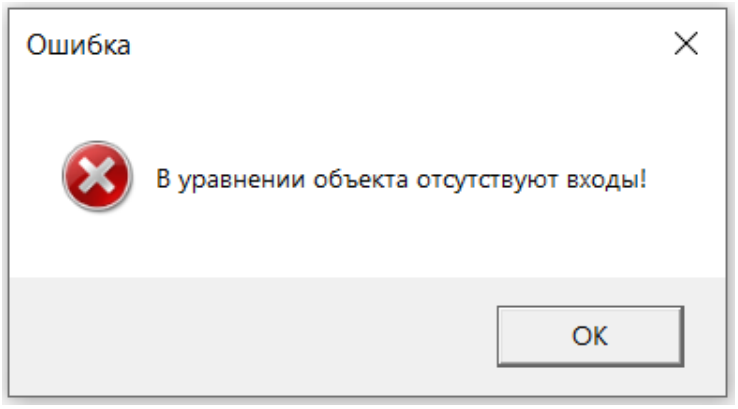

Рисунок 21 – Сообщения об отсутствии входов в уравнении

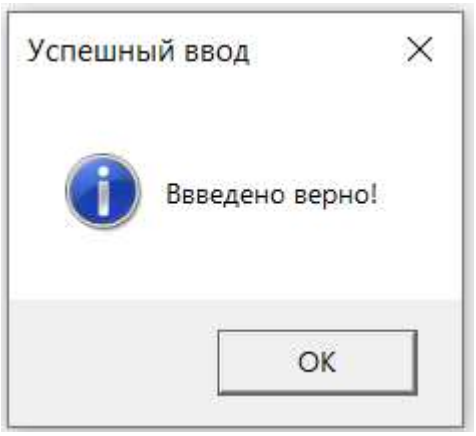

Рисунок 22 – Сообщения об успешном вводе

Если ошибок не было выявлено, будет заполнен список компонентов (рисунок 23).

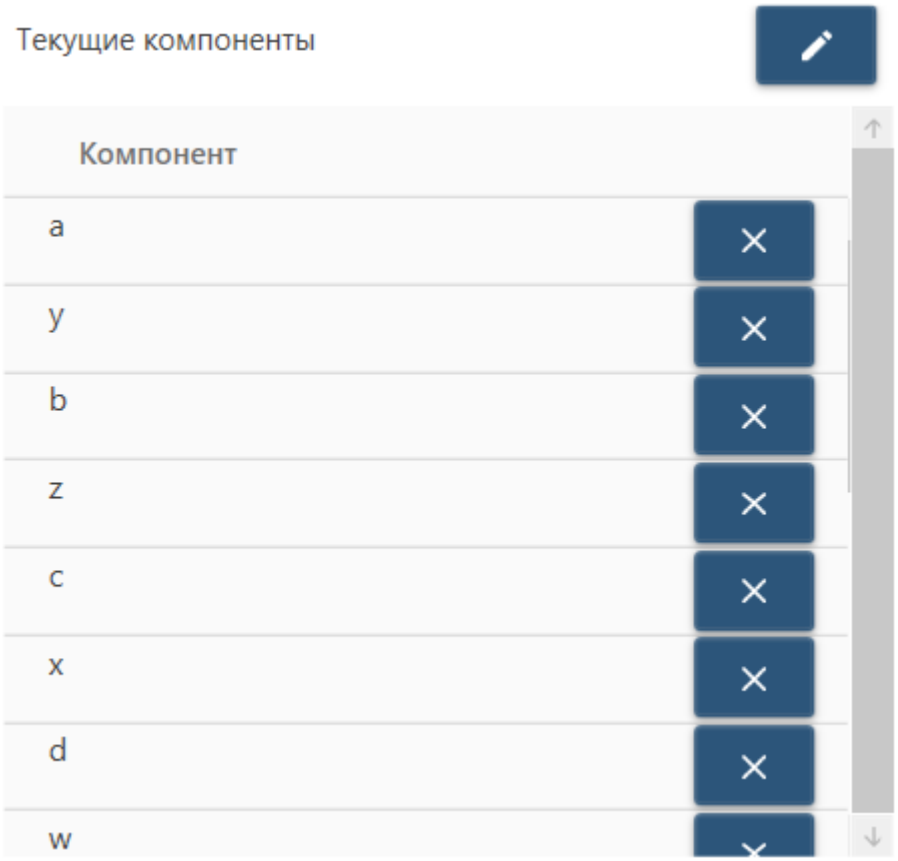

Рисунок 23 – Список компонентов

При нажатии на кнопку «Удалить» напротив нужного элемента в списке, выбранный компонент будет удален, а уравнение объекта изменено.

Для редактирования списка компонентов необходимо нажать на кнопку «Редактировать», расположенную над списком.

В появившемся окне (рисунок 24), пользователю предоставляется возможность создавать простые компоненты, совершать математические операции над двумя выбранным компонентами и добавлять составные компоненты в готовое уравнение.

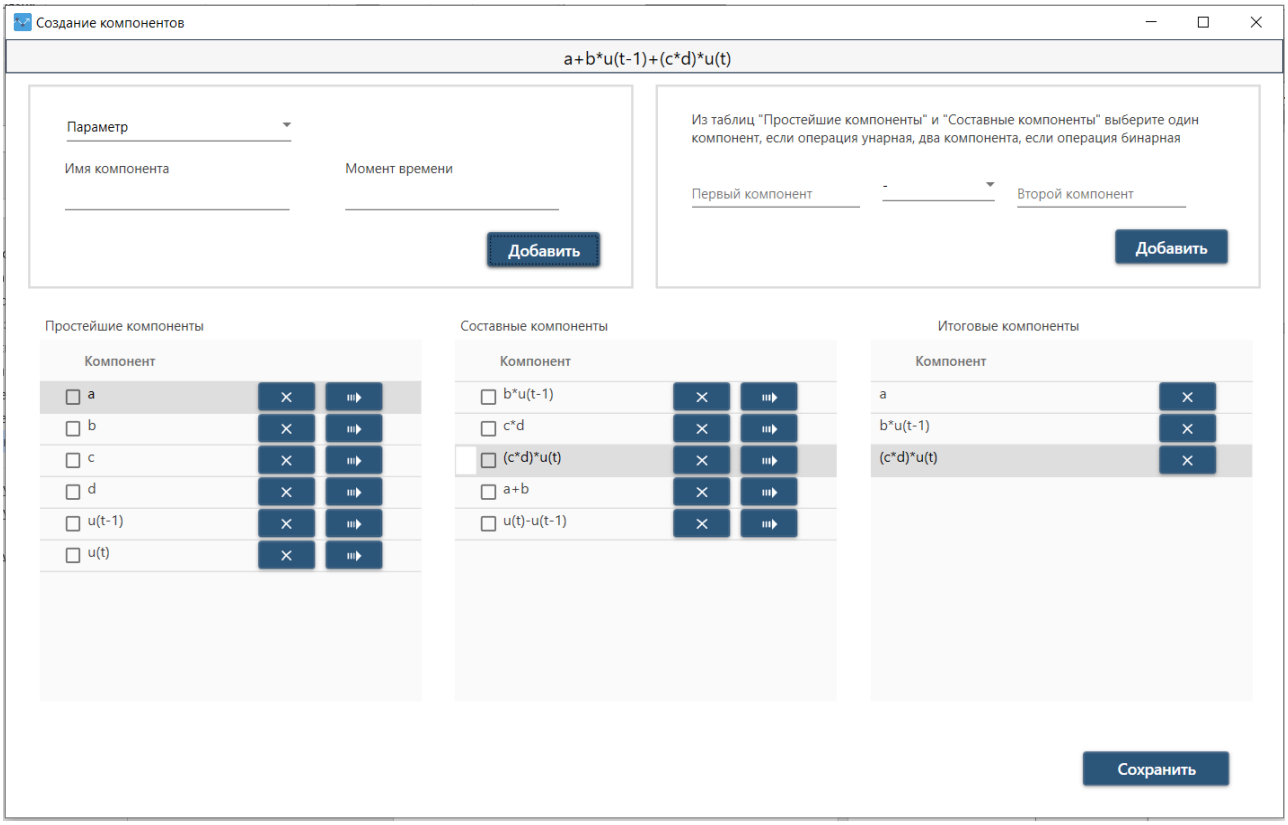

Рисунок 24 – Список компонентов

В панели расположенной в левом верхнем углу можно добавить простые компоненты:

– параметр;

– управляющее воздействие;

– выход в предыдущие моменты времени (если объект динамический).

Для этого необходимо выбрать тип компонента в выпадающем списке и ввести имя компонента (рисунок 25).

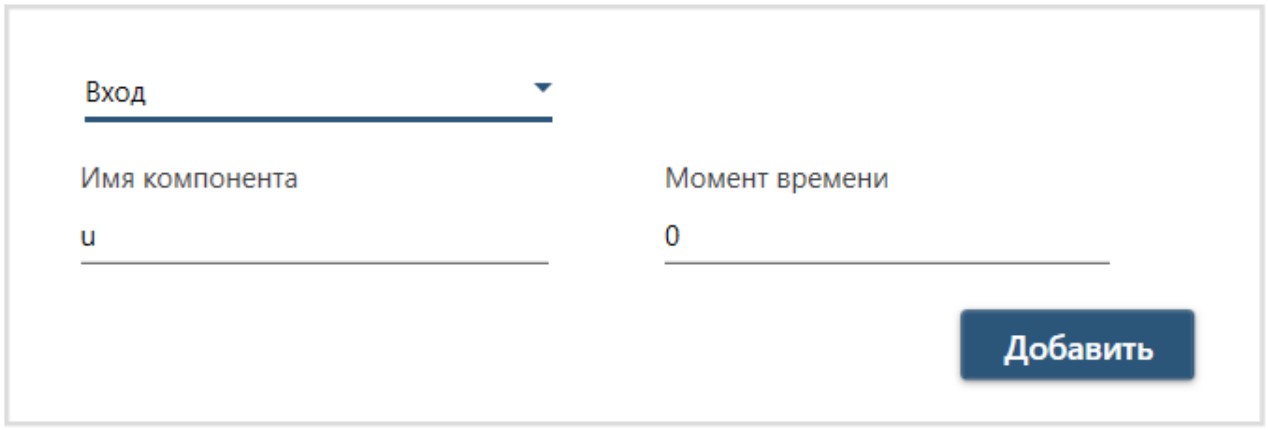

Рисунок 25 – Панель добавления простейших компонентов

Все созданные простейшие компоненты будут отображены в списке, показанном на рисунке 26.

