

## СИСТЕМНЫЙ АНАЛИЗ, УПРАВЛЕНИЕ И ОБРАБОТКА ИНФОРМАЦИИ

УДК 681.518

*Н.Н. Астахова, Л.А. Демидова*

### ПОДХОД К ПРОГНОЗИРОВАНИЮ ГРУПП ВРЕМЕННЫХ РЯДОВ С ПРИМЕНЕНИЕМ ТЕХНОЛОГИЙ КЛАСТЕРНОГО АНАЛИЗА

*Предложен подход к прогнозированию групп временных рядов с применением технологий кластерного анализа. Реализовано описание временных рядов – центроидов кластеров с использованием моделей прогнозирования на основе строго бинарных деревьев и модифицированного алгоритма клонального отбора. Показана перспективность применения общих моделей для прогнозирования временных рядов, входящих в один кластер.*

**Ключевые слова:** *временной ряд, прогнозирование, кластеризация, строго бинарное дерево, модифицированный алгоритм клонального отбора.*

**Введение.** В настоящее время наблюдается активное ускоренное развитие различных социально-экономических процессов, которые становятся все более взаимосвязанными. В связи с этим существует необходимость как более широкого применения уже известных, так и разработки новых подходов к решению задачи прогнозирования групп временных рядов (ВР), описывающих те или иные показатели процессов [1].

Всё более очевидным является тот факт, что при разработке моделей прогнозирования ВР необходимо учитывать не только тенденции прогнозируемого ВР, но и тенденции других ВР, влияющих на него. Так, в частности, в явной взаимосвязи находятся такие описываемые с помощью ВР макропоказатели, как валовой внутренний продукт и уровень экспорта, средний уровень дохода и уровень образования, уровень медицинского обеспечения и средняя продолжительность жизни.

При этом очевидно и то, что многие ВР имеют сходные законы изменения значений своих элементов и могут быть объединены в те или иные подгруппы по критерию сходства этих законов. Таким образом, целесообразна разработка общей для всех ВР каждой подгруппы модели прогнозирования с целью непосредственного использования для прогнозирования всех частных ВР подгруппы или дальнейшего уточнения общей модели прогнозирования для отдельных частных ВР (в случае недостаточной точности прогнозных значений, получаемых с применением общей модели).

При прогнозировании различных ВР именно

процесс разработки модели прогнозирования характеризуется самой высокой вычислительной трудоемкостью. Использование одной общей модели прогнозирования для подгруппы ВР может быть существенным шагом в развитии подходов к анализу коррелирующих ВР, так как позволит выполнять прогнозирование частных ВР подгруппы с приемлемыми временными затратами.

Несомненно, что применение технологий кластерного анализа, а именно алгоритмов кластеризации, таких как алгоритм четких  $c$ -средних ( $k$ -means) [2, 3], алгоритм нечетких  $c$ -средних (fuzzy  $c$ -means – FCM) [3 – 5], EM-алгоритм [6], позволит объединить сходные ВР в подгруппы (кластеры) и определить ВР-центроиды кластеров, что, по сути, и необходимо для решения задачи разработки общих моделей прогнозирования для подгрупп ВР. При этом для непосредственной разработки моделей прогнозирования ВР-центроидов кластеров может быть использован подход, предполагающий построение моделей прогнозирования на основе строго бинарных деревьев (СБД) и модифицированного алгоритма клонального отбора (МАКО) с формированием аналитических зависимостей, наилучшим образом описывающих известные значения ВР и обеспечивающих получение минимальных значений аффинитета – средней относительной ошибки прогнозирования [7, 8].

**Цель работы** – разработка подхода к прогнозированию групп временных рядов с применением технологий кластерного анализа, обеспечивающего получение прогноза с приемлемы-

ми временными затратами.

**Теоретическая часть.** Пусть имеется группа ВР  $T: t_i$  ( $i = \overline{1, m}$ ). Пусть, кроме того, каждый ВР  $t_i$  содержит  $n$  ( $10 \leq n \leq 30$ ) элементов  $t_i^j$  ( $j = \overline{1, n}$ ), значения которых измерены в упорядоченные последовательные отсчеты времени.

Значительный интерес представляет решение задачи разработки моделей прогнозирования всех ВР группы  $T$  с приемлемыми временными затратами.

Временные ряды могут быть объединены, как и другие объекты анализа данных, в кластеры (подгруппы) с учетом значений тех или иных своих характеристик (например, на основе значений математического ожидания, дисперсии и т.п., вычисленных на основе значений элементов ВР, на основе максимального или минимального значений элементов ВР).

