

## Аннотация

**Биданец А. В. Нейросетевое управление в задаче преследования.**  
Выпускная квалификационная работа бакалавра по направлению 01.03.02 Прикладная математика и информатика. Таврическая академия Крымского федерального университета имени В. И. Вернадского.

Исследуется применимость модели многослойного перцептрона для управления преследователем в задаче о мягкой посадке. Разработано приложение, позволяющее моделировать процесс преследования, а также обучать и тестировать нейронные сети, и применять их в процессе преследования. Выполнен обзор имеющихся методов решения задачи о «мягкой посадке».

Ключевые слова: многослойный перцептрон, задача о «мягкой посадке», задача оптимального быстрогодействия, нейросетевое управление, метод экстремального прицеливания Н. Н. Красовского.

Страниц – 34, таблиц – 7, иллюстраций – 24, приложений – 3, библиографических источников – 10.



## Оглавление

Оглавление	2
Введение	3
Раздел 1. Постановка задачи о «мягкой посадке». Обзор существующих методов решения задачи	6
1.1. Постановка задачи о «мягкой посадке»	6
1.2. Обзор аналитических и численных методов решения задачи о «мягкой посадке»	7
Раздел 2. Описание аналитического решения. Нейросетевое управление в задаче о «мягкой посадке»	10
2.1. Аналитический метод решения задачи о «мягкой посадке»	10
2.2. Описание обучающих и тестовых выборок	13
2.3. Обучение нейросетевой модели	17
Раздел 3. Описание приложения и вычислительного эксперимента	18
3.1. Описание приложения	18
3.2. Описание эксперимента	19
Результаты эксперимента	22
Заключение	23
Список использованных источников	24
Приложение 1. Примеры выполненной мягкой посадки при помощи аналитического решения	24
Приложение 2. Результаты моделирования мягкой посадки при помощи нейросетевой модели	28
Приложение 3. Схема нейросетевой модели	34

## Введение

Актуальность задачи о «мягкой посадке» продиктована, прежде всего, важными практическими приложениями, такими задачами как стыковка космических аппаратов, посадка летательного аппарата на палубу корабля, посадка самолета в аэропорту при сильном боковом ветре предполагают проведение мягкой посадки. Часто построение управления в задаче о «мягкой посадке» требует решения задачи оптимального быстродействия [7] [8], применения принципа максимума Понтрягина [8] [6], а также правила экстремального прицеливания Н. Н. Красовского.

Решение подобных задач достаточно трудоёмко. В случае, когда параметры системы дифференциальных уравнений изменяются с течением времени (дифференциальные уравнения не являются линейными) вычислительная сложность алгоритмов, решающих поставленные задачи, возрастает многократно. Это служит препятствием для создания систем автоматического управления мягкой посадкой, т.к. время на выбор управления достаточно мало. В свою очередь, нейронные сети обладают обобщающей способностью. Научный интерес заключается в построении нейросетевой модели, способной решать с необходимой точностью задачу о «мягкой посадке» для различных значений параметров системы дифференциальных уравнений (в частности для различных значений коэффициента сопротивления). Это позволит снизить вычислительную сложность алгоритмов, решающих задачу о «мягкой посадке», а также других задач математической теории управления, т.к. в таком случае не придется пересчитывать всю траекторию при изменениях значений параметров системы дифференциальных уравнений.

В работе [7] установлены достаточные условия на параметры конфликтно-управляемого процесса при которых задача о «мягкой посадке» разрешима за конечное время. Задача о «мягкой посадке» решается в два этапа. На первом этапе происходит выравнивание вектора скорости преследователя по отношению к вектору скорости убегающего на основе правила экстремального прицеливания Н. Н. Красовского. После того как скорости объектов были выровнены переходят ко второму этапу преследования. На втором этапе решается задача оптимального быстродействия при условии, что управление является кусочно-постоянным. Решение можно найти аналитически, и оно представляет из себя кусочно-постоянную функцию с одним переключением.

В работе [8] рассмотрен способ решения задачи оптимального быстродействия при помощи модифицированного метода Нелдера-Мида и принципа максимума Понтрягина. Задача оптимального быстродействия сводится к краевой задаче.

Во всех перечисленных статьях динамика объектов описывается системой линейных дифференциальных уравнений.

Объектом исследования работы является задача преследования с участием одного преследователя и одного убегающего – задача о «мягкой посадке».

Предметом исследования является применимость нейросетевого подхода к решению задачи о «мягкой посадке».

Целью работы является исследование применимости нейросетевого подхода для решения задачи о «мягкой посадке» и разработка приложения, позволяющего для моделирования процесса преследования, а также создания и обучения нейронных сетей, их использования в качестве управляющего устройства в рассматриваемой задаче.

В работе были поставлены следующие задачи:

- Построить нейросетевую модель для управления преследователем в задаче о «мягкой посадке».
- Разработать приложение, моделирующее процесс преследования, а также позволяющее применять различные аналитические, численные и нейросетевые методы управления преследователем.

Разработать приложение, позволяющее создавать, обучать, тестировать нейронные сети, производить моделирование процесса преследования, создавать обучающие и тестовые выборки, содержащие примеры управления, осуществляющего мягкую посадку.

- Определить степень применимости нейросетевого подхода к задаче о мягкой посадке.

В первой главе дан обзор существующих аналитических, численных и нейросетевых методов решения задачи о «мягкой посадке».

Во второй главе дается постановка задачи о «мягкой посадке», подробно описан аналитический метод, используемый в работах [1], [7]. На основе выборки, построенной по данному методу была обучена нейросетевая модель. Дается описание обучающих и тестовых выборок. Описана схема нейросетевой модели, а также описан процесс ее обучения и тестирования.

В главе 3 описывается структура приложения, моделирующего процесс «мягкой посадки». Также дается описание вычислительного эксперимента, а также описаны результаты вычислительного эксперимента.

# Раздел 1. Постановка задачи о «мягкой посадке». Обзор существующих методов решения задачи

## 1.1. Постановка задачи о «мягкой посадке»

Динамика преследователя и убегающего задается дифференциальными уравнениями

$$\dot{x} = -\alpha x + \rho u, \quad (1)$$

$x = (x_1, x_2, x_3) \in R^3$  – координаты в пространстве,

$x(0) = x^0, \quad \dot{x}(0) = \dot{x}^0$  – начальные фазовые координаты,

$u \in R^3, \|u\| \leq 1$  – управление (преследователь),

$\alpha > 0$  коэффициент сопротивления среды,  $\rho > 0$  – параметр управления,

$$\dot{y} = -\beta y + \sigma v, \quad (2)$$

$y = (y_1, y_2, 0) \in R^3,$

$y(0) = y^0, \quad \dot{y}(0) = \dot{y}^0$  – начальные фазовые координаты,

$v \in R^3, \|v\| \leq 1$  – управление (убегающий),

$\beta > 0$  – коэффициент сопротивления среды,  $\sigma > 0$  – параметр управления.

Преследователь должен совершить точную мягкую посадку, то есть за конечное время добиться выполнения равенств  $x(T) = y(T), \dot{x}(T) = \dot{y}(T)$ .

Имеется фазовое ограничение  $x_3(t) \geq 0, y_3(t) \equiv 0$  при  $t \geq 0$ . Из этих условий следует, что  $v_3(t) \equiv 0$ .

## 1.2. Обзор аналитических и численных методов решения задачи о «мягкой посадке»

В работах [1] и [7] изучаются свойства множества достижимости. Также рассматривается вопрос управляемости при наличии фазового ограничения, аналитически описана область управления эквивалентной задачи управления.

В работе [7] задача о «мягкой посадке» решается в два этапа. На первом этапе происходит выравнивание вектора скорости преследователя по отношению к вектору скорости убегающего на основе правила экстремального прицеливания Н. Н. Красовского. После того как скорости объектов были выровнены переходят ко второму этапу преследования. На втором этапе решается задача оптимального быстрогодействия при нулевой начальной скорости и при условии, что управление является кусочно-постоянным. Длительность второго этапа можно определить непосредственно перед его началом. Решение представляет собой кусочно-постоянную функцию с одним переключением. На каждом этапе даётся явный вид оптимального управления. Был программно реализован процесс «мягкой посадки».

В работе [9] предложено решение задачи о «мягкой посадке», которое состоит из трёх стадий. На первой производится сближение по геометрическим координатам. На второй стадии осуществляется переход из точки в точку с попаданием на след убегающего в плоскости. На третьей стадии происходит преследование по следу в плоскости.

В работе [4] используется нейросетевой подход к решению задачи о «мягкой посадке». Для обучения нейронных сетей используются генетические алгоритмы, алгоритм обратного распространения ошибки.

Также в работах [4] и [8] рассмотрен способ решения задачи оптимального быстрогодействия. Используются модификация метода Нелдера-Мида и принцип максимума Л. С. Понтрягина. Затем задача оптимального быстрогодействия сводится к краевой задаче. Рассмотрим этот способ подробнее.

Для решения задачи оптимального быстрогодействия (4) согласно принципу максимума Л. С. Понтрягина, который состоит в следующем, сведём систему (4) из трёх уравнений второго порядка к системе из шести уравнений первого порядка относительно переменных  $\Delta_1, \Delta_2, \dots, \Delta_6$ .

Функция Гамильтона-Понтрягина для системы (4), (5) – (6) с учётом фазовых ограничений имеет вид:

$$H = \varphi_1 \Delta_2 + \varphi_2 (-\alpha \Delta_2 + \omega_x) + \varphi_3 \Delta_4 + \varphi_4 (-\alpha \Delta_4 + \omega_y) + \varphi_5 \Delta_6 + \varphi_6 (-\alpha \Delta_6 + \omega_z) + \varphi_7 \Delta_3 ,$$

где  $\varphi = (\varphi_1, \dots, \varphi_6)$  – вектор сопряженных переменных. Эта функция является линейной по независимым переменным  $\omega_x, \omega_y, \omega_z$  :

$$H(\omega_x, \omega_y, \omega_z) = C + \varphi_2 \omega_x + \varphi_4 \omega_y + \varphi_6 \omega_z ,$$

где  $C$  – величина, зависящая от оставшихся  $\varphi, \Delta, \alpha$ .