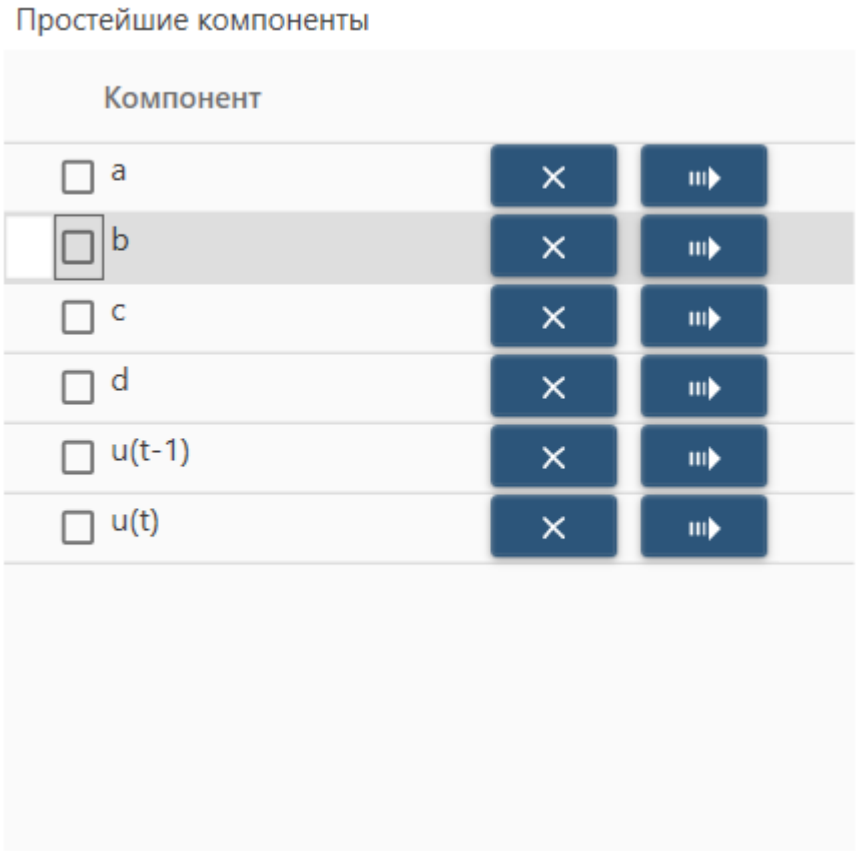

Рисунок 26 – Список простейших компонентов

Для добавления составных компонентов необходимо галочкой выбрать два компонента из списков с простейшими и составными компонентами, затем в

панели, расположенной в верхнем правом углу выбрать нужную операцию и нажать кнопку «Добавить» (рисунок 27). Все созданные составные компоненты будут отображены в списке, показанном на рисунке 28.

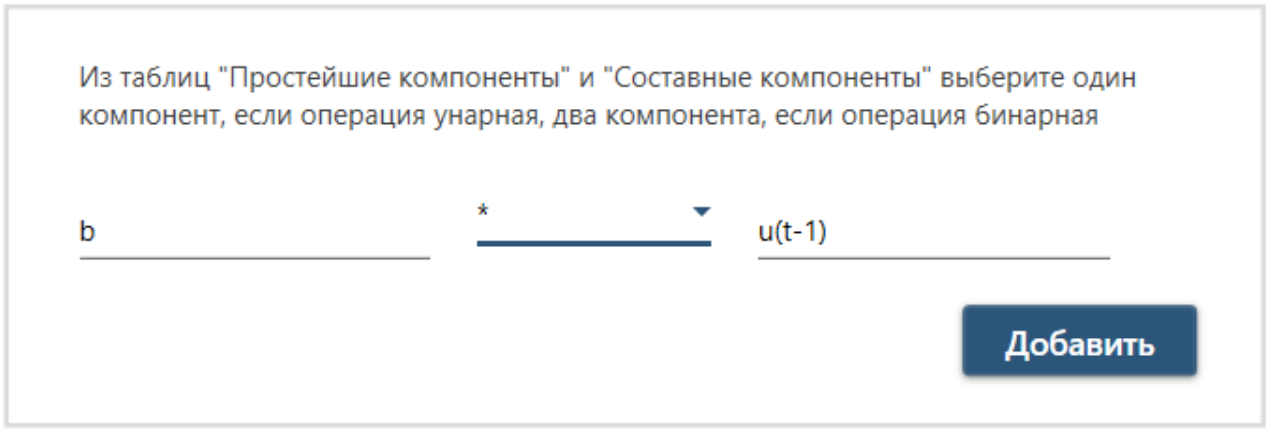

Рисунок 27 – Добавление составного компонента

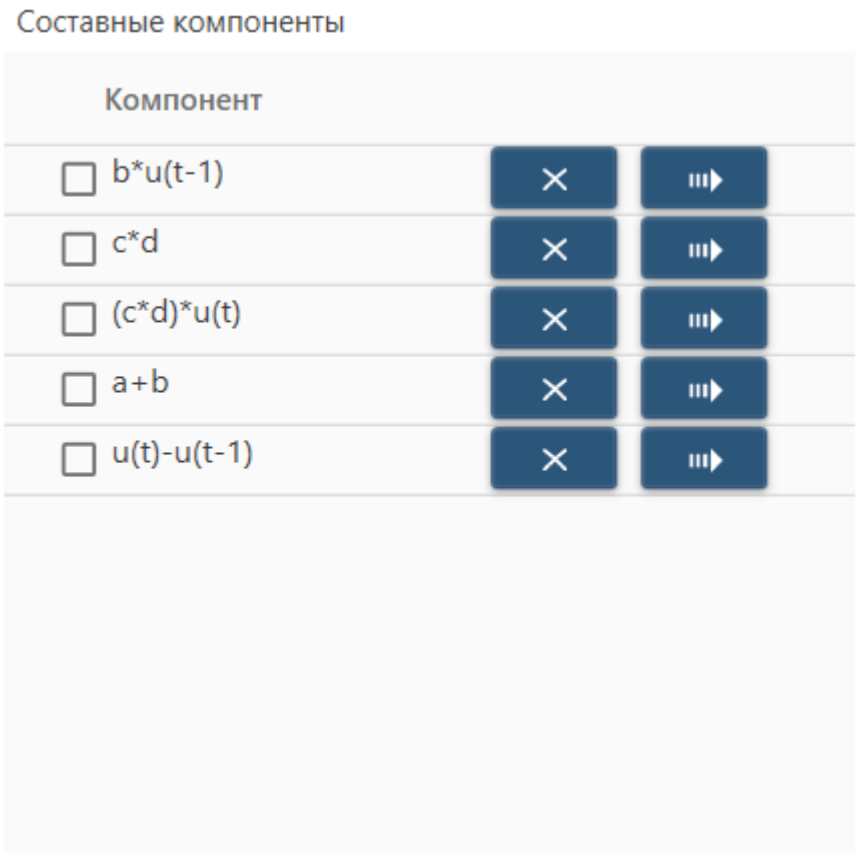

Рисунок 28 – Список составных компонентов

Для того, чтобы добавить компонент в список итоговых элементов, входящих в уравнение, которое отображено в верхней части окна, необходимо нажать на кнопку «Переместить» напротив нужного элемента в итоговом списке (рисунок 29). Уравнение объекта будет обновлено при добавлении новых элементов.

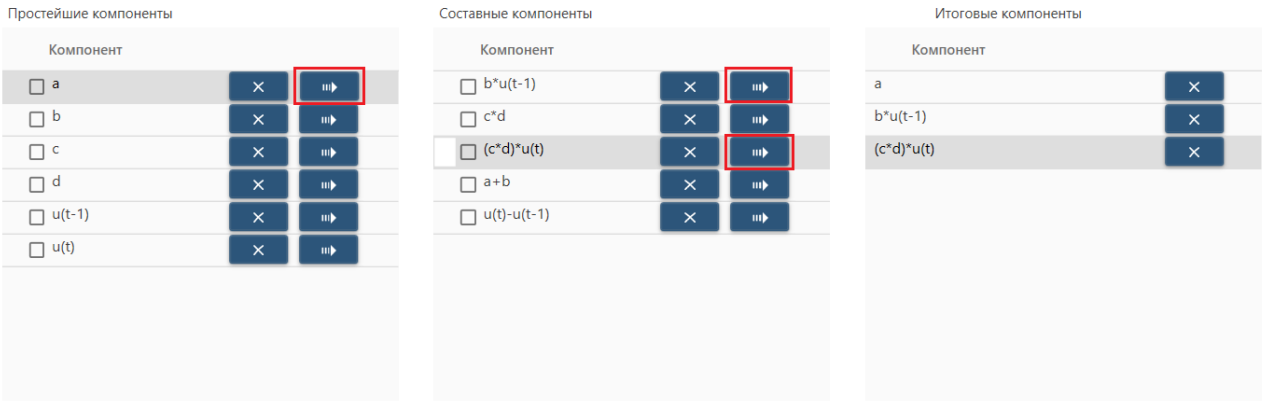

Рисунок 29 – Перемещение элементов в список итоговых

Также предоставляется возможность удалять элемента из трех выше описанных списков. При удалении элементов из списка с итоговыми компонентами, уравнение объекта будет обновлено.