Поскольку обычно при разработке моделей прогнозирования ВР решается задача подбора аналитической зависимости, наилучшим образом описывающей закон изменения значений элементов ВР во времени, то представляется целесообразным использовать сходство математических законов изменения значений элементов ВР во времени для объединения ВР в кластеры (подгруппы).

Предположение о возможном наличии сходства математических законов изменения значений элементов ВР, а следовательно, и о сходстве соответствующих видов аналитических зависимостей может быть обосновано тем, что многие социально-экономические показатели взаимосвязанными между собой и изменение тенденций одного из них ведет к изменению тенденций других.

Зачастую информация о том, какие ВР образуют связанные подгруппы в группе анализируемых ВР, отсутствует. В связи с этим очевидна необходимость привлечения дополнительных приемов анализа данных, например, алгоритмов кластерного анализа. Применение таких алгоритмов кластерного анализа, как алгоритм четких  $c$ -средних, алгоритм нечетких  $c$ -средних, EM-алгоритм, позволяет выполнить в ходе итерационных вычислений разбиение группы объектов на заранее заданное количество кластеров  $c$  в соответствии с некоторым критерием оптимальности и определить координаты центроидов кластеров.

При решении задачи кластеризации ВР координаты центроидов кластеров могут быть использованы для формирования обобщающих ВР-центроидов, характеризующих частные ВР, входящие в соответствующие кластеры (подгруппы). Очевидно, что для ВР-центроида возможна разработка так называемой общей модели прогнозирования, которая может быть использована в дальнейшем либо непосредственно для про-

гнозирования всех частных ВР, отнесенных к кластеру, либо в качестве базовой модели с целью ее последующего уточнения и применения для прогнозирования отдельных частных ВР.

Для минимизации временных затрат на разработку общих моделей прогнозирования предлагается при решении задачи кластеризации ВР использовать алгоритм четких  $c$ -средних [3], который реализует разбиение группы объектов  $T$  на подгруппы  $T_r$  ( $r = \overline{1, c}$ ) так, что [3]:

$$\bigcup_{r=1}^c T_r = T; \quad (1)$$

$$T_r \cap T_h = \emptyset; \quad r = \overline{1, c}; \quad h = \overline{1, c}; \quad r \neq h; \quad (2)$$

$$\emptyset \subset T_r \subset T, \quad r = \overline{1, c}. \quad (3)$$

Именно ВР следует понимать под объектом при решении задачи кластеризации ВР. Пусть  $u_r(t_i)$  – характеристическая функция, которая может принимать значение 0, если объект  $t_i$  не принадлежит кластеру  $T_r$ , и значение 1, если объект  $t_i$  принадлежит кластеру. В этом случае четкое  $c$ -разбиение группы объектов  $T$  на кластеры (подгруппы) может быть описано матрицей  $U = [u_r(t_i)]$  ( $u_r(t_i) \in \{0, 1\}$ ;  $r = \overline{1, c}$ ;  $i = \overline{1, m}$ ).

При этом матрица  $U = [u_r(t_i)]$  должна удовлетворять следующим условиям:

$$\sum_{r=1}^c u_r(t_i) = 1 \quad (i = \overline{1, m}); \quad (4)$$

$$0 < \sum_{i=1}^m u_r(t_i) < m \quad (r = \overline{1, c}). \quad (5)$$

Алгоритм четких  $c$ -средних осуществляет минимизацию целевой функции:

$$J(U, V) = \sum_{r=1}^c \sum_{i=1}^m u_r(t_i) \cdot d^2(v_r, t_i), \quad (6)$$

где  $U = [u_r(t_i)]$  – четкое  $c$ -разбиение группы объектов  $T$  на основе характеристических функций  $u_r(t_i)$ , определяющих принадлежность объекта  $t_i$  кластеру  $T_r$ ;  $V = (v_1, \dots, v_c)$  – центроиды кластеров;  $d(v_r, t_i)$  – расстояние между центром кластера  $v_r$  и объектом  $t_i$ ;  $c$  – количество кластеров  $T_r$ ;  $m$  – количество объектов;  $r = \overline{1, c}$ ;  $i = \overline{1, m}$ .

Реализация алгоритма четких  $c$ -средних предполагает выполнение следующих шагов [3].

1. Инициализация начального четкого  $c$ -разбиения  $U = [u_r(t_i)]$ , удовлетворяющего условиям (4) и (5).