По принципу максимума Понтрягина оптимальное управление доставляет максимум функции  $H(\omega_x, \omega_y, \omega_z)$  и при ограничениях  $\|\omega\| \leq r$  имеет вид

$$\omega_x = \frac{r \varphi_2}{\sqrt{\varphi_2^2 + \varphi_4^2 + \varphi_6^2}}, \quad \omega_y = \frac{r \varphi_4}{\sqrt{\varphi_2^2 + \varphi_4^2 + \varphi_6^2}}, \quad \omega_z = \frac{r \varphi_6}{\sqrt{\varphi_2^2 + \varphi_4^2 + \varphi_6^2}} \quad (*)$$

Объединяя сопряженную систему  $\psi = -\partial H / \partial \Delta$ , систему (4), в которую подставим выражения для компонент управления (\*), краевые условия (5) – (6), получим краевую задачу принципа максимума, которая решается методом пристрелки.

Задав некоторые начальные условия для сопряженных координат  $\psi(0) = \psi_0$ , получим задачу Коши, которую можно проинтегрировать на промежутке от 0 до  $T$ , используя один из известных численных методов, и найти значения фазовых координат  $\Delta_1, \Delta_2, \dots, \Delta_6$  в конечный момент времени  $T$ .

Для решения задачи оптимального управления необходимо найти минимум функции

$$\Phi(\psi_1(0), \psi_2(0), \dots, \psi_6(0), T) \cong \sum_{i=1}^6 \Delta_i(T)^2, \quad (**)$$

то есть такой вектор  $\psi_1^*(0), \psi_2^*(0), \dots, \psi_6^*(0)$  и время  $T$ , при которых сумма в правой части выражения (\*\*) обращается в 0. Рассчитанное для таких



$\psi^*(0)$  в соответствии с выражениями (\*) управление  $w(t)$  будет использоваться затем в выражении

$$u(t) = \frac{(\alpha - \beta)\dot{y} + \sigma v + w}{\rho}$$

для вычисления управления преследователя дающее решение задачи о мягкой посадке.

Функцию  $\Phi(\psi_1(0), \psi_2(0), \dots, \psi_6(0), T)$  можно минимизировать одним из методов оптимизации. В работе [8] применялся модифицированный метод Нелдера-Мида. Минимизируемая функция  $\Phi(\psi_1(0), \psi_2(0), \dots, \psi_6(0), T)$  не всегда является выпуклой. Это свойство делает невозможным применение некоторых градиентных методов.

Метод решения задачи о «мягкой посадке», описанный в работах [1], [7] хотя и решает задачу о «мягкой посадке», не является оптимальным и на практике не всегда применим, т.к. не всегда возможно выровнять скорость преследователя по отношению к скорости убегающего из-за специфики технических систем, таких как, например, самолёт и авианосец.

С другой стороны, решение описанное в работах [4], [8] является более универсальным, но и соответственно более трудоёмким. В случае, когда параметры системы дифференциальных уравнений изменяются с течением времени, сложность вычислений, необходимых для решения рассматриваемой задачи, возрастает многократно, т.к. при изменении значений параметров системы приходится пересчитывать решение задачи оптимального быстродействия, которая сводится к краевой задаче.

Во всех перечисленных статьях динамика объектов описывается системой линейных дифференциальных уравнений.

Успешная реализация нейроконтроллера, способного обобщать управление для различных значений параметров системы в некотором диапазоне позволит сократить вычислительную сложность алгоритмов, решающих поставленную задачу.

## Раздел 2. Описание аналитического решения. Нейросетевое управление в задаче о «мягкой посадке»

### 2.1. Аналитический метод решения задачи о «мягкой посадке»

В работе [1] сформулированная выше игровая задача сводится к задаче оптимального быстрогодействия следующим образом.

**Условие 1.** Пусть  $\frac{\rho}{\sigma} > 1 + \left|1 - \frac{\alpha}{\beta}\right|$ .

Введём новую переменную

$$\Delta = x - y. \quad (3)$$

Вычитая уравнение (1) из уравнения (2) с учётом (3), получим:

$$\ddot{\Delta} = -\alpha\dot{\Delta} + (\beta - \alpha)\dot{y} + \rho u - \sigma v.$$

Будем считать, что начальные координаты, скорости убегающего и преследователя заданы, тогда заданы начальные значения  $\Delta, \dot{\Delta}$ .

В работе [1] рассматривается следующая задача оптимального быстрогодействия.

Объект описывается системой дифференциальных уравнений:

$$\begin{aligned} \ddot{\Delta} &= -\alpha\dot{\Delta} + w, \text{ где } \Delta, w \in R^3, \|w\| \leq r, \text{ где} \\ r &= \rho - \sigma \left(1 + \left|1 - \frac{\alpha}{\beta}\right|\right), \end{aligned} \quad (4)$$

при наличии фазовых ограничений  $\Delta_3 \geq 0$ . Начальное и конечное положение объекта задано

$$\Delta(0) = \Delta_0, \dot{\Delta}(0) = V_{\Delta_0}, \quad (5)$$

$$\Delta(T) = 0, \dot{\Delta}(T) = 0 \quad (6)$$

Необходимо найти управление  $w(t)$ , переводящее объект из заданного начального положения (5) в конечное положение (6) за минимально возможное время  $T$ .

Выполнение условий (6) будет означать для преследователя точную мягкую посадку.

Зная значение управления  $w(t)$  и значение скорости убегающего  $\dot{y}$  в каждый момент времени, нетрудно получить значение управления  $u(t)$  преследователя, обеспечивающее «мягкую посадку» при любом допустимом управлении убегающего (при выполнении условия 1):

$$u(t) = \frac{(\alpha - \beta)\dot{y} + \sigma v + w}{\rho} \quad (7)$$

На первом этапе производится выравнивание скоростей объектов, а на втором этапе непосредственно осуществляется мягкая посадка, путём решения задачи оптимального быстродействия.

На первом этапе преследования управление осуществляется нейронной сетью обученной аппроксимировать управление, описанное в разделе «игровое выравнивание скоростей объектов», в работе [1]

$$u(t, X, Y) = - \frac{e^{-\alpha(T_i - t)} X - e^{-\beta(T_i - t)} Y}{\|e^{-\alpha(T_i - t)} X - e^{-\beta(T_i - t)} Y\|}, \text{ где } t - \text{ текущий момент времени,}$$

$T_i = t_{i+1}$ ,  $X$  – вектор скорости преследователя в момент времени  $t$ ,  $Y$  – вектор скорости убегающего в момент времени  $t$ .

Упростим модель, считая, что  $\Delta t = t_{i+1} - t_i \forall i$ .

Теперь рассмотрим второй этап преследования. Будем решать задачу оптимального быстродействия (4), (5) – (6) [1].

В частном случае, когда оптимальное управление является релейным, удаётся получить аналитическое решение задачи оптимального быстродействия [1], учитывающее фазовое ограничение  $x_3 \geq 0$ .

Рассмотрим следующую теорему.

**Теорема.** Пусть  $\dot{\Delta}^0 = 0$  и  $\Delta^0 = \Delta$  – произвольно. Введём положительное число

$$\mu = \frac{\sqrt{\rho^2 \|\Delta\|^2 - s^2 |\Delta_3| - s \|\pi \Delta\|}}{\|\Delta\|^2}$$

Тогда время оптимального быстродействия в задаче (4), (5) – (6) равно

$$T = -\frac{2}{\alpha} \ln \left( e^{\frac{\alpha^2}{2\mu}} - \sqrt{e^{\frac{\alpha^2}{\mu}} - 1} \right), \quad (8)$$

а оптимальное управление является кусочно-постоянным с одним переключением:

$$w^*(t) = \{-\mu\Delta, t \in [0, \frac{1}{2}(T + \frac{\alpha}{\mu})], \mu\Delta, t \in [\frac{1}{2}(T + \frac{\alpha}{\mu}), T]\}.$$

## 2.2. Описание обучающих и тестовых выборок

Была обучена нейросетевая модель, состоящая из двух многослойных персептронов. Первая нейронная сеть была обучена на выборке, описывающей процесс выравнивания скоростей. Вторая нейронная сеть аппроксимирует управление  $w(t)$ , которое является решением задачи оптимального быстрогодействия (4), (5) – (6). В качестве метода обучения нейронных сетей был выбран метод обратного распространения ошибки.

Размеры обучающих и тестовых выборок приведены в таблицах ниже.

Таблица 2.2.1а.

Размеры обучающих и тестовых выборок

Этап	Train	test
Этап выравнивания скоростей	10000	10000
Этап решения задачи оптимального быстрогодействия	10000	10000

Таблица 2.2.1б.

Размеры обучающих и тестовых выборок (обобщенный случай)

Этап	Train	test
Этап выравнивания скоростей	7000	7000
Этап решения задачи оптимального быстрогодействия	7000	7000

Диапазоны значений входных параметров приведены в таблицах (2.2.2а), (2.2.2б). Процесс обучения осуществляется методом обратного распространения ошибки. Обучение продолжается до стабилизации функционала качества на обучающей выборке.

На вход первой нейронной сети (обеспечивающей выравнивание скоростей преследователя и убегающего) в каждый момент времени поступают значения следующих величин.

Таблица 2.2.2а.

Диапазоны значений входных параметров в обучающей и тестовой выборках на этапе выравнивания скоростей (обобщенный случай №1)

Входной параметр	Train		test	
	min	Max	min	max
1. Коэффициент сопротивления среды (убегающий)	0.01	0.5, 0.6, 0.7, 0.8	0.01	0.5, 0.6, 0.7, 0.8
2. Коэффициент сопротивления среды (преследователь)	0.01	0.5, 0.6, 0.7, 0.8	0.01	0.5, 0.6, 0.7, 0.8
3. Вектор текущей скорости преследователя $\dot{X} = (\dot{x}_1, \dot{x}_2, \dot{x}_3)$ (для каждой компоненты)	-2.5	2.5	-2.5	2.5
4. Вектор текущей скорости убегающего $\dot{Y} = (\dot{y}_1, \dot{y}_2)$ (для каждой компоненты)	-2.5	2.5	-2.5	2.5

Таблица 2.2.26.