Все изменения в уравнении объекта будут сохранены при нажатии на кнопку «Сохранить».

Для выбора метода идентификации необходимо выбрать нужный метода в выпадающем списке во вкладке, изображенной на рисунке 17. При изменении выбора, описание метода, расположенное ниже также будет меняться (рисунок 30).

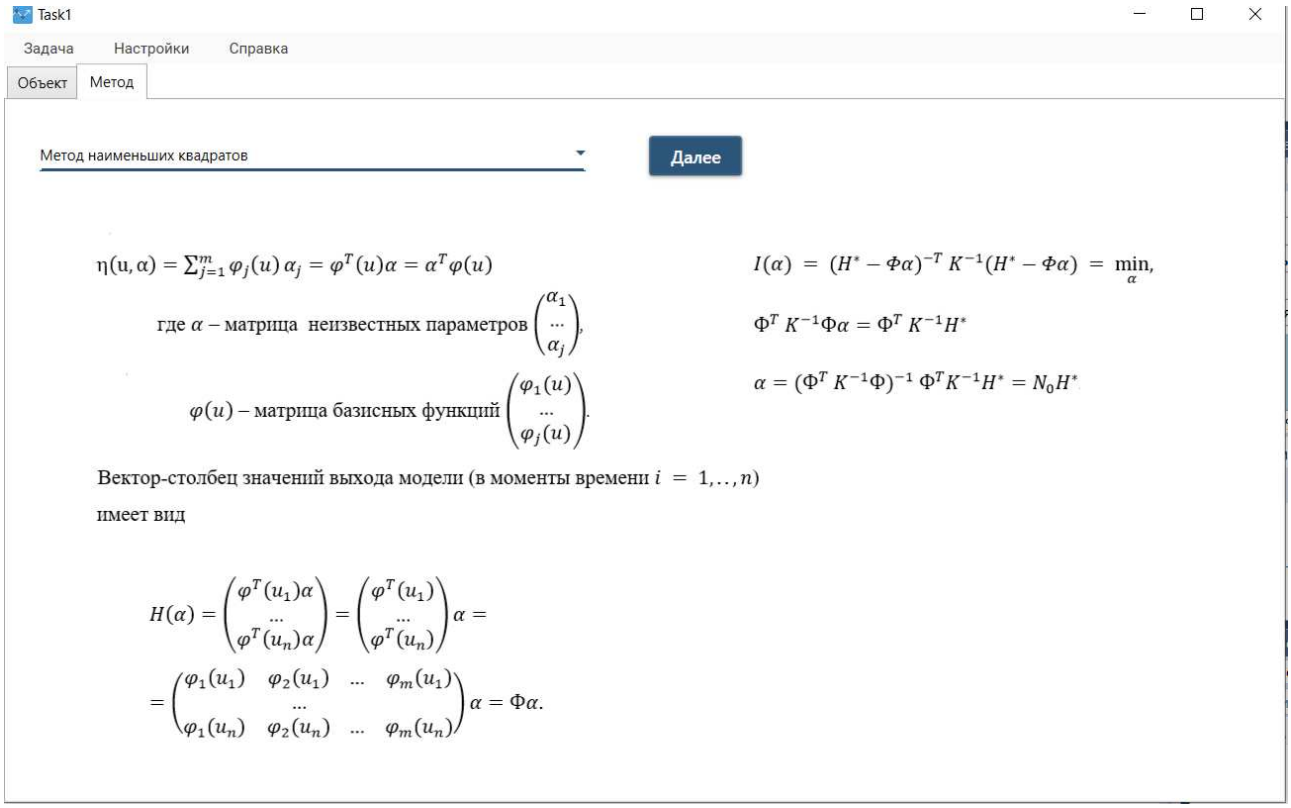

Рисунок 30 – Описание метода наименьших квадратов

Для открытия окна расчета параметров (рисунок 31) модели необходимо нажать на кнопку «Далее».

Кнопка, расположенная в верхнем правом углу появившегося окна, позволяет вернуться к первому окну.

Значения параметров объекта и начальных параметров модели можно ввести в панелях, представленных на рисунке 32. Галочкой у параметров объекта нужно отметить дифференцирование. Это означает, что данный параметр не является фиксированным и его необходимо найти. Если количество параметров слишком большое, каждый из параметров можно «свернуть», нажав на стрелочку рядом с именем параметра.

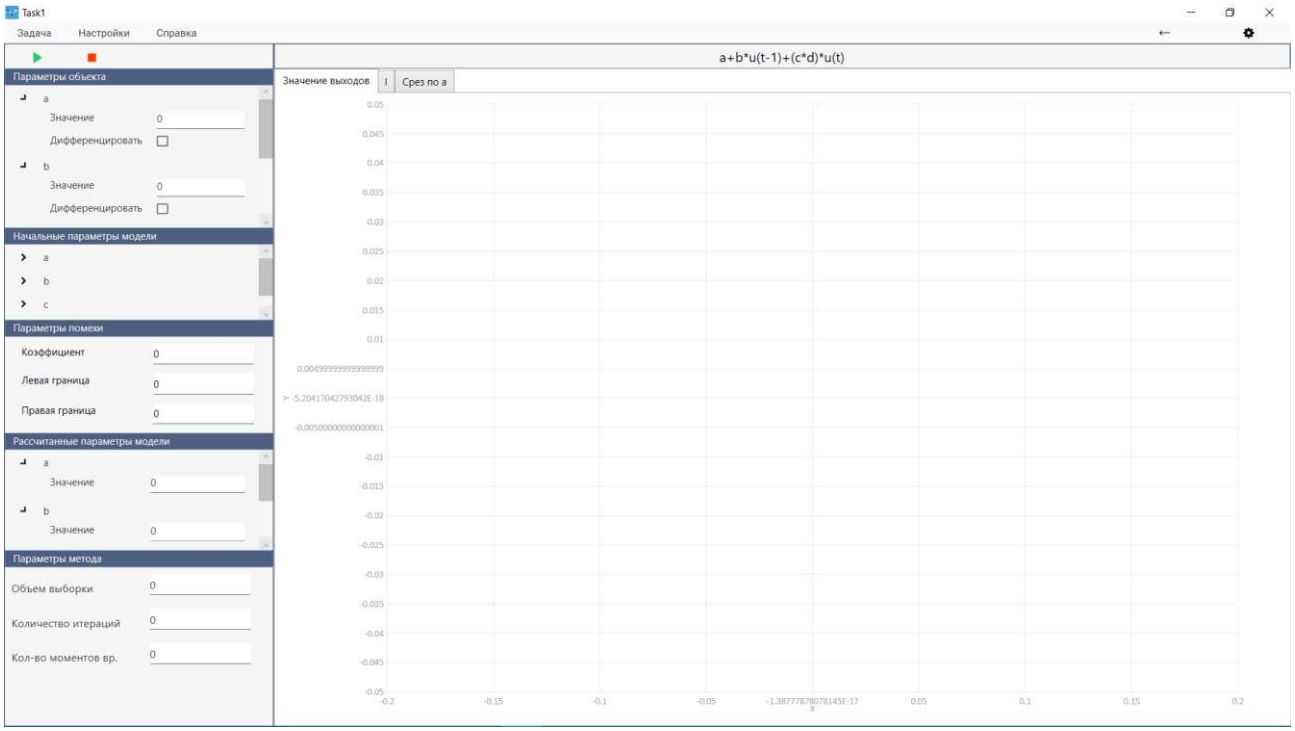

Рисунок 31 – Окно расчета параметров модели

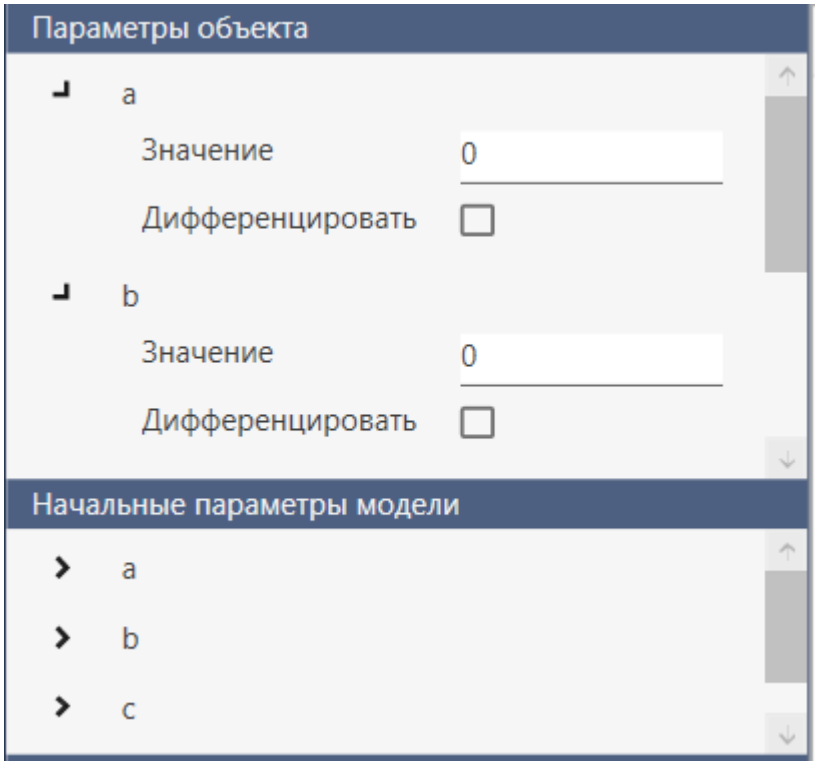

Рисунок 32 – Панели для ввода значений параметров объекта и начальных параметров модели

Для расчета параметров нужно нажать на кнопку «Рассчитать» и поля в панели (рисунок 33) будут заполнены рассчитанными параметрами модели.