2. Вычисление координат центроидов кластеров:

$$v_r^j = \frac{1}{m_r} \cdot \sum_{i=1}^m u_r(t_i) \cdot t_i^j, \quad (7)$$

где  $m_r$  – количество объектов, отнесенных к  $r$ -му

кластеру;  $j = \overline{1, n}$ .

3. Вычисление нового четкого  $c$ -разбиения  $U = [u_r(t_i)]$ , удовлетворяющего условиям (4) и (5).

4. Шаги 2 и 3 повторяются до достижения наперед заданной точности  $\varepsilon$ :  $|J(U, V) - J'(U, V)| \leq \varepsilon$ , где  $J(U, V)$ ,  $J'(U, V)$  – значения целевой функции (6) на двух последовательных итерациях (или пока не будет выполнено наперед заданное количество итераций  $H$ ).

Если обычно для вычисления расстояния  $d(v_r, t_i)$  между центром кластера  $v_r$  и объектом  $t_i$  используется евклидова метрика и  $d(v_r, t_i)$  вычисляется как  $d(v_r, t_i) = (\sum_{j=1}^n (v_r^j - t_i^j)^2)^{0.5}$ , где  $n$  – количество характеристик объекта, то при решении задачи кластеризации ВР на основе сходства математических законов изменения значений элементов ВР предлагается выполнить модификацию этой метрики таким образом, чтобы обеспечить учет различной актуальности элементов ВР: меньшей актуальности одних элементов ВР (наиболее удаленных во времени от момента прогнозирования) и большей актуальности других элементов ВР (наиболее близких во времени к моменту прогнозирования):

$$d(v_r, t_i) = (\sum_{j=1}^n \frac{j}{n} \cdot (v_r^j - t_i^j)^2)^{0.5}. \quad (8)$$

Необходимость такой модификации объясняется тем, что с течением времени зависимости между социально-экономическими показателями изменяются, поэтому при построении моделей прогнозирования большее предпочтение следует отдавать самым близким к моменту прогнозирования элементам ВР. Использование весовых коэффициентов вида  $\frac{j}{n}$  позволяет считать наиболее значимыми расхождения между значениями самых актуальных элементов ВР (например, при  $j = n$  значение весового коэффициента равно 1, а при  $j = 1$  – равно  $\frac{1}{n}$ ).

Применение формулы (8) позволит не только удовлетворить требование учета актуальности элементов ВР, но и обеспечить объединение ВР в кластеры (подгруппы) на основе сходства тенденций (благодаря высокой чувствительности к разнонаправленности тенденций ВР).

Принципиальным в случае применения технологий кластерного анализа является решение проблемы, связанной с разной масштабностью анализируемых ВР, характеризующих те или иные социально-экономические показатели, имеющие различные единицы измерения, различные диапазоны изменения и соответствующие им статистические характеристики (математическое ожидание и т.п.).

Решение этой проблемы возможно посредством использования алгоритмов нормализации, которые широко применяются в статистике, математической экономике и эконометрике.

Суть этих алгоритмов заключается в определении некоторого среднего уровня, относительно которого выравниваются все анализируемые ВР [1]. В качестве такого среднего уровня может выступать какая-либо условная прямая, один из ВР анализируемой группы ВР или же ВР-центроид  $S$ , элементы  $S_j$  которого определяются как:

$$S_j = \frac{1}{m} \cdot \sum_{i=1}^m t_i^j, \quad (9)$$

где  $t_i^j$  –  $j$ -й элемент  $i$ -го ВР;  $i = \overline{1, m}$ ;  $j = \overline{1, n}$ ;  $m$  – количество ВР;  $n$  – количество элементов ВР-центроида.

Реализация алгоритма нормализации  $i$ -го ВР ( $i = \overline{1, m}$ ) на основе ВР-центроида  $S$  предполагает выполнение следующих шагов.

#### Шаг 1.

1.1. Определение среднего шага  $hS$  изменения элементов ВР-центроида:

$$hS = \frac{\max_{j=1, n}(S_j) - \min_{j=1, n}(S_j)}{n}, \quad (10)$$

где  $n$  – количество элементов ВР-центроида;  $S_j$  –  $j$ -й элемент ВР-центроида  $S$ .

1.2. Определение среднего шага  $ht_i$  изменения элементов  $i$ -го ВР ( $i = \overline{1, m}$ ):

$$ht_i = \frac{\max_{j=1, n}(t_i^j) - \min_{j=1, n}(t_i^j)}{n}, \quad (11)$$

где  $n$  – количество элементов ВР;  $t_i^j$  –  $j$ -й элемент  $i$ -го ВР.