Диапазоны значений входных параметров в обучающей и тестовой выборках на этапе выравнивания скоростей

Входной параметр	Train		test	
	min	Max	min	max
1. Вектор текущей скорости преследователя $\dot{X} = (\dot{x}_1, \dot{x}_2, \dot{x}_3)$ (для каждой компоненты)	-5.0	5.0	-5.0	5.0
2. Вектор текущей скорости убегающего $\dot{Y} = (\dot{y}_1, \dot{y}_2)$ (для каждой компоненты)	-5.0	5.0	-5.0	5.0

Первая нейронная сеть содержит 3 нейрона на выходном слое. Выходом нейронной сети является вектор  $u(t)$  – управление преследователя на первом этапе преследования (выравнивание скоростей).

На вход второй нейронной сети (обученной аппроксимировать управление  $w(t)$ , которое является решением задачи оптимального быстрогодействия (4), (5) – (6)) в каждый момент времени поступают значения следующих величин.

Таблица 2.2.3а.

Диапазоны значений входных параметров в обучающей и тестовой выборках на этапе решения задачи оптимального быстродействия (при фиксированных параметрах системы)

Входной параметр	Train		test	
	Min	Max	Min	max
Текущий момент времени $t$ от начала второго этапа преследования	0.0	T	0.0	T
Разность фазовых координат к началу второго этапа преследования				
$\Delta_1$	-0.9	0.9	-0.9	0.9
$\Delta_2$	-0.9	0.9	-0.9	0.9
$\Delta_3$	0.0	0.9	0.0	0.9

Таблица 2.2.3б.

Диапазоны значений входных параметров в обучающей и тестовой выборках на этапе решения задачи оптимального быстродействия (обобщенный случай №1)

Входной параметр	Train		test	
	Min	Max	Min	max
1. Коэффициент сопротивления среды (убегающий)	0.01	0.5, 0.6, 0.7, 0.8	0.01	0.5, 0.6, 0.7, 0.8
2. Коэффициент управления (убегающий)	0.01	0.5, 0.6, 0.7, 0.8	0.01	0.5, 0.6, 0.7, 0.8
3. Коэффициент сопротивления среды (преследователь)	0.01	0.5, 0.6, 0.7, 0.8	0.01	0.5, 0.6, 0.7, 0.8
4. Коэффициент управления (преследователь)	0.01	0.5, 0.6, 0.7, 0.8	0.01	0.5, 0.6, 0.7, 0.8
5. Текущий момент времени $t$ от начала второго этапа преследования	0.0	T	0.0	T
Разность фазовых координат к началу второго этапа преследования				
$\Delta_1$	-0.9	0.9	-0.9	0.9
$\Delta_2$	-0.9	0.9	-0.9	0.9
$\Delta_3$	0.0	0.9	0.0	0.9

$T$  – время, необходимое для осуществления мягкой посадки при начальных одинаковых скоростях объектов (длительность второго этапа преследования), которое вычисляется по формуле (8).

Вторая нейронная сеть содержит 3 нейрона на выходном слое. Выходом нейронной сети является вектор  $w(t)$  – решение задачи оптимального быстрогодействия.



### 2.3. Обучение нейросетевой модели

При обучении нейронных сетей объекты из выборки извлекаются случайным образом. Обученная модель функционирует следующим образом. На первом этапе нейронная сеть №1 должна выровнять скорость преследователя по отношению к скорости убегающего. По окончании первого этапа управление передается нейронной сети №2. На втором этапе управление преследователя строится по формуле (7). При этом нейронная сеть №2 задаёт управление  $w(t)$ , которое затем подставляется в управление (7).

Перебор различных конфигураций нейронных сетей осуществляется автоматически. Количество скрытых слоёв варьировалось от 1 до 3. Количество нейронов в каждом слое варьировалось в диапазоне от 3 до 33.

В качестве функции активации нейронов многослойного персептрона использовались: арктангенс, гиперболический тангенс и логарифмическая функция  $\ln(x + \sqrt{x^2 + 1})$ .

Схема нейросетевой модели приведена в приложениях (см. схему 1).

В связи с тем, что нейронная сеть может имеет погрешность управления, на выходе каждой из двух нейронных сетей производится нормализация вектора управления.

На втором этапе «мягкой посадки» управление преследователя описывается выражением (7). В нейросетевой модели первые два слагаемых в числителе выражения (7) гарантировано включены в управление. Решение задачи оптимального быстрогодействия  $w(t)$ , аппроксимированное многослойным персептроном, имеет погрешность. Поэтому множитель  $\eta$ , с которым входит управление  $w(t)$ , определим так:

$$\eta = \{1, \|\frac{(\alpha-\beta)\dot{y}+\sigma v+w}{\rho}\| \leq 1, \|\frac{(\alpha-\beta)\dot{y}+\sigma v+\eta w}{\rho}\|, \|\frac{(\alpha-\beta)\dot{y}+\sigma v+w}{\rho}\| > 1,$$

$$\text{где } u(t) = \frac{(\alpha-\beta)\dot{y}+\sigma v+\eta w}{\rho}.$$

Программно эта формула реализуется за счет бинарного поиска (см. листинг. п. 4, п. 5). Таким образом,  $\|u(t)\| \leq 1$ .

## **Раздел 3. Описание приложения и вычислительного эксперимента**

### **3.1. Описание приложения**

Для разработки приложения был выбран язык программирования C#. Он обеспечивает высокую производительность приложений. Это требование является одним из важнейших для разработки приложений, моделирующих динамические процессы, как, например, процесс преследования и мягкой посадки, описываемые дифференциальными уравнениями.

Для визуализации процесса преследования используется библиотека Tao Framework – это библиотека, предоставляющая разработчикам .NET и Mono доступ к возможностям популярных библиотек вроде OpenGL и SDL [10]. Использование этой библиотеки значительно сокращает время на разработку графических модулей приложений.

Приложение позволяет генерировать обучающие и тестовые выборки на основе описанного ранее аналитического решения задачи о «мягкой посадке». Эти выборки могут быть использованы для обучения нейронных сетей и проверки их обобщающей способности.

При запуске приложения открывается окно выбора режима управления убегающим. На данный момент в приложении реализован единственный режим управления убегающим – управление с клавиатуры.

Приложение позволяет генерировать выборки, описывающие различные стратегии преследования. Также оно позволяет обучать, тестировать и применять в процессе моделирования нейронные сети. Обучение нейронных сетей может производиться либо в ручном, либо в автоматическом режиме.

В главном окне приложения моделируется процесс преследования, отображаются все основные показатели процесса, такие как: координаты объектов, их векторы скорости, расстояние между объектами, разность координат, разность скоростей по каждой из координат, управление преследователя и др.

Другие элементы интерфейса приложения представлены на рис. п. 4.1. – п. 4.5.

### 3.2. Описание эксперимента

Эталонной моделью для сравнения с нейросетевым подходом является аналитическое решение, описанное в разделе 2.1.

Выполним в автоматическом режиме обучение двух нейронных сетей, соответствующих ранее рассмотренной схеме (в разделе 2.3.). Выберем из них две нейронные сети с наилучшим эмпирическим риском, соответствующие двум этапам процесса «мягкой посадки»: выравнивание скоростей и решение задачи оптимального быстродействия.

Параметры обучения нейронных сетей представлены на рис. 1, 2.

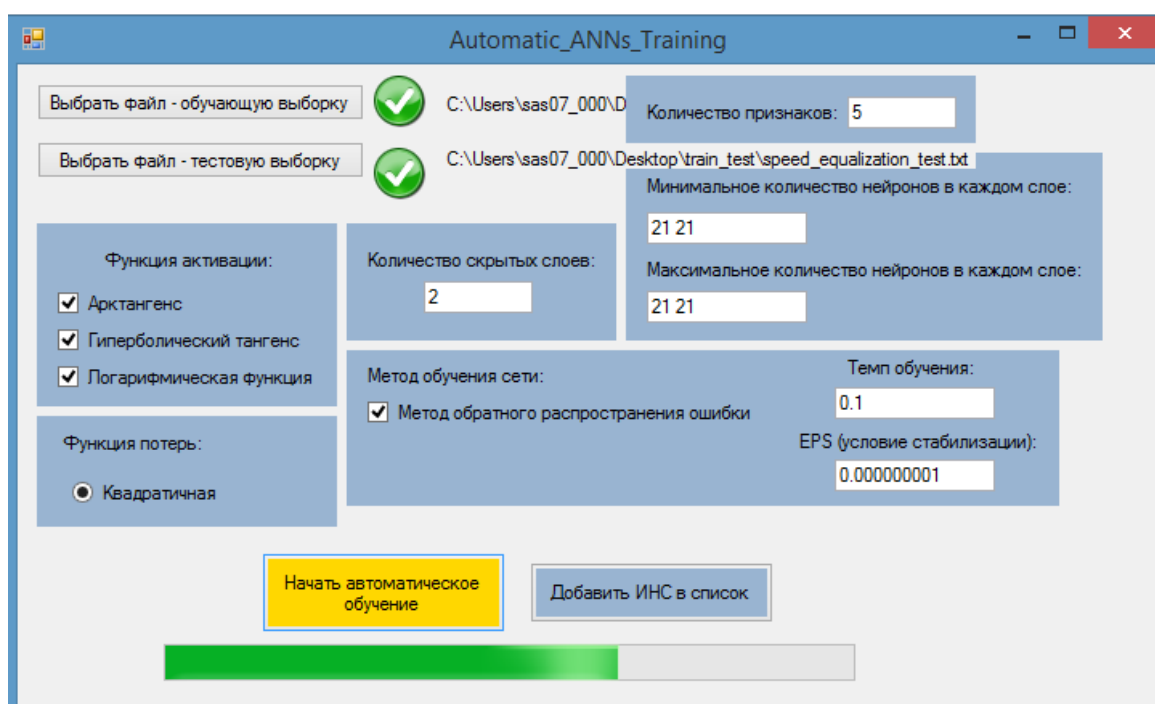


Рис. 1. Параметры обучения нейронных сетей для первого этапа преследования (выравнивание скоростей)