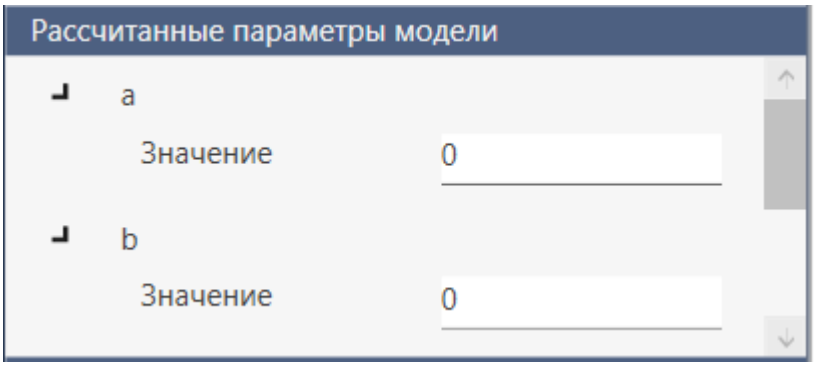

Рисунок 33 – Параметры модели

Параметры метода разные для каждого, их можно ввести в панели, представленной на рисунке 34.

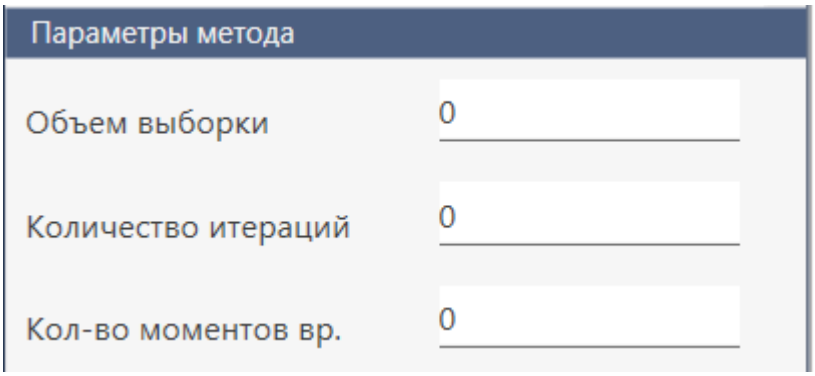

Рисунок 34 – Параметры метода

К выходу объекта также можно добавить помеху, распределенную по равномерному закону (рисунок 35).

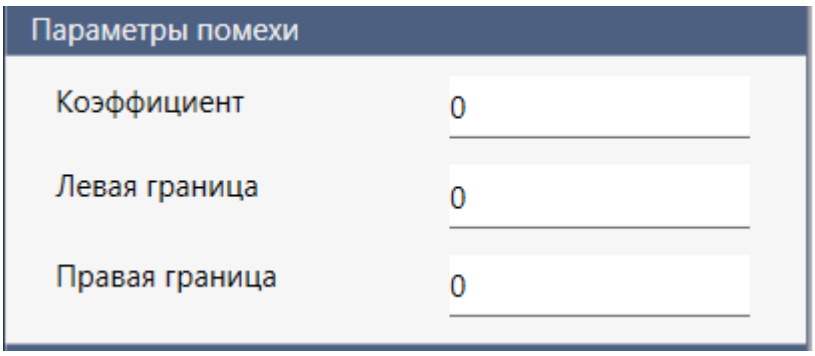

Рисунок 35 – Параметры помехи

При расчете параметров производится построение графика, представленного на рисунке 36.

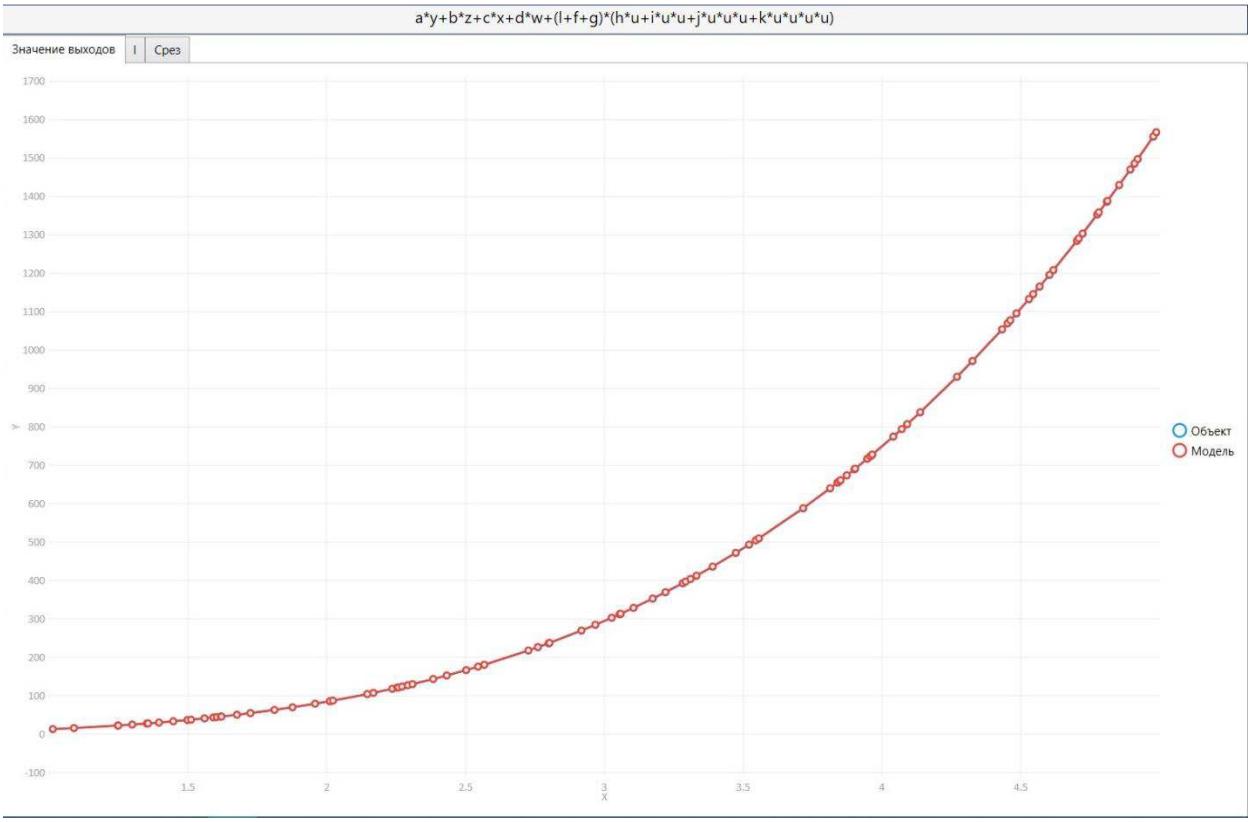

Рисунок 36 – График значений выходов модели и объекта в зависимости от входного воздействия

Данный график показывает значения выходов модели и объекта в каждый момент времени (если объект динамический) или за каждую итерацию (если объект статический).

Также пользователь может выбрать построение дополнительных графиков. Для этого необходимо нажать кнопку «Настройка отображаемых графиков» расположенную в верхней правой части окна.

В появившемся окне (рисунок 37) отметить галочкой необходимые графики. График «*I*» показывает значение критерия *I* в каждый момент времени (если объект динамический) или за каждую итерацию (если объект статический).

Для графика «Срезы» необходимо также выбрать переменные, по которым будут отображены срезы [20].

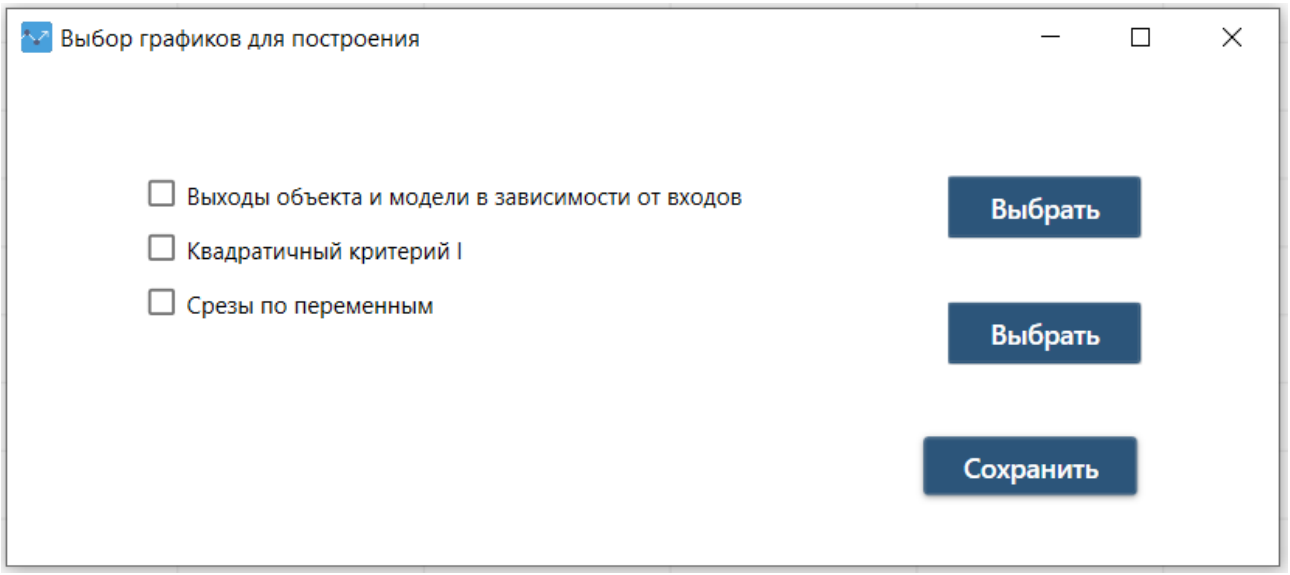

Рисунок 37 – Окно выбора графиков для построения

Во втором режиме программы уравнение объекта вводится также, как и в первом, однако вкладка «Метод» заменяется на «Методы» (рисунок 38).