#### Шаг 2.

2.1. Определение среднего уровня ВР-центроида:

$$\bar{S} = \frac{\sum_{j=1}^n S_j}{n}, \quad (12)$$

где  $n$  – количество элементов ВР-центроида;  $S_j$  –  $j$ -й элемент ВР-центроида.

2.2. Определение среднего уровня  $\bar{t}_i$   $i$ -го ВР ( $i = \overline{1, m}$ ):

$$\bar{t}_i = \frac{\sum_{j=1}^n t_i^j}{n}, \quad (13)$$

где  $n$  – количество элементов ВР;  $t_i^j$  –  $j$ -й элемент  $i$ -го ВР.

#### Шаг 3.

Вычисление величины  $dt_i^j$ , представляющей собой отношение разности между средним

уровнем  $\bar{t}_i$   $i$ -го ВР и  $j$ -м элементом  $t_i^j$   $i$ -го ВР к среднему шагу  $ht_i$ :

$$\Delta t_i^j = \frac{\bar{t}_i - t_i^j}{ht_i}, \quad (14)$$

где  $t_i^j$  –  $j$ -й элемент  $i$ -го ВР;  $\bar{t}_i$  – средний уровень  $i$ -го ВР;  $ht_i$  – средний шаг изменения элементов  $i$ -го ВР;  $i = \overline{1, m}$ ;  $j = \overline{1, n}$ .

Величина  $\Delta t_i^j$  может быть интерпретирована как количество шагов (возможно, не целое) длиной  $ht_i$ , приходящихся на интервал длиной  $|\bar{t}_i - t_i^j|$ .

#### Шаг 4.

Преобразование каждого  $j$ -го элемента  $t_i^j$   $i$ -го ВР к виду:

$$t_i^j = \bar{S} + \Delta t_i^j \cdot hS. \quad (15)$$

В дальнейшем преобразованные таким образом ВР могут быть использованы для группирования в кластеры (подгруппы) с использованием алгоритма четких  $c$ -средних.

Поскольку центры кластеров выражают общие тенденции для ВР, формирующих соответствующие кластеры (подгруппы), то целесообразной является разработка моделей прогнозирования для ВР-центроидов кластеров.

Известны различные подходы к построению моделей прогнозирования ВР. Одним из перспективных является подход, реализующий использование различных эволюционных алгоритмов (генетических алгоритмов [9 – 11], алгоритмов клонального отбора [7, 8, 12, 13] и т.п.), основанных на принципах естественного отбора и обеспечивающих построение моделей прогнозирования ВР, наилучшим образом описывающих известные значения ВР. При этом построение моделей прогнозирования осуществляется с приемлемыми временными затратами.

При решении задачи разработки моделей прогнозирования для ВР-центроидов кластеров целесообразным является применение модифицированного алгоритма клонального отбора, моделирующего законы функционирования естественной иммунной системы и обеспечивающего формирование достаточно сложных аналитических зависимостей [8, 12].

В [7, 8, 12] дано описание принципов разработки моделей прогнозирования  $k$ -го порядка с применением МАКО, позволяющего сформировать при приемлемых временных затратах на основе СБД аналитическую зависимость, наилучшим образом описывающую известные значения ВР и обеспечивающую получение минимального значения аффинитета (affinity)  $Aff$  – средней относительной ошибки прогнозирования (Average Forecasting Error Rate)  $AFER$ :

$$AFER = \frac{\sum_{j=k+1}^n |(f^j - d^j)/d^j|}{n-k} \cdot 100\%, \quad (16)$$

где  $d^j$  и  $f^j$  – соответственно реальное и предсказанное значения для  $j$ -го элемента ВР (для  $j$ -го отсчета времени);  $n$  – количество элементов ВР (количество отсчетов времени).

При решении задачи прогнозирования подгрупп ВР в качестве  $d^j$  следует использовать  $j$ -й элемент  $t_i^j$   $i$ -го ВР или  $j$ -й элемент  $v_r^j$  ВР-центроида  $r$ -го кластера.

Допустимые варианты аналитических зависимостей представляются в виде антител  $Ab$ , осуществляющих распознавание антигенов  $Ag$  (в данном случае – известных значений элементов ВР), а в качестве «лучшего» антитела  $Ab$  выбирается то, которое обеспечивает минимальное значение аффинитета  $Aff$  [7, 8, 13, 14].