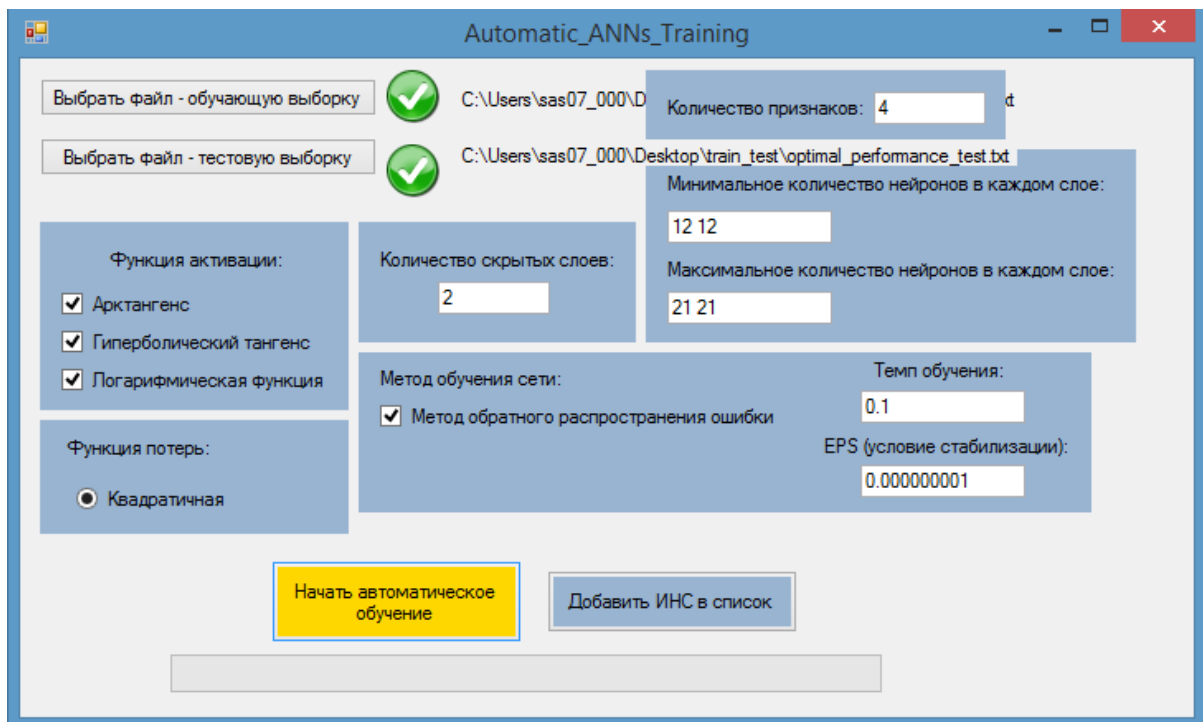


Рис. 2. Параметры обучения нейронных сетей для второго этапа преследования (решение задачи оптимального быстродействия)

№	Количество слоёв	Количество нейронов в каждом слое	Функция активации	Темп обучения	Качество на тестовой выборке
1	3	21 18 3	atan	0.1	0.00121609524727154
2	3	18 13 3	atan	0.1	0.00125695228332034
3	3	16 12 3	atan	0.1	0.00142984891662252
4	3	21 9 3	atan	0.1	0.0014807924108023
5	3	16 21 3	atan	0.1	0.0014816162762072
6	3	17 21 3	atan	0.1	0.00151725285380093
7	3	20 12 3	atan	0.1	0.00610429668389712
8	3	16 12 3	atan	0.1	0.00693337439265305
9	3	21 20 3	atan	0.1	0.00714291282171519
10	3	14 18 3	atan	0.1	0.00715964234772135
11	3	19 18 3	logarithmic	0.1	0.0074062166756242
12	3	17 13 3	atan	0.1	0.00742575092410625

Рис. 3. Список нейронных сетей в эксперименте

На рис. 3 представлены результаты эксперимента при фиксированных параметрах системы дифференциальных уравнений. Обобщенный случай, при котором значения параметров системы д.у. находятся в некотором диапазоне, рассматривается далее. В качестве значений параметров системы дифференциальных уравнений будем использовать значения, представленные на рис. 4.

Начальные параметры преследователя:

$x_0$ : 0.2     $y_0$ : 0.8     $z_0$ : 0.9     $\alpha$ : 2

$dx_0$ : 1.0     $dy_0$ : 0.0     $dz_0$ : 0.0     $\rho$ : 10

Начальные параметры убегающего:

$x_0$ : 0.0     $y_0$ : 0.0     $\beta$ : 2

$dx_0$ : -1.0     $dy_0$ : 0.0     $\sigma$ : 5

Шаг времени, настройки таймера:

Шаг реального времени (в секундах): 7E-5     $\epsilon$ : 7E-4    auto

Шаг таймера (в миллисекундах): 1  
(натуральное число)

Приращение к координате вектора управления (для убегающего) при управлении с клавиатуры: 0.1

"Наивное" преследование (нейроуправление)  
 Выравнивание скоростей и оптимальное быстродействие  
 "Наивное" преследование  
 Выравнивание скоростей и оптимальное быстродействие (нейроуправление)

ANN Tool    Additional Tools

Начать процесс моделирования

Рис. 4. Окно выбора параметров системы

Параметры обучения нейронных сетей для обобщенного случая представлены на рис. п. 5.1.

### 3.3. Результаты эксперимента

Результаты моделирования процесса «мягкой посадки» с помощью аналитического решения представлены на рис. п. 1.1. – п. 1.5. Они показывают, что мягкая посадка успешно выполняется за счет аналитических методов.

Результаты моделирования процесса «мягкой посадки» с помощью двух многослойных персептронов представлены на рис. п. 2.1. – п.2.3.

Если ограничить пространство прямоугольным параллелепипедом  $[-1, 1] \times [-1, 1] \times [0, 1]$ , то максимально возможное расстояние между объектами будет равно 3, а минимально возможное расстояние, на которое может приблизиться преследователь за счет нейросетевого управления составляет порядка 0.034 (см. рис. п. 2.3.). Т.е. в процентном соотношении погрешность составляет около 1.13%.

Результаты обучения нейронных сетей для обобщенного случая представлены на рис. п. 6.1 – п. 6.4. Мы видим, что при увеличении диапазона значений параметров системы снижается обобщающая способность нейросетевой модели. Зависимость значения минимума эмпирического риска от максимального значения параметров системы представлена в таблице 3.3.1.

Таблица 3.3.1

Зависимость значения минимума эмпирического риска от максимального значения параметров системы

№ эксперимента	1	2	3	4
Максимальные значения коэффициентов системы	0.5	0.6	0.7	0.8
Минимальный эмпирический риск для I этапа	0.00129946	0.00111457	0.00125544	0.00109172
Минимальный эмпирический риск для II этапа	0.02170835	0.03182829	0.04443000	0.05768196

## Заключение

Обучена нейросетевая модель, управляющая движением преследователя. Проведены эксперименты на данных большого объема. Результаты экспериментов показывают, что модель многослойного персептрона не применима к задаче о мягкой посадке. При увеличении диапазона значений параметров системы дифференциальных уравнений, описывающих динамику объектов, растёт значение эмпирического риска на тестовой выборке. Это свидетельствует о снижении обобщающей способности многослойного персептрона. Эта модель не обеспечивает необходимой точности управления что, приводит к отклонению от оптимальной траектории. Это происходит как при фиксированных параметрах системы, так и при вариации параметров в некотором диапазоне.

Разработанное приложение позволяет моделировать процесс мягкой посадки с помощью аналитических методов, а также при помощи нейронных сетей. Это программное средство можно использовать для дальнейших исследований задачи о мягкой посадке и других смежных задач математической теории управления. Также приложение можно использовать в образовательных целях, так как оно позволяет наглядно изобразить рассматриваемые процессы, описываемые дифференциальными уравнениями.

Результаты экспериментов позволяют исключить из рассмотрения модель многослойного персептрона в качестве модели, решающей задачу о «мягкой посадке» как с постоянными, так и с переменными параметрами системы дифференциальных уравнений.

## Список использованных источников

1. Альбус, Дж. Аналитический метод решения игровой задачи о «мягкой посадке» для движущихся объектов. / Дж. Альбус [и др.] // Кибернетика и системный анализ. — 2001. — № 1 — С. 97 – 115.
2. Воронцов, К. В. Математические методы обучения по прецедентам. Курс лекций по машинному обучению. [Электронный ресурс] / К. В. Воронцов. — 2011. — 141 с. Режим доступа: <http://www.machinelearning.ru/wiki/images/6/6d/Voron-ML-1.pdf>
3. Красовский Н. Н. Позиционные дифференциальные игры / Н. Н. Красовский, А. И. Субботин. — М. : Наука, 1974. — 340 с.
4. Муниб, М. Б. Численная реализация нейросетевого управления в задаче о мягкой посадке / М. Б. Муниб // Динамические системы — 2011. — том 1(29) №1, — С. 157 – 168.
5. Петросян, Л. А. Теория игр: Учеб. пособие для ун-тов. / Л. А. Петросян, Н. А. Зенкевич, Е. А. Семина. — М. : Высш. шк., Книжный дом «Университет», 1998. — 304 с: ил.
6. Понтрягин, Л. С. Линейные дифференциальные игры преследования / Л. С. Понтрягин // Математический сборник. 1980. — Т. 112. — № 3.
7. Чикрий, А. А. Игровая задача о «мягкой посадке» для систем второго порядка / А. А. Чикрий, А. А. Белоусов // Современная математика и ее приложения. — 2005. — Том 23. С. 166 – 180.
8. Муниб, М. Б. Использование модифицированного метода Нелдера-Мида для численного решения задачи мягкой посадки в теоретико-игровой постановке. [Электронный ресурс] / М. Б. Муниб // Инженерия программного обеспечения — 2012 — № 3 – 4 (11 – 12), — С. 27 – 30. Режим доступа: [http://irbis-nbuv.gov.ua/cgi-bin/irbis\\_nbuv/cgiirbis\\_64.exe?C21COM=2&I21DBN=UJRN&P21DBN=UJRN&IMAGE\\_FILE\\_DOWNLOAD=1&Image\\_file\\_name=PDF/Ipz\\_2012\\_3-4\\_6.pdf](http://irbis-nbuv.gov.ua/cgi-bin/irbis_nbuv/cgiirbis_64.exe?C21COM=2&I21DBN=UJRN&P21DBN=UJRN&IMAGE_FILE_DOWNLOAD=1&Image_file_name=PDF/Ipz_2012_3-4_6.pdf)
9. Chikrii A. A. Game problem on soft landing for moving objects // Proc. Int. Conf. “Intelligent Systems and Semiotics” / NIST, 1997. – Gaithersburg, USA, 1997 – p. 443-449.
10. Tao\_Framework [Электронный ресурс]. — Режим доступа: [https://ru.wikipedia.org/wiki/Tao\\_Framework](https://ru.wikipedia.org/wiki/Tao_Framework).