Пользователь должен отметить галочкой два метода идентификации из списка, которые пользователь хотел бы сравнить и исследовать.

Если при нажатии на кнопку «Далее» будет выбран только один метод, будет выведено сообщение, представленное на рисунке 39.

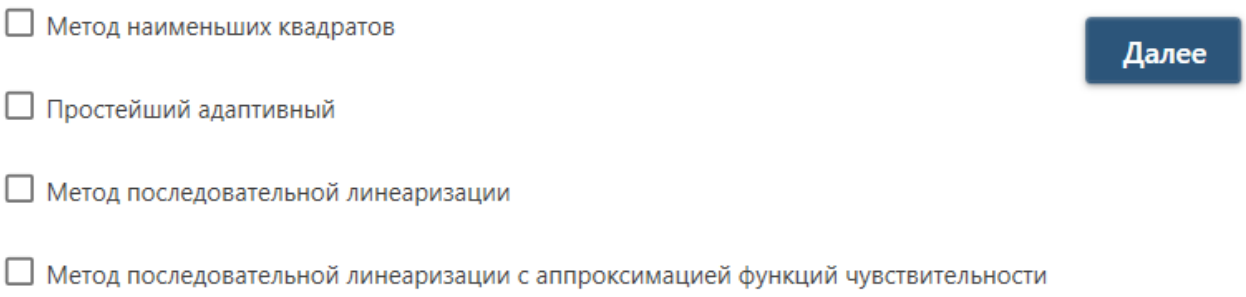

Рисунок 38 – Выбор методов идентификации

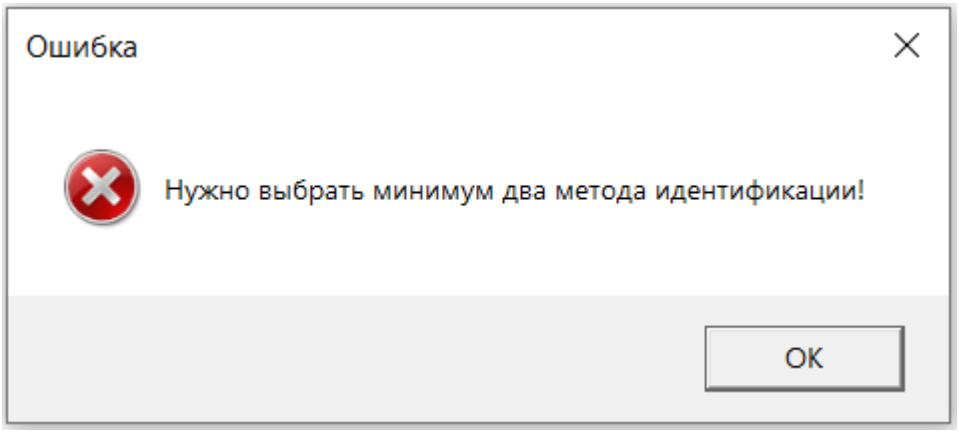

Рисунок 39 – Сообщение о неправильном выборе методов

Второе окно большой частью функционала совпадает с окном из первого режима, однако расчет параметров идет с помощью каждого выбранного ранее метода. Для ввода параметров необходимо выбрать нужный метод из выпадающего списка (рисунок 40).

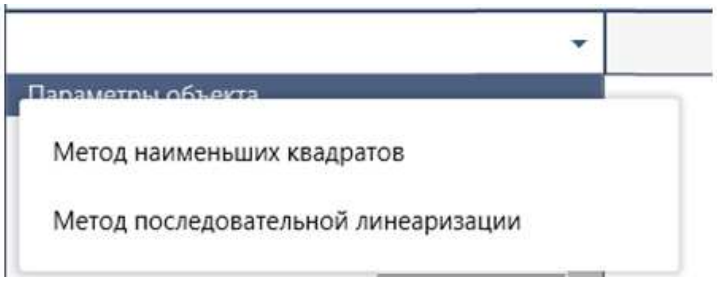

Рисунок 40 – Выбор метода идентификации для ввода параметров

4 Анализ результатов работы

Рассмотрим несколько случаев подстройки параметров дискретных моделей. В каждом из них модель имеет ту же структуру, что и объект, входным процессом является равномерно распределенная последовательность случайных чисел, имеющих нулевое среднее и лежащих в пределах от -1 до +1. Объем выборки $N = 100$.

Пусть уравнение модели записывается как:

$$y(t) = \alpha_0 + \alpha_1 \cdot u(t) + \alpha_2 \cdot u^2(t) \quad (4.1)$$

где a, b, c – искомые параметры модели;

u – управляющее воздействие.

Объем выборки для расчета оценок функций чувствительности n взят как $m, m + 10, m + 30$.

В таблицах 2–7 представлены результаты подстройки параметров модели с использованием аппроксимаций ФЧ представлен при различных n и $\delta\alpha^0$. Таблица 1 содержит результаты подстройки параметров модели с использованием точных функций чувствительности.

Таблица 1 – МПЛ

l	0	1	2	3	4	5	Истинные
α_0	1	1,205	1,789	1,992	1,999	2	2
α_1	1	0,529	0,096	0,0505	0,05	0,05	0,05
α_2	1	0,0447	-0,916	-1,383	-1,399	-1,4	-1,4
J	42,426	0,0029	3,729e-08	4,575e-15	6,078e-24	3,25e-27	

Таблица 2 – МПЛ с использованием оценок ФЧ, $\delta\alpha^0 = 0.1, n = m$

l	0	1	2	3	4	5	Истинные
α_0	1	1,99962	1,999	1,999	2	2	2

Окончание таблицы 2

α_1	1	0,05003	0,04999	0,04999	0,05	0,05	0,05
α_2	1	-1,39911	-1,3999	-1,3999	-1,3999	-1,4	-1,4
J	7,792e-06	1,104e-14	1,678e-25	3,52e-27	8,678e-28	4,946e-28	

Таблица 3 – МПЛ с использованием оценок ФЧ, $\delta\alpha^0 = 0.1$, $n = m + 10$

l	0	1	2	3	4	5	Истинные
α_0	1	1,9996	1,999	1,999	2	2	2
α_1	1	0,05003	0,04999	0,04999	0,04999	0,04999	0,05
α_2	1	-1,39918	-1,3999	-1,3999	-1,4	-1,4	-1,4
J	7,701e-06	9,687e-15	1,318e-25	3,423e-27	7,911e-28	5,99e-28	

Таблица 4 – МПЛ с использованием оценок ФЧ, $\delta\alpha^0 = 0.1$, $n = m + 30$

l	0	1	2	3	4	5	Истинные
α_0	1	1,9996	1,999	1,999	2	2	2
α_1	1	0,05006	0,04999	0,04999	0,05	0,05	0,05
α_2	1	-1,39906	-1,3999	-1,3999	-1,3999	-1,4	-1,4
J	9,365e-06	1,533e-14	2,64e-25	3,235e-27	1,129e-27	2,986e-28	

Таблица 5 – МПЛ с использованием оценок ФЧ, $\delta\alpha^0 = 1.1$, $n = m$

l	0	1	2	3	4	5	Истинные
α_0	1	1,99963	1,999	1,999	2	2	2
α_1	1	0,050013	0,05	0,05	0,05	0,05	0,05
α_2	1	-1,39911	-1,3999	-1,3999	-1,3999	-1,4	-1,4
J	7,548e-06	1,104e-15	3,316e-25	2,782e-27	6,66e-27	7,581e-27	

Таблица 6 – МПЛ с использованием оценок ФЧ, $\delta\alpha^0 = 1.1$, $n = m + 10$

l	0	1	2	3	4	5	Истинные
α_0	1	1,9996	1,999	1,999	2	2	2
α_1	1	0,05001	0,05	0,05	0,05	0,05	0,05
α_2	1	-1,3991	-1,3999	-1,3999	-1,4	-1,4	-1,4

Окончание таблицы 6

J	7,986e-06	4,499e-15	3,282e-25	3,423e-27	4,867e-28	3,858e-28	

Таблица 7 – МПЛ с использованием оценок ФЧ, $\delta\alpha^0 = 1.1$, $n = m + 30$

l	0	1	2	3	4	5	Истинные
α_0	1	1,9995	1,999	1,999	2	2	2
α_1	1	0,05006	0,04999	0,04999	0,05	0,05	0,05
α_2	1	-1,39906	-1,3999	-1,3999	-1,3999	-1,4	-1,4
J	7,712e-06	9,472e-14	1,122e-25	3,641e-27	5,738e-27	2,99e-28	

Рассмотрим случай $n = m + 50$ и $n = m$ при значениях параметров $\delta\alpha^0 = (1.1, \ldots., 1.1)$ и $\delta\alpha^0 = (10.1, \ldots., 10.1)$. Уравнение модели записывается как:

$$y(t) = \alpha_0 + {\alpha_1}^{u(t)} \tag{4.2}$$

где a, b – искомые параметры модели;

u – управляющее воздействие.

Исходя из значений критерия J в таблицах 8–10 можно сделать вывод, что для того, чтобы улучшить скорость схождения алгоритма необходимо при увеличении параметра $\delta\alpha^0$ увеличивать параметр n . Это связано с тем, что чем больше $\delta\alpha^0$, тем больше размеры гиперпрямоугольной области, по точкам которой мы оцениваем матрицу функций чувствительности на каждой итерации l.