## Приложение 1. Примеры выполненной мягкой посадки при помощи аналитического решения

Примеры выполненной двухэтапной «мягкой посадки» при помощи аналитического решения, описанного в [1] (рис. п. 1. 1. – п. 1. 5.)

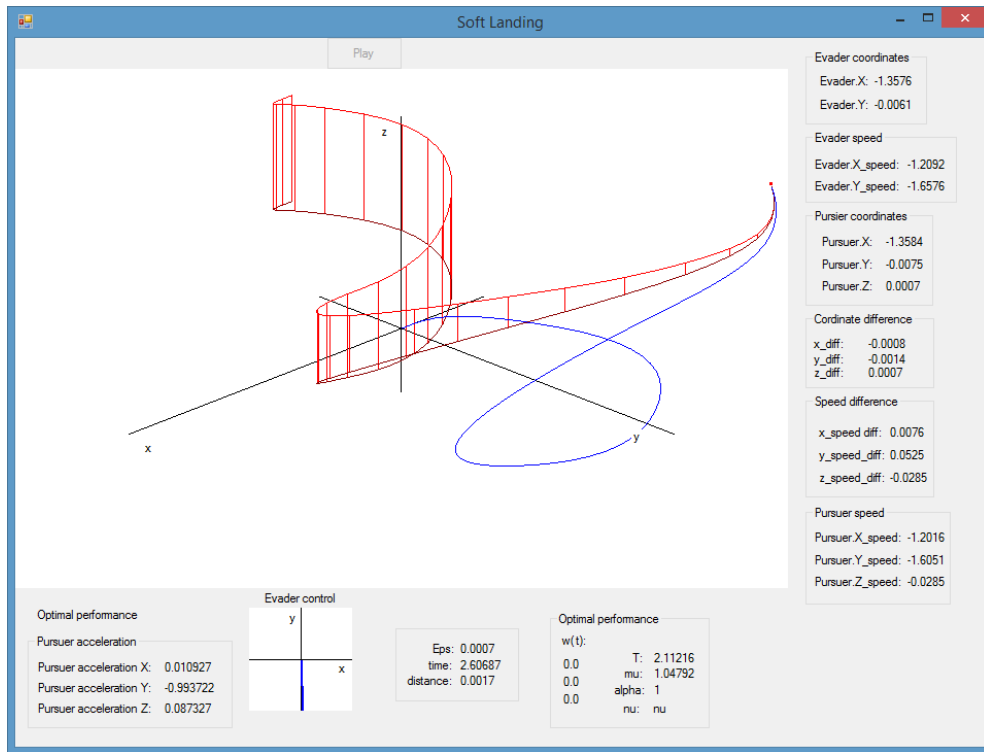


Рис. п. 1. 1.

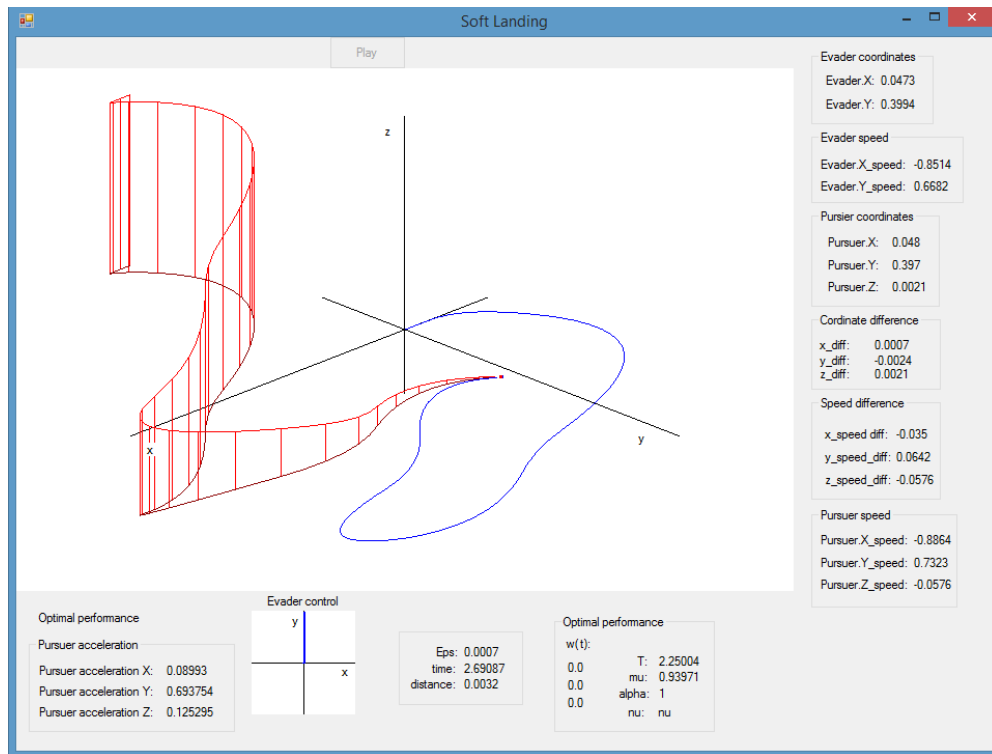


Рис. п. 1. 2.

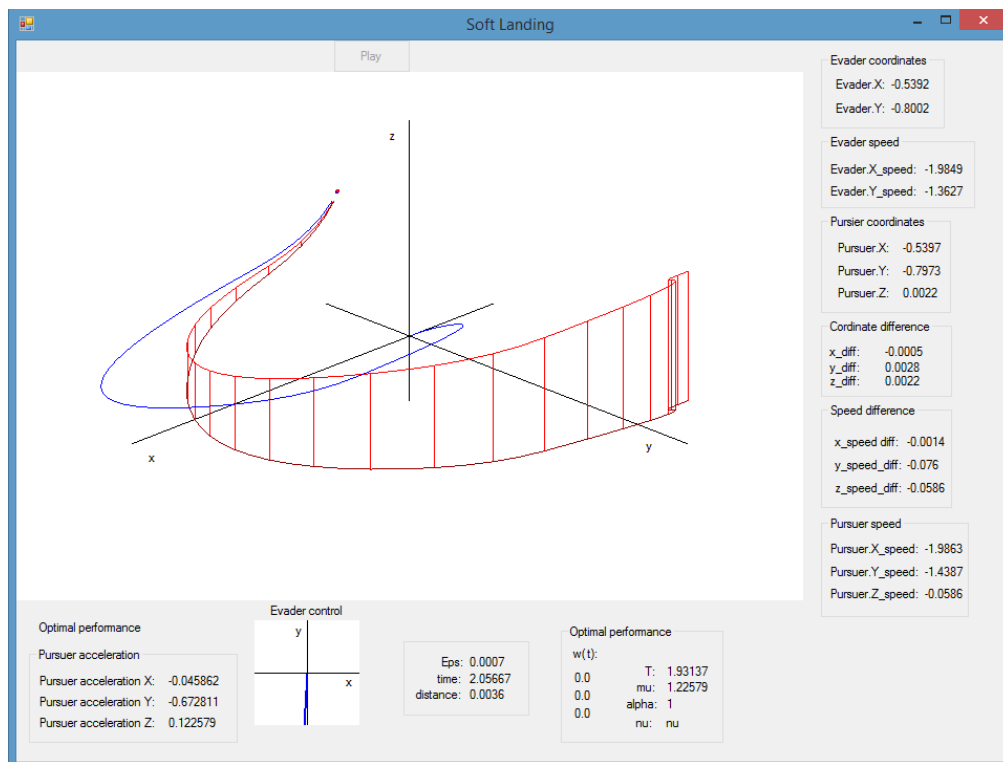


Рис. п. 1. 3.

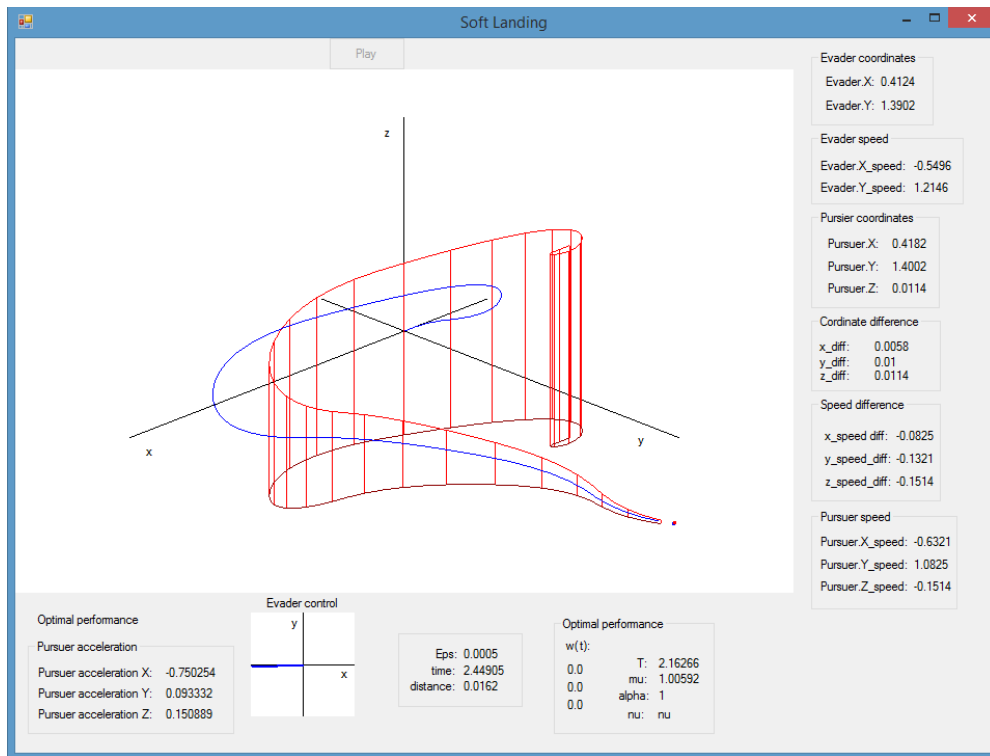


Рис. п. 1. 4.