Таблица 8 – МПЛ с использованием оценок ФЧ, $\delta\alpha^0 = 10.1$, $n = m + 10$

l	0	1	2	3	4	5	Истинные
α_0	1	5,6314	5,0251	5,00003	4,9999	5	5
α_1	1	3,4285	5,8003	5,9994	6	6	6
J	34,637	0,213	1,991e-06	3,26e-13	1,511e-19	9,037e-27	

Таблица 9 – МПЛ АС, $\delta\alpha^0 = 10.1$, $n = m + 50$

l	0	1	2	3	4	5	Истинные
α_0	1	5,527	5,0315	5,0001	5	5	5
α_1	1	3,246	5,6918	5,9979	6	6	6
J	25,025	0?258	9,812 e-06	7,931e-12	1,629e-17	3,624 e-27	

Таблица 10 – МПЛ с использованием оценок ФЧ, $\delta\alpha^0 = 0.1$, $n = m$

l	0	1	2	3	4	5	Истинные
α_0	1	5,7048	5,025	5,00002	5	5	5
α_1	1	3,3709	5,812	5,999	6	6	6
J	40,14	0,262	1,461e-06	2,689e-12	1,839e-19	4,506e-26	

Рассмотрим модель, описываемую уравнением:

$$y(t) = \alpha_0 + \alpha_1 \cdot \sin(w \cdot u(t)) + \alpha_2 \cdot \cos(q \cdot u(t)) \quad (4.3)$$

где a, b, c – искомые параметры модели;

w, q – фиксированные параметры модели;

u – управляющее воздействие.

Истинные значения параметров $a = -5$, $b = 0{,}001$, $c = 4.2$, $w = -1$, $q = 0.5$. В таблицах 11–12 представлены результаты подстройки параметров модели с использованием аппроксимаций ФЧ представлен при различных n и $\delta\alpha^0$. Таблица 11 содержит результаты подстройки параметров модели с использованием точных функций чувствительности. Исходя из результатов расчета, можно сделать вывод, что алгоритм МПЛ с использованием аппроксимаций ФЧ обеспечивает высокую скорость сходимости.

Таблица 11 – МПЛ

l	0	1	2	3	4	5	Истинные
α_0	1	2,6692	3,482	3,4871	3,0305	3	3

Окончание таблицы 11

α_1	1	0,0918	0,0066	0,0075	0,0099	0,001	0,001
α_2	1	3,6049	3,6911	3,9161	4,1681	4,19999	4.2
J	293,6	0,0376	0,0117	0,00017	2,295e-08	3,655e-14	

Таблица 12 – МПЛ с использованием оценок ФЧ, $\delta\alpha^0 = 0.1$, $n = m$+10

l	0	1	2	3	4	5	Истинные
α_0	1	3,0631	3,00084	3	3	3	3
α_1	1	0,00941	0,00998	0,0099	0,001	0,001	0,001
α_2	1	4,1338	4,19911	4,19999	4,19999	4.2	4.2
J	0,0006	1,055e-07	1,936e-13	3,56e-21	3,81e-27	2,146e-27	

Таблица 13 – МПЛ с использованием оценок ФЧ, $\delta\alpha^0 = 0.1$, $n = m + 30$

l	0	1	2	3	4	5	Истинные
α_0	1	3,058	3,0007	1,999	3	3	3
α_1	1	0,00991	0,009997	0,00999	0,001	0,001	0,001
α_2	1	4,139	4,1993	4,1999	4.2	4.2	4.2
J	0,0006	8,155e-08	1,24e-13	1,899e-21	3,837e-27	3,301e-27	

Таблица 14 – МПЛ с использованием оценок ФЧ, $\delta\alpha^0 = 0.1$, $n = m + 50$

l	0	1	2	3	4	5	Истинные
α_0	1	3,0748	3,0011	3	3	3	3
α_1	1	0,0098	0,009995	0,0099	0,0099	0,001	0,001
α_2	1	4,1219	4,1988	4,1999	4,199	4,199	4.2
J	0,0006	1,761e-07	4,607e-13	1,209e-20	3,842e-27	3,284e-27	

Таблица 15 – МПЛ с использованием оценок ФЧ, $\delta\alpha^0 = 3.1$, $n = m + 30$

l	0	1	2	3	4	5	Истинные
α_0	1	3,0575	3,0007	3	3	3	3
α_1	1	0,01008	0,00999	0,0099	0,001	0,001	0,001
α_2	1	4,199	4,199	4.2	4.2	4.2	4.2

Окончание таблицы 15

J	0,0006	7,939e-08	1,181e-13	1,622e-21	3,992e-27	3,741e-27	

Таблица 16 – МПЛ с использованием оценок ФЧ, $\delta\alpha^0 = 3.1$, $n = m + 70$

l	0	1	2	3	4	5	Истинные
α_0	1	3,058	3,0007	3	3	3	3
α_1	1	0,01024	0,01	0,01	0,01	0,01	0,01
α_2	1	4,138	4,1992	4.2	4.2	4.2	4.2
J	0,0006	8,255e-08	1,251e-13	1,9e-21	3,67e-27	7,205e-28	

Рассмотрим модель, имеющую несколько управляющих воздействий:

$$y(t) = \alpha_0 + \alpha_1 \cdot u_1(t) + \alpha_2 \cdot e^{u_2(t)} \quad (4.4)$$

где a, b, c – искомые параметры модели;

u_1, u_2 – управляющие воздействия.

Даже при небольшом значении параметра $\delta\alpha^0 = 0.1$ алгоритм показывает высокую скорость сходимости (таблицы 17–20).

Таблица 17 – МПЛ

l	0	1	2	3	4	5	Истинные
α_0	1	0,1529	0,2814	0,0413	0,338	0,3999	0,4
α_1	1	1,0725	1,3684	1,6675	2,039	2,0999	2,1
α_2	1	0,4228	0,1095	-0,1304	-0,448	-0,4999	-0,5
J	73,67	1,019	0,299	0,005	1,524e-06	1,079e-19	

Таблица 18 – МПЛ с использованием оценок ФЧ, $\delta\alpha^0 = 0.1$, $n = m$

l	0	1	2	3	4	5	Истинные
α_0	1	0,3929	0,3999	0,3999	0,3999	0,3999	0,4
α_1	1	2,093	2,0999	2,0999	2,0999	2,1	2,1
α_2	1	-0,4938	-0,4999	-0,4999	-0,4999	-0,5	-0,5

Окончание таблицы 18

J	0,00014	1,226e-10	1,079e-18	2,144e-27	1,888e-27	1,733e-27	

Таблица 19 – МПЛ с использованием оценок ФЧ, $\delta\alpha^0 = 0.1$, $n = m + 10$

l	0	1	2	3	4	5	Истинные
α_0	1	0,39595	0,3999	0,3999	0,4	0,4	0,4
α_1	1	2,0958	2,0999	2,0999	2,1	2,1	2,1
α_2	1	-0,4962	-0,4999	-0,4999	-0,4999	-0,5	-0,5
J	7,95e-05	2,254e-11	8,345e-20	1,592e-27	1,124e-27	5,186e-28	

Таблица 20 – МПЛ с использованием оценок ФЧ, $\delta\alpha^0 = 0.1$, $n = m + 30$

l	0	1	2	3	4	5	Истинные
α_0	1	0,3951	0,3999	0,3999	0,4	0,4	0,4
α_1	1	2,0951	2,099	2,099	2,1	2,1	2,1
α_2	1	-0,4955	-0,4999	-0,4999	-0,5	-0,5	-0,5
J	9,365e-06	1,533e-14	2,64e-25	3,235e-27	1,129e-27	2,986e-28	

Сравним метод последовательной линеаризации с использованием точных значений функций чувствительности и их аппроксимации на примере подстройки неизвестных параметров одномерной модели Гаммерштейна:

$$y(t) = \sum_{i=1}^{m_\alpha} \alpha_i y(t-i) + \sum_{i=0}^{m_\beta} \beta_i z(t-i), \beta_0 = 1, t \in \overline{0,N},$$
$$z(t) = \sum_{j=1}^{m_v} v_j \varphi_j\big(u(t)\big), \qquad (4.5)$$

где α_i, β_i, v_j – искомые параметры модели;

$y(t-i)$ – значение выхода модели в момент времени $t-i$;

$\{\varphi_j(\cdot)\}$ – система известных функций.

В данном случае:

$$\varphi_j(u) = u^j. \quad (4.6)$$

В таблицах 22–30 представлены результаты подстройки параметров модели с использованием аппроксимаций ФЧ представлен при различных n и $\delta\alpha^0$. Таблица 21 содержит результаты подстройки параметров модели с использованием точных функций чувствительности. Исходя из результатов расчета, можно сделать вывод, что алгоритм МПЛ с использованием аппроксимаций ФЧ обеспечивает высокую скорость сходимости вне зависимости от сложности структуры модели.

Таблица 21 – МПЛ

l	0	1	2	3	4	5	Истинные
α_0	1	-0.5505	-0.5123	-0.4983	-0.5000	-0.5000	-0,5
α_1	1	-0.3473	-0.3647	-0.3495	-0.3500	-0.3500	-0,35
α_2	1	0.6097	0.5857	0.6003	0.6000	0.6000	0,6
α_3	1	0.0572	0.0019	-0.0009	0.0010	0.0010	0,001
β_1	1	2,9260	2,9875	2,7991	2,7995	2,80000	2,8
β_2	1	0,68033	1,6957	1,4931	1,4996	1,5000	1,5
β_3	1	1,1364	-0,1519	-0,5458	-0,4996	-0,5000	-0,5
v_1	1	-2,0377	-0,9921	-1,0121	-0,9999	-1	-1
v_2	1	0,2452	-0,0303	0,07825	0,1003	0,09999	0,1
v_3	1	0,3216	0,24903	0,2136	0,1997	0,20000	0,2
v_4	1	1,6060	2,2127	2,487	2,5003	2,5000	2,5
J	8,5307	0,902	0,0001	7,50 e-07	5,698e-15	5,034e-25	

Таблица 22 – МПЛ с использованием оценок ФЧ, $\delta\alpha^0 = 0.1$, $n = m$

l	0	1	2	3	4	5	Истинные
α_0	1	-0.5591	-0.5288	-0.4938	-0.4996	-0.4999	-0,5
α_1	1	-0.3631	-0.3734	-0.3509	-0.3502	-0.3502	-0,35
α_2	1	0.5951	0.5765	0.5982	0.5997	0.5997	0,6
α_3	1	0.0516	0.0107	0.0073	0.0002	0.0006	0,001