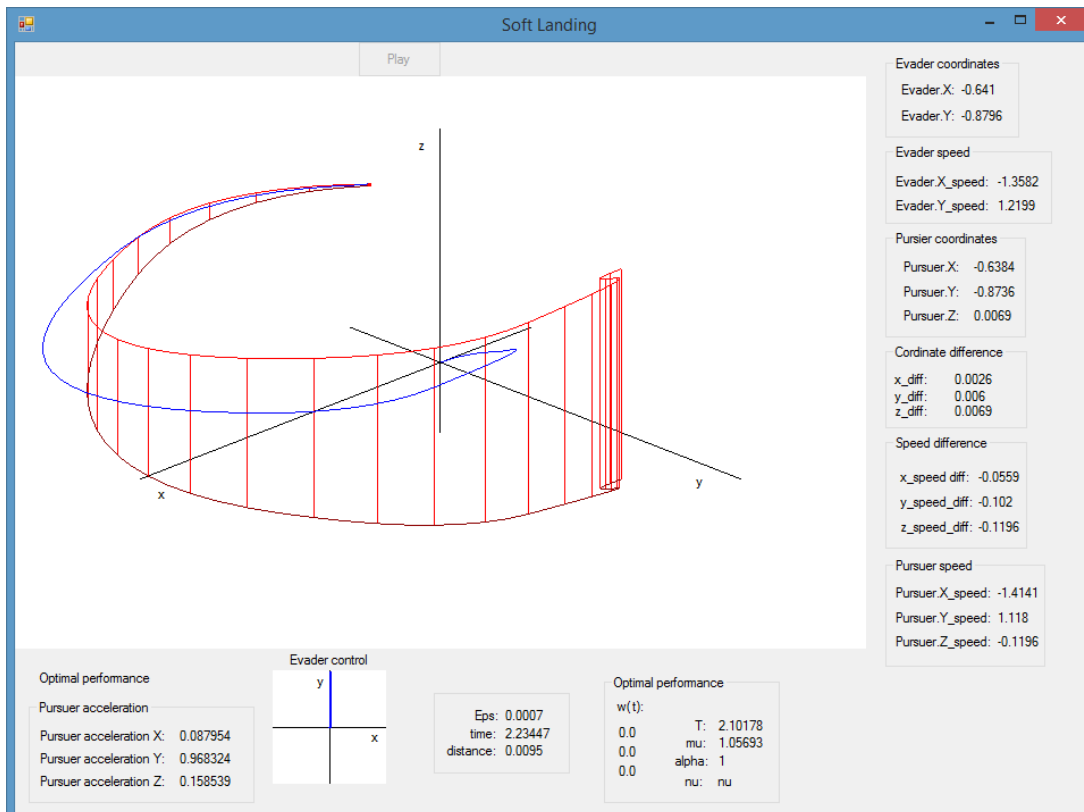


Рис. п. 1. 5.

## Приложение 2. Результаты моделирования мягкой посадки при помощи нейросетевой модели

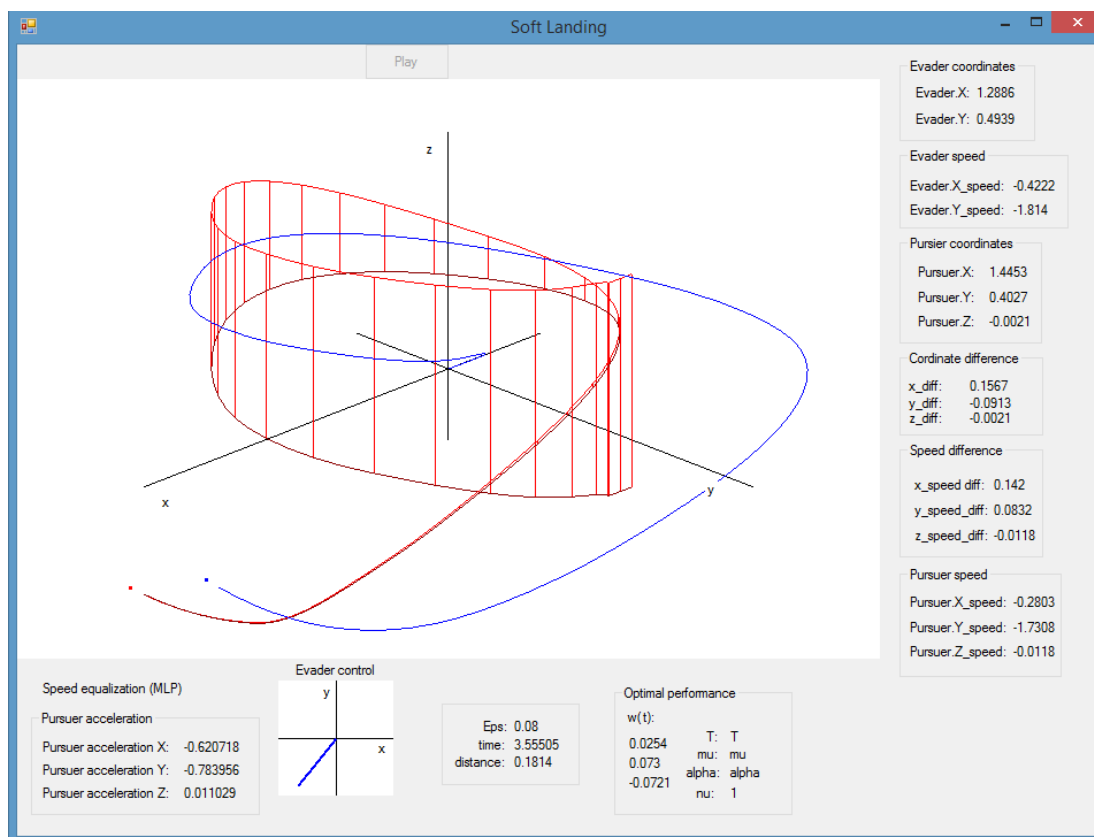


Рис. п. 2.1.

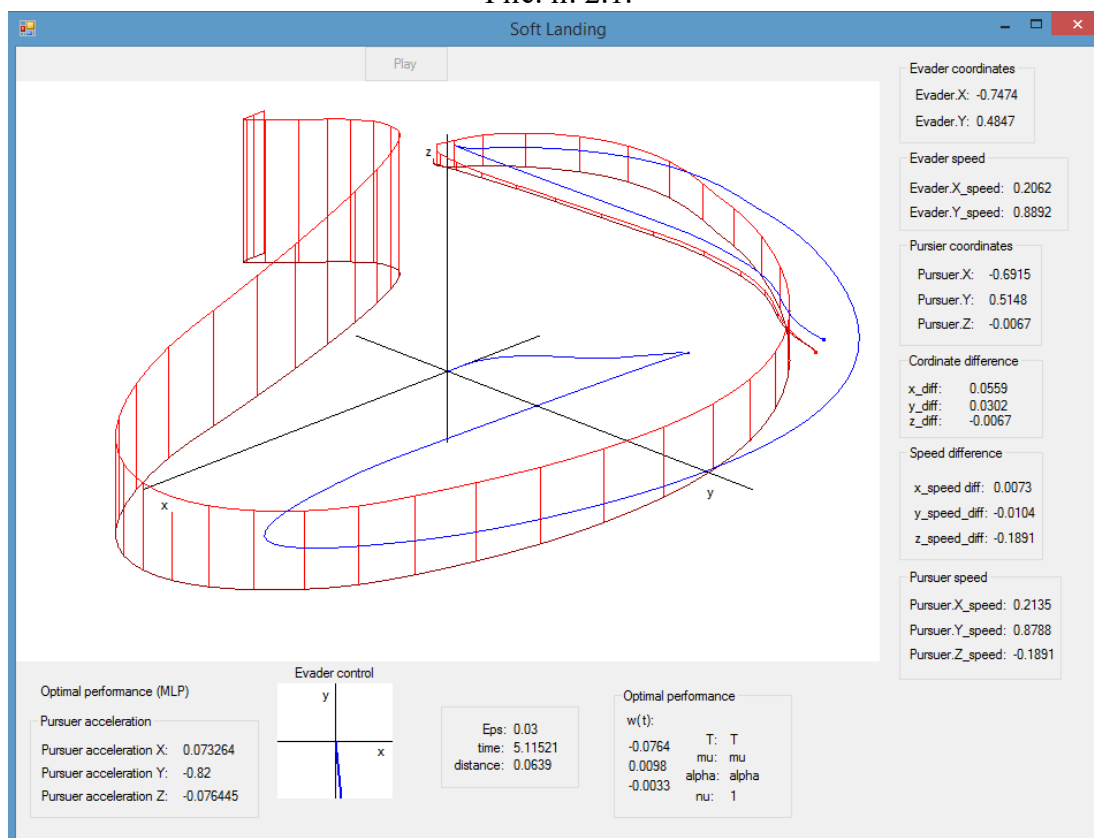


Рис. п. 2. 2.