Окончание таблицы 22

β_1	1	2,7896	2,832	2,7952	2,799	2,799	2,8
β_2	1	1,393	1,476	1,492	1,498	1,499	1,5
β_3	1	-0,667	-0,571	-0,515	-0,505	-0,501	-0,5
v_1	1	-1,0846	-1,0068	-1,0046	-1,0012	-1,0004	-1
v_2	1	0,2968	0,1074	0,1001	0,1002	0,1000	0,1
v_3	1	0,0781		0,2009	0,1992	0,1998	0,2
v_4	1	2,6640	2,5111	2,5074	2,5016	2,5006	2,5
J	8,5307	0,0004	3,677e-05	2,846e-06	1,892e-07	3,708e-09	

Таблица 23 – МПЛ с использованием оценок ФЧ, $\delta\alpha^0 = 0.1$, $n = m + 10$

l	0	1	2	3	4	5	Истинные
α_0	1	-0.5574	-0.5136	-0.4977	-0.5000	-0.5000	-0,5
α_1	1	-0.3531	-0.3638	-0.3495	-0.3500	-0.3500	-0,35
α_2	1	0.6050	0.5871	0.6003	0.6000	0.6000	0,6
α_3	1	0.0592	0.0040	0.0014	0.0010	0.0010	0,001
β_1	1	2,9438	2,7937	2,8001	2,7999	2,8000	2,8
β_2	1	1,5656	1,4955	1,5002	1,4999	1,500	1,5
β_3	1	-0,4724	-0,5054	-0,49992	-0,4999	-0,500	-0,5
v_1	1	1,1564	-1,0008	-1,0003	-0,999	-1,000	-1
v_2	1	0,1807	0,0977	0,1005	0,0999	0,1000	0,1
v_3	1	0,1462	0,1959	0,1999	0,1999	0,2000	0,2
v_4	1	2,3584	2,5028	2,4997	2,4999	2,5000	2,5
J	0,0238	2,678e-06	5,627e-08	1,743e-11	7,743e-14	1,347e-17	

Таблица 24 – МПЛ с использованием оценок ФЧ, $\delta\alpha^0 = 0.1$, $n = m + 30$

l	0	1	2	3	4	5	Истинные
α_0	1	1	-0.5671	-0.5136	-0.4978	-0.5000	-0,5
α_1	1	1	-0.4541	-0.3638	-0.3495	-0.3500	-0,35
α_2	1	1	0.7180	0.5871	0.6005	0.6000	0,6
α_3	1	1	0.0683	0.0052	0.0013	0.0010	0,001
β_1	1	2,6814	2,7625	2,7986	2,7999	2,7999	2,8

Окончание таблицы 24

β_2	1	1,3099	1,4665	1,5049	1,5003	1,5000	1,5
β_3	1	-1,1223	-0,4387	-0,4989	-0,4999	-0,4999	-0,5
v_1	1	-1,0473	-0,9847	-1,0003	-1	-1	-1
v_2	1	-0,1609	0,1245	0,1004	0,1	0,1	0,1
v_3	1	0,1727	0,1885	0,2005	0,2	0,2	0,2
v_4	1	2,5046	2,5054	2,5003	2,5	2,5	2,5
J	0,0543	0,0015	2,124e-07	1,921e-10	5,605e-13	1,058e-15	

Таблица 25 – МПЛ с использованием оценок ФЧ, $\delta\alpha^0 = 1.1$, $n = m$

l	0	1	2	3	4	5	Истинные
α_0	1	-0.5407	-0.5462	-0.5299	-0.5338	-0.5286	-0,5
α_1	1	-0.3423	-0.3759	-0.3791	-0.3645	-0.3601	-0,35
α_2	1	0.6177	0.5878	0.5804	0.5931	0.5961	0,6
α_3	1	0.0559	0.0279	0.0049	0.0245	0.0224	0,001
β_1	1	2,4310	2,8027	2,8334	2,8213	2,8249	2,8
β_2	1	1,2753	1,3969	1,5140	1,5125	1,5124	1,5
β_3	1	-0,4263	-0,4734	-0,4874	-0,4868	-0,4868	-0,5
v_1	1	-1,2464	-0,9995	-1,0065	-0,9896	-0,9899	-1
v_2	1	0,2669	0,1594	0,1168	0,1004	0,1044	0,1
v_3	1	0,2035	0,2106	0,1855	0,1854	0,1846	0,2
v_4	1	2,5513	2,4647	2,4647	2,4756	2,4732	2,5
J	0,0287	0,0032	0,0005	0,0003	0,0003	0,0003	

Таблица 26 – МПЛ с использованием оценок ФЧ, $\delta\alpha^0 = 1.1$, $n = m + 10$

l	0	1	2	3	4	5	Истинные
α_0	1	-0.5622	-0.5143	-0.4981	-0.4999	-0.4999	-0,5
α_1	1	-0.3620	-0.3529	-0.3501	-0.3500	-0.3500	-0,35
α_2	1	0.6008	0.5982	0.5999	0.5999	0.6000	0,6
α_3	1	0.0606	0.0125	-0.0007	0.0008	0.0009	0,001
β_1	1	2,7190	2,6573	2,7833	2,7990	2,8	2,8
β_2	1	1,4390	1,4893	1,5100	1,5030	1,4998	1,5

Окончание таблицы 26

β_3	1	-0,1316	-0,4026	-0,4969	-0,5	-0,5	-0,5
v_1	1	-0,8162	-0,9798	-1,0069	-1	-1	-1
v_2	1	-0,0278	0,1437	0,0997	0,0996	0,1	0,1
v_3	1	0,4727	0,3321	0,2320	0,2018	0,2	0,2
v_4	1	2,6048	2,5049	2,5049	2,5006	2,4999	2,5
J	0,2081	0,0221	0,0015	4,984e-06	8,688e-09	3,126e-08	

Таблица 27 – МПЛ с использованием оценок ФЧ, $\delta\alpha^0 = 1.1$, $n = m + 30$

l	0	1	2	3	4	5	Истинные
α_0	1	-0.5923	-0.6422	-0.5720	-0.5246	-0.5484	-0,5
α_1	1	-0.3386	-0.3396	-0.3932	-0.3872	-0.3854	-0,35
α_2	1	0.6340	0.6296	0.5725	0.5694	0.5743	0,6
α_3	1	0.1196	0.1625	0.0401	-0.0118	0.0166	0,001
β_1	1	2,2722	2,3651	2,4017	2,6488	2,6706	2,8
β_2	1	1,2837	1,3518	1,2555	1,3653	1,3908	1,5
β_3	1	-0,2760	-0,4611	-0,4269	-0,4269	-0,4286	-0,5
v_1	1	-1,4581	-1,0984	-1,1389	-1,0270	-1,0273	-1
v_2	1	-0,0729	0,2286	0,1088	0,1286	0,1063	0,1
v_3	1	0,1640	0,2266	0,2291	0,2388	0,2125	0,2
v_4	1	2,4591	2,8219	2,8233	2,5686	2,5762	2,5
J	0,5872	0,0294	0,0151	0,0073	0,0076	0,0076	

Таблица 28 – МПЛ с использованием оценок ФЧ, $\delta\alpha^0 = 3.1$, $n = m$

l	0	1	2	3	4	5	Истинные
α_0	1	-0.5035	-0.5072	-0.5000	-0.4999	-0.4999	-0,5
α_1	1	-0.3386	-0.3549	-0.3500	-0.3499	-0.3500	-0,35
α_2	1	0.6121	0.5969	0.6000	0.6000	0.6000	0,6
α_3	1	0.0186	0.0041	0.0010	0.0009	0.0009	0,001
β_1	1	3,3003	3,1719	2,7662	2,7967	2,8063	2,8
β_2	1	3,4290	2,6667	2,0089	1,0790	1,4954	1,5
β_3	1	-0,3149	-0,4278	-0,4922	-0,4998	-0,4996	-0,5

Окончание таблицы 28

v_1	1	-2,0792	-0,9778	-0,9635	-0,9956	-1,0004	-1
v_2	1	0,0607	0,1165	0,1201	0,1062	0,0990	0,1
v_3	1	0,1212	0,1739	0,1864	0,2008	0,1995	0,2
v_4	1	2,9350	2,2141	2,4599	2,4950	2,5007	2,5
J	0,2113	0,0064	0,0010	1,962e-06	1,661e-07	1,712e-07	

Таблица 29 – МПЛ с использованием оценок ФЧ, $\delta\alpha^0 = 3.1$, $n = m + 10$

l	0	1	2	3	4	5	Истинные
α_0	1	-0.6015	-0.5923	-0.5084	-0.5034	-0.5034	-0,5
α_1	1	-0.3900	-0.3561	-0.3488	-0.3496	-0.3498	-0,35
α_2	1	0.5639	0.6136	0.6037	0.6013	0.6011	0,6
α_3	1	0.0674	0.1020	0.0127	0.0055	0.0054	0,001
β_1	1	3,2770	2,2375	2,7408	2,7774	2,7813	2,8
β_2	1	0,9776	1,2941	1,4792	1,5237	1,5049	1,5
β_3	1	-0,7512	-0,5375	-0,5489	-0,5054	-0,4944	-0,5
v_1	1	-1,0896	-1,1254	-1,0013	-0,9821	-0,9969	-1
v_2	1	-0,0844	-0,1216	0,0704	0,0953	0,10608	0,1
v_3	1	-0,0614	0,3492	0,1752	0,1868	0,2013	0,2
v_4	1	3,0404	2,9034	2,5268	2,4861	2,4869	2,5
J	0,0746	0,0064	6,277e-05	7,793e-05	7,961e-06	1,345e-05	

Таблица 30 – МПЛ с использованием оценок ФЧ, $\delta\alpha^0 = 3.1$, $n = m + 30$

l	0	1	2	3	4	5	Истинные
α_0	1	-0.5406	-0.5035	-0.5072	-0.5000	-0.4999	-0,5
α_1	1	-0.2802	-0.3386	-0.3549	-0.3500	-0.3499	-0,35
α_2	1	0.6816	0.6121	0.5969	0.6000	0.6000	0,6
α_3	1	0.1168	0.0186	0.0041	0.0010	0.0009	0,001
β_1	1	3,5780	3,0084	2,8134	2,8017	2,8130	2,8
β_2	1	1,7317	1,3789	1,4148	1,4971	1,5011	1,5
β_3	1	-0,6243	-0,4752	-0,5017	-0,5016	-0,5011	-0,5
v_1	1	-0,2545	-0,7762	-0,9628	-0,9941	-0,9991	-1

Окончание таблицы 30

v_2	1	1,0229	0,2725	0,1186	0,0932	0,1006	0,1
v_3	1	-0,1030	0,0241	0,1687	0,2068	0,2	0,2
v_4	1	2,0415	2,2883	2,4793	2,5054	2,4990	2,5
J	0,3887	0,0068	0,0001	1,65e-05	4,289e-07	2,938e-08	

Рассмотрим случай, когда к выходу объекта приложена помеха.