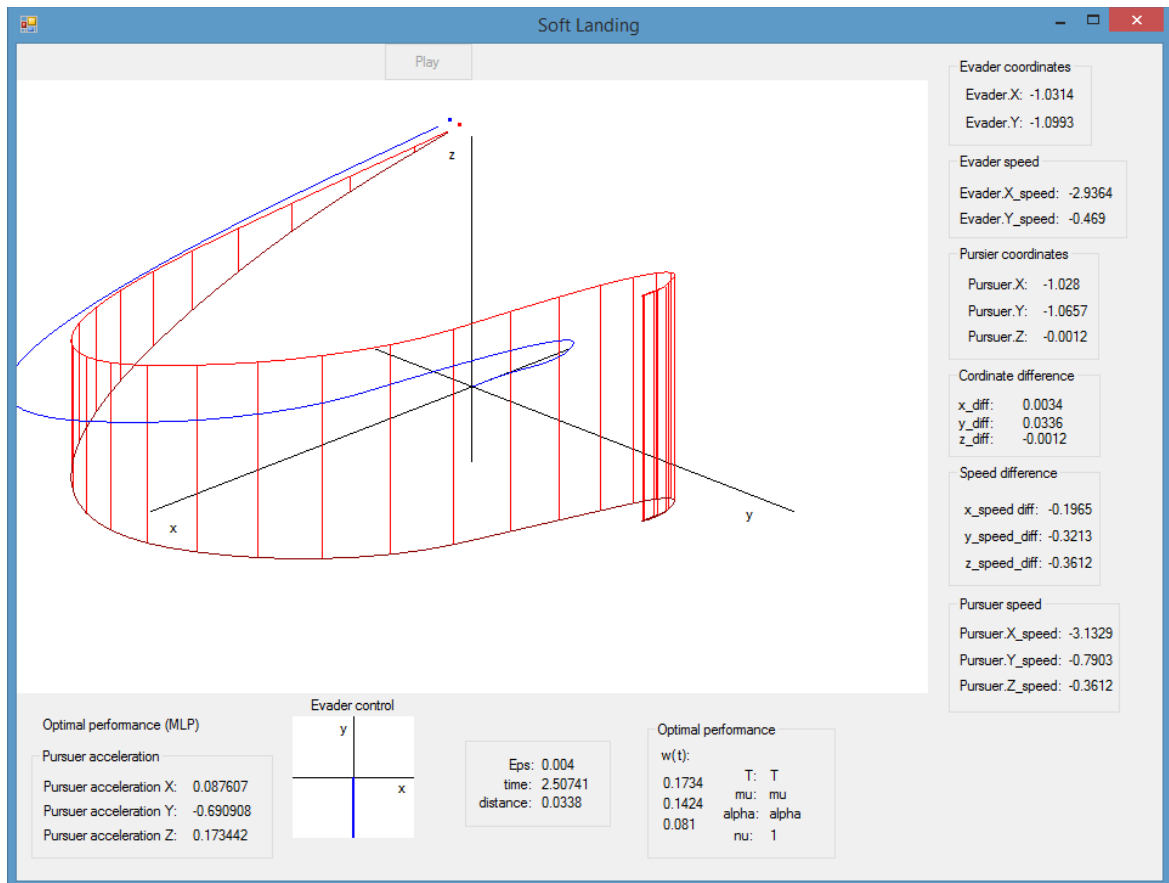


Рис. п. 2. 3.

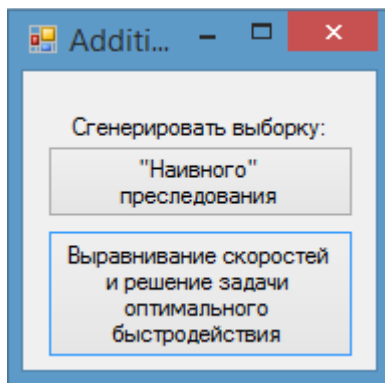


Рисунок п. 3.1. Окно выборка стратегии преследования

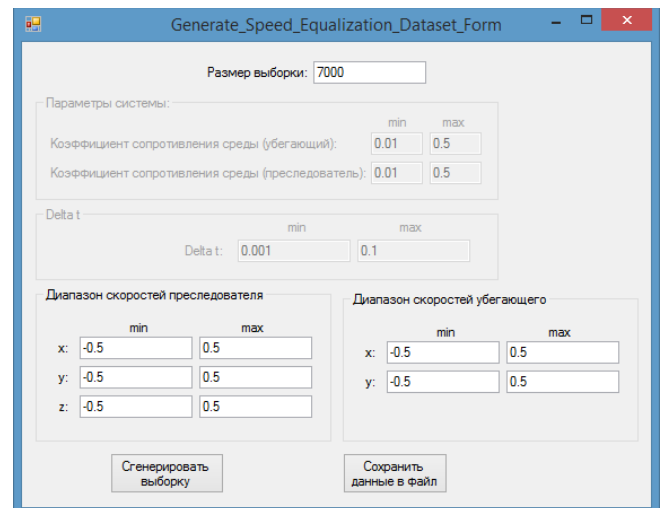


Рис. п. 3.2 Окно генератора выборки, описывающей процесс выравнивания скоростей объектов



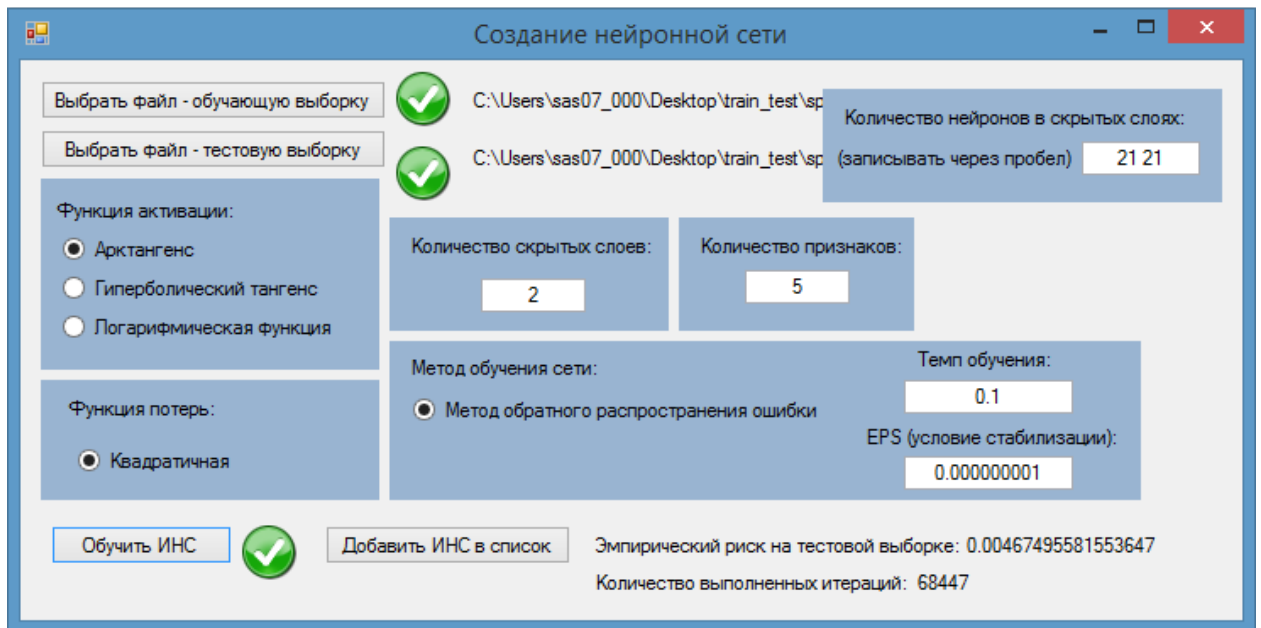


Рис. п. 4.2.

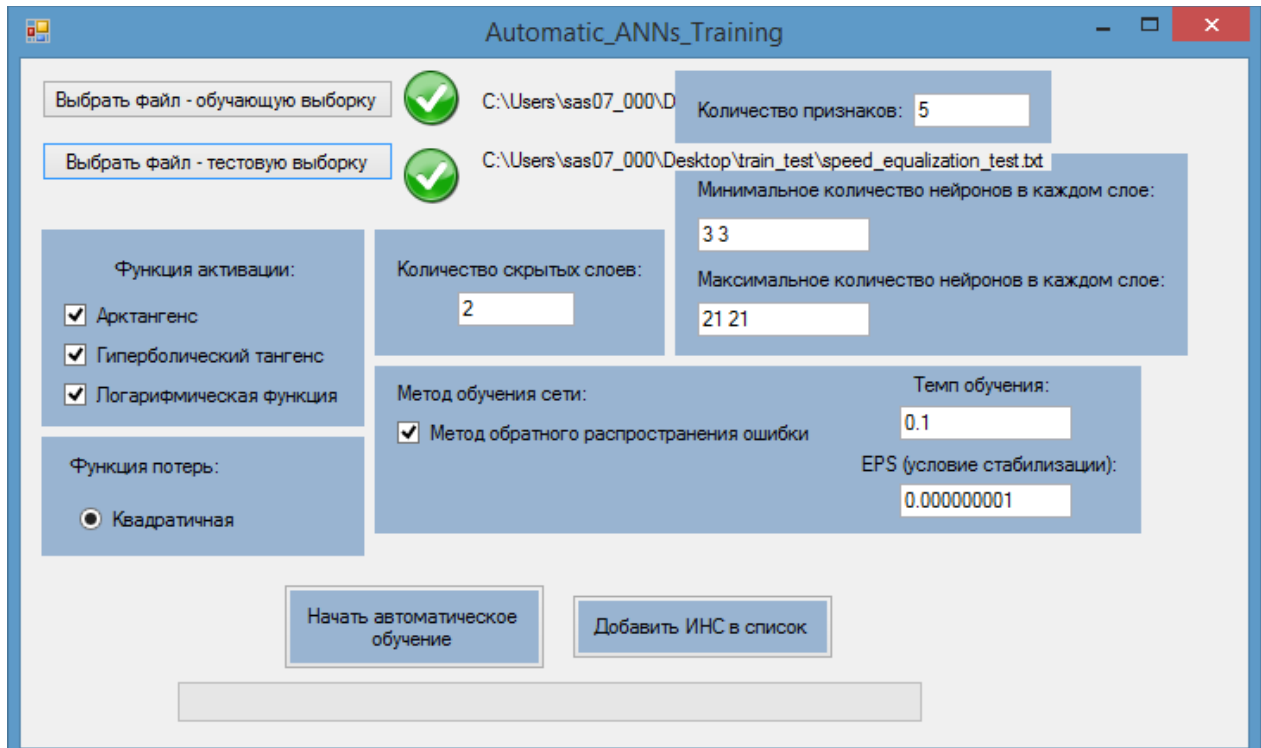


Рис. п. 4.3.