Помеха является равномерно распределенным случайным числом, имеющее нулевое среднее и лежащее в пределах от -1 до +1.

Из таблиц 22–30 были выбраны параметры, при которых показаны наилучшие результаты ($\delta\alpha^0 = 1.1, n = m + 10$).

Результаты численного эксперимента приведены в таблице 31.

Таблица 31 – МПЛ с использованием оценок ФЧ, к выходу приложена помеха

l	0	1	2	3	4	5	Истинные
α_0	1	-1.5707	-0.5462	-0.5299	-0.5338	-0.5286	-0,5
α_1	1	-0.3423	-0.3759	-0.3791	-0.3645	-0.3821	-0,35
α_2	1	0.6277	0.5078	0.5804	0.5931	0.5971	0,6
α_3	1	0.1279	0.0559	0.0149	0.0245	0.0254	0,001
β_1	1	0,698	2,668	2,8247	2,841	2,8797	2,8
β_2	1	2,026	1,639	1,6338	1,6146	1,600	1,5
β_3	1	-1,166	-0,975	-0,649	--0,46	0,490	-0,5
v_1	1	-0,888	-0,9487	-0,958	-0,966	-0,99	-1
v_2	1	0,3204	0,1158	0,1002	0,082	0,0926	0,1
v_3	1	0,1995	0,1926	0,19495	0,2048	0,2203	0,2
v_4	1	2,56380	2,4237	2,44464	2,4327	2,4176	2,5
J	1,127	0,372	0,3442	0,3432	0,3421	0,3410	

ЗАКЛЮЧЕНИЕ

Результатом выполнения данной выпускной квалификационной работы стало разработанное программное обеспечение для решения задачи идентификации при построении систем управления динамическими объектами.

Были успешно решены следующие задачи:

– проанализирована предметная область с целью понимания решаемой проблемы. Изучены алгоритмы идентификации: метод наименьших квадратов, простейший адаптивный алгоритм, метод последовательной линеаризации, метод последовательной линеаризации с использованием аппроксимаций функций чувствительности;

– выбраны наиболее оптимальные средства разработки и языки программирования;

– спроектирована архитектура;

– разработан интерфейс для создания, редактирования и просмотра объектов, проведения расчетов параметров, сравнения алгоритмов;

– разработаны модули согласно программной архитектуре;

– проведено тестирование;

– проведено исследование метода последовательной линеаризации с использованием аналогов функций чувствительности.

Исходя из результатов численных исследований метода последовательной линеаризации с использованием аналогов функций чувствительности для одномерной модели Гаммерштейна, а также для различных простых объектов, было выявлено, что данный алгоритм обеспечивает почти такую же высокую скорость сходимости, как и метод последовательной линеаризации с использованием точных функций чувствительности.

В дальнейшем планируется расширение функционала программного обеспечения. В следующей версии появится возможность выбирать цвет шума, критерий минимума, также будут внедрены методы адаптивного управления.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Рубан, А. И. Методы анализа данных : учебное пособие / А. И. Рубан. – Красноярск: Краснояр. гос. техн. ун-т, 2004. – 319 с.

2. Дилигенская, А. Н. Идентификация объектов управления : учебное пособие / А. Н. Дилигенская. – Самара: Самар. гос. техн. ун-т., 2009. – 136 с.

3. Андриевская, Н. В. Проектирование и исследование идентификационных моделей управляющих систем реального времени : учебное пособие / Н. В. Андриевская. – Пермь : Изд-во Перм. нац. исслед. политехн. ун-та, 2013. – 202 с.

4. Гантмахер, Ф. Р. Теория матриц / Ф. Р. Гантмахер. – Москва: Наука, 1967. – 576 с.

5. Райбман, Н. С. Построение моделей процессов производства / Н. С. Райбман, В. М. Чадеев. – Москва.: Энергия, 1965. – 376 с.

6. Рубан, А. И. Идентификация и чувствительность сложных систем / А. И. Рубан. – Томск: Изд-во Томск. ун-та, 1982. – 302 с.

7. Рубан, А. И. Корректность линейного приближения в теории чувствительности для динамических моделей с запаздываниями / А. И. Рубан, К. Т. Уташев // Автоматика и телемеханика. – 1996. – №7. – С. 43–51.

8. Блюмин, С. Л. Псевдообращение : учебное пособие / С. Л. Блюмин, С. П. Миловидов. – Воронеж: ВорПИ-ЛипПИ, 1990. – 72с.

9. Рубан, А. И. Метод непараметрической поисковой оптимизации / А. И. Рубан // Изв. вузов. Физика. – 1995. – Т. 38, №9. – С. 65–73.

10. Рубан, А.И. Параметрическая оптимизация динамических систем методом последовательной линеаризации с использованием оценок функций чувствительности / А. И. Рубан // Информатика и системы управления: Межвуз. сб. науч. статей. Красноярск: КГТУ. – 1996. – С. 10–16.

11. Центр разработки MSDN [Электронный ресурс] : ресурсы для разработчиков. – Режим доступа: https://msdn.microsoft.com/ru-ru/

12. Microsoft Docs [Электронный ресурс] : Windows Presentation Foundation. – Режим доступа: https://docs.microsoft.com/ru-ru/dotnet/framework/wpf/

13. Prism Library [Электронный ресурс] : Prism Library Documentation. – Режим доступа: https://prismlibrary.github.io/docs/

14. Math.NET Numerics [Электронный ресурс] : Matrices and Vectors. – Режим доступа: https://numerics.mathdotnet.com/Matrix.html

15. mXparser [Электронный ресурс] : Tutorial. – Режим доступа: http://mathparser.org/mxparser-tutorial/

16. Material Design In XAML [Электронный ресурс] : Material Design In XAML Toolkit. – Режим доступа: http://materialdesigninxaml.net/

17. Visual Studio IDE [Электронный ресурс] : Visual Studio 2019. – Режим доступа: https://visualstudio.microsoft.com/ru/vs/

18. Microsoft Docs [Электронный ресурс] : Привязка данных и MVVM. – Режим доступа: https://docs.microsoft.com/ru-ru/windows/uwp/data-binding/data-binding-and-mvvm

19. Буч, Г. Язык UML. Руководство пользователя / Г. Буч, Д. Рамбо, И. Якобсон. – Изд. 2-е, перераб. и доп. – Москва : ДМК Пресс, 2006. – 496 с.

20. Рубан, А. И. Методы оптимизации : учеб. пособие / А. И. Рубан. – Изд. 2-е, исправ. и доп. – Красноярск : НИИ ИПУ, 2001. – 528 с.

21. Тарасенко, Ф. П. Непараметрическая статистика : монография / Ф. П. Тарасенко. – Томск : Изд-во Том. ун-та, 1976. – 292 с.

22. Льюнг, Л. Идентификация систем. Теория для пользователя : учебник / Л. Льюнг. – Москва : Наука, 1991. – 432 с.

23. Эйкхофф, П. Основы идентификации систем управления / П. Эйкхофф. – Москва : Мир, 1975. – 683 с.

24. Isermann, R. Identification of dynamic system / R. Isermann, M. Münchhof. – Berlin : Springer, 2011. – 732 p.

25. Цыпкин, Я. З. Основы информационной теории идентификации / Я. З. Цыпкин. – Москва : Наука, 1984. – 320 с.

26. Ельцов, А. А. Робастная идентификация распределенных динамических объектов, описываемых нагруженными уравнениями, с применением оценок функций чувствительности / А. А. Ельцов, А. И. Рубан // Автомат. и телемех.. – 1993. – № 2. – С. 140–148.

27. Рубан, А. И. Адаптивное управление с идентификацией : монография / А. И. Рубан. – Томск : Изд-во Том. ун-та, 1983. – 136 с.

28. Seber, G. A. F. Nonlinear Regression / G. A. F. Seber, C. J. Wild. – New York : John Wiley and Sons, 1989. – 360 p.

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

Институт Космических и Информационных технологий
Кафедра «Информатика»

УТВЕРЖДАЮ
Заведующий кафедрой
_____ А. С. Кузнецов
подпись инициалы, фамилия
« 05 » 07 2019 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.04 «Программная инженерия»

Разработка программного обеспечения для решения задачи идентификации при построении систем управления динамическими объектами

Руководитель _____ старший преподаватель А. С. Михалев
подпись, дата должность, ученая степень инициалы, фамилия

Выпускник _____ 05.07.2019 Н. М. Луговая
подпись, дата инициалы, фамилия

Консультант _____ профессор, д.т.н. А. И. Рубан
подпись, дата должность, ученая степень инициалы, фамилия

Нормоконтролер _____ 05.07.2019 О. А. Антамошкин
подпись, дата инициалы, фамилия

Красноярск 2019