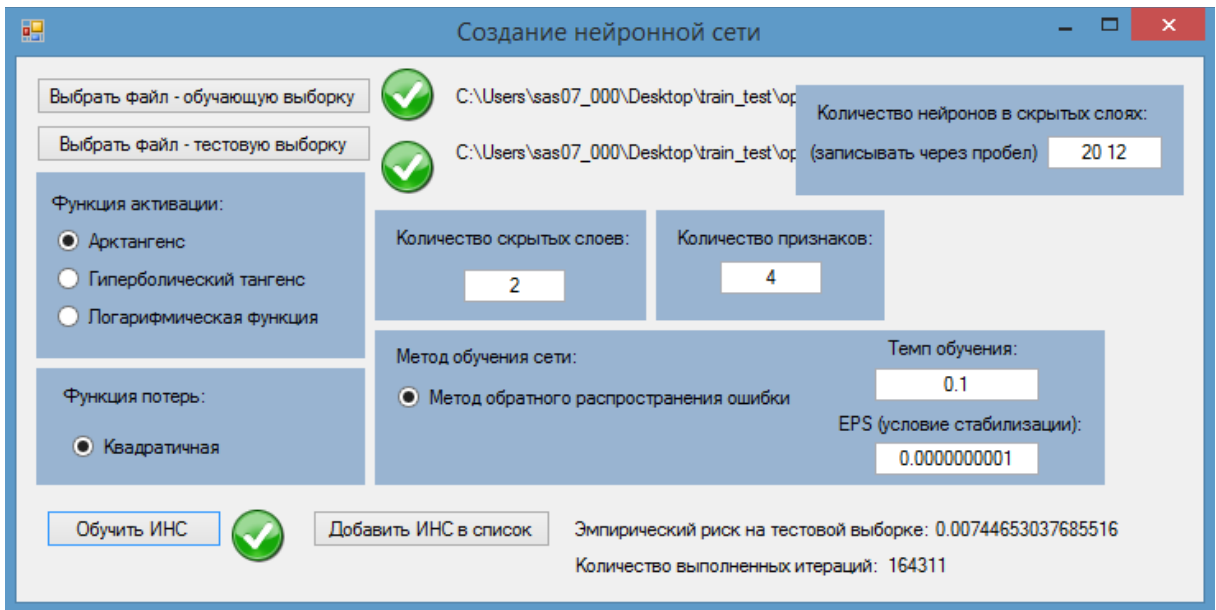


Рис. п. 4. 5.

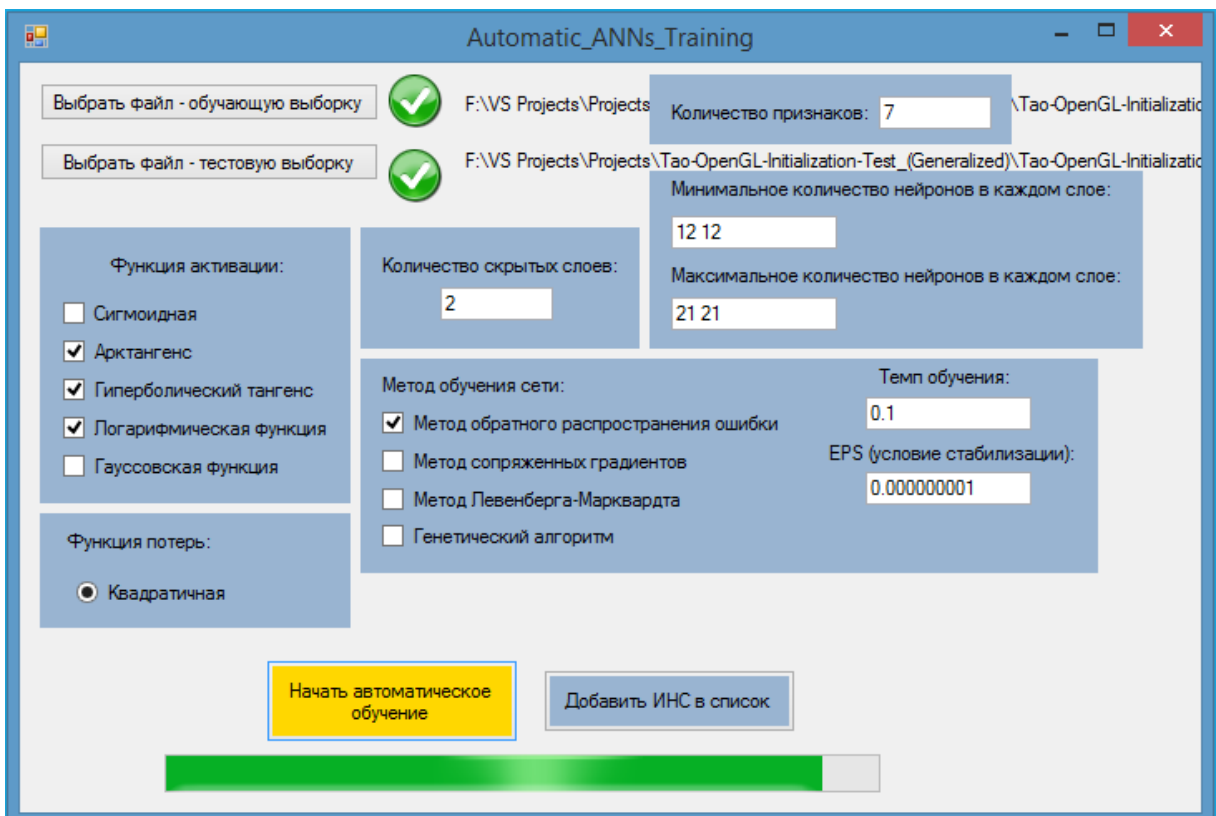


Рис. п. 5.1.



№	Количество слоёв	Количество нейронов в каждом слое	Функция активации	Темп обучения	Качество на тестовой выборке
1	3	15 16 3	atan	0.1	0.00129946121511632
2	3	13 16 3	atan	0.1	0.00138622205494414
3	3	16 17 3	atan	0.1	0.00145490952705376
4	3	14 20 3	atan	0.1	0.00145712220250709
5	3	18 12 3	atan	0.1	0.00149263039855356
6	3	16 14 3	atan	0.1	0.00162637510958547
7	3	12 13 3	atan	0.1	0.00164027281201457
8	3	18 19 3	atan	0.1	0.00168585795555447
9	3	12 13 3	logarithmic	0.1	0.0217083469842889
10	3	20 17 3	atan	0.1	0.0218432431676114
11	3	16 18 3	tanh	0.1	0.0219146264218699
12	3	15 21 3	tanh	0.1	0.0219236538319812
13	3	17 13 3	tanh	0.1	0.0219293584401005
14	3	15 19 3	tanh	0.1	0.0219346528364676

Рис п. 6.1.

№	Количество слоёв	Количество нейронов в каждом слое	Функция активации	Темп обучения	Качество на тестовой выборке
1	3	20 21 3	atan	0.1	0.00111456956566967
2	3	19 21 3	atan	0.1	0.00141270762404976
3	3	13 12 3	atan	0.1	0.00143348318337292
4	3	17 14 3	atan	0.1	0.00144673318839702
5	3	18 14 3	atan	0.1	0.00154123436914326
6	3	18 13 3	atan	0.1	0.00155906700446224
7	3	19 15 3	atan	0.1	0.00158844778217648
8	3	21 21 3	atan	0.1	0.00159822295335384
9	3	19 21 3	atan	0.1	0.0318282871614724
10	3	13 20 3	tanh	0.1	0.0318326572043087
11	3	12 16 3	tanh	0.1	0.0318677269618103
12	3	18 21 3	logarithmic	0.1	0.0318953293443236
13	3	18 12 3	atan	0.1	0.0318958660322902
14	3	19 16 3	atan	0.1	0.031895968526474

Рис п. 6.2.

№	Количество слоёв	Количество нейронов в каждом слое	Функция активации	Темп обучения	Качество на тестовой выборке
1	3	19 17 3	atan	0.1	0.00125543569168489
2	3	21 19 3	atan	0.1	0.00125899634007389
3	3	19 21 3	atan	0.1	0.00127637473807404
4	3	21 13 3	atan	0.1	0.0014000544118772
5	3	15 18 3	atan	0.1	0.0014315203542183
6	3	19 18 3	atan	0.1	0.00148244668464265
7	3	18 18 3	atan	0.1	0.00156661181636686
8	3	14 20 3	tanh	0.1	0.0444300028574653
9	3	20 21 3	tanh	0.1	0.0444357223655303
10	3	12 12 3	tanh	0.1	0.0444561124810762
11	3	21 16 3	logarithmic	0.1	0.0444648629737003
12	3	19 12 3	tanh	0.1	0.0445033967869534
13	3	20 14 3	logarithmic	0.1	0.0445070975401875
14	3	15 12 3	logarithmic	0.1	0.0445142804292707

Рис п. 6.3.

№	Количество слоёв	Количество нейронов в каждом слое	Функция активации	Темп обучения	Качество на тестовой выборке
1	3	20 12 3	atan	0.1	0.0010917177084187
2	3	12 20 3	atan	0.1	0.00139928609477705
3	3	12 17 3	atan	0.1	0.00146846313461744
4	3	13 19 3	atan	0.1	0.00150317270626905
5	3	17 17 3	atan	0.1	0.00153182992814114
6	3	17 19 3	atan	0.1	0.00161722989919631
7	3	18 18 3	atan	0.1	0.00162488771860843
8	3	20 14 3	atan	0.1	0.00166118435838404
9	3	13 13 3	logarithmic	0.1	0.0576819568544276
10	3	19 20 3	tanh	0.1	0.0578869478280089
11	3	16 13 3	logarithmic	0.1	0.0579043157340652
12	3	16 16 3	logarithmic	0.1	0.0579490215836188
13	3	17 13 3	logarithmic	0.1	0.0579763246903496
14	3	17 16 3	tanh	0.1	0.058019953104078

Рис п. 6.4.

### Приложение 3. Схема нейросетевой модели