Кодирование антитела  $Ab$  осуществляется посредством записи в строку символов, выбираемых из трёх символьных алфавитов [8, 12]: алфавита арифметических операций (операций сложения, вычитания, умножения и деления) –  $Operation = \{ '+', '-', '·', '/' \}$ ; алфавита функционалов  $Functional = \{ 'S', 'C', 'Q', 'L', 'E', '_' \}$ , в котором символы  $S', C', Q', L', E'$  определяют математические функции «синус», «косинус», «квадратный корень», «натуральный логарифм», «экспонента», а символ '\_' – отсутствие математической функции; алфавита терминалов  $Terminal = \{ 'a', 'b', \dots, 'z', '?' \}$ , в котором символы  $'a', 'b', \dots, 'z'$  определяют аргументы аналитической зависимости, а символ '?' – константу.

Использование трёх символьных алфавитов обеспечивает корректное преобразование в аналитические зависимости случайным образом формируемых антител, структура которых может быть описана с помощью СБД [7, 8, 12, 13].

Максимально возможный порядок  $K$  моделей прогнозирования ( $K \geq k$ , где  $k$  – реальный порядок модели) определяется количеством символов из алфавита терминалов  $Terminal$  в антителе  $Ab$  [10, 11]. В этом случае при прогнозировании значения элемента  $d^j$  ВР на момент времени  $j$  могут использоваться  $K$  значений элементов ВР:  $d^{j-K}, \dots, d^{j-2}, d^{j-1}$ .

Использование СБД вида, приведенного на рисунке 1, позволяет строить сложные аналитические зависимости и обеспечивает высокую точность прогнозирования ВР [8, 12, 13]. Искомое СБД может быть сформировано в результате композиции одного «левого» поддерева максимально возможного порядка  $K = 3$  и некоторого количества «правых» поддерева максимум возможного порядка  $K = 2$ . При этом термин

«левое» поддерево («правое» поддерево) используется для указания того, в какую ветвь (в левую или в правую) некоторого уровня СБД следует включать новое поддерево [12, 13].

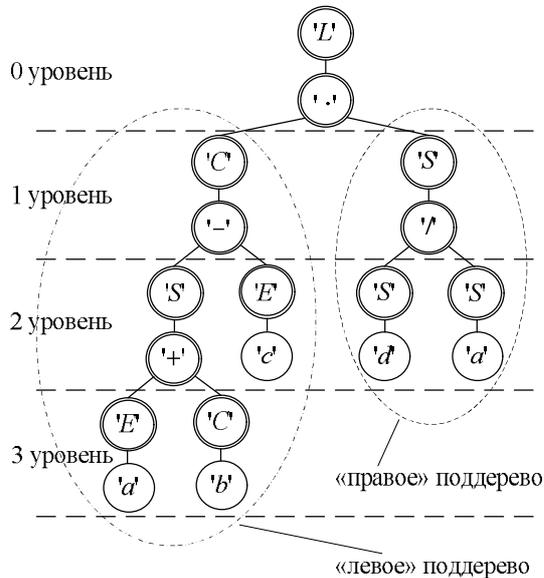


Рисунок 1 – Пример строго бинарного дерева

При формировании антитела целесообразно сначала реализовать разбиение СБД на поддеревья, далее выполнить обход вершин каждого поддерева с формированием упорядоченных списков символов, находящихся в его вершинах, а затем – последовательное объединение этих списков [12]. При формировании упорядоченного списка символов на основе поддерева осуществляется последовательный двухкратный обход его вершин. При этом сначала при движении по поддереву снизу вверх слева направо попарно обходятся вершины, содержащие символы из алфавита терминалов *Terminal*, и соответствующие им расположенные сверху вершины, содержащие символы из алфавита функционалов *Functional*. Затем при движении в том же направлении (снизу вверх слева направо) попарно обходятся вершины, содержащие символы из алфавита арифметических операций *Operation*, и соответствующие им расположенные сверху вершины, содержащие символы из алфавита функционалов *Functional* [12].

Сформированное таким образом антитело (при движении по нему слева направо) в качестве первых двух символов содержит пару символов нулевого уровня СБД из алфавита функционалов *Functional* и алфавита арифметических операций *Operation*. Затем в антителе располагаются списки символов, соответствующие «правым» поддеревьям максимально возможного порядка  $K = 2$  (при движении по СБД сверху вниз), и, наконец, список символов, входящих в «левое» поддерево максимально возможного порядка  $K = 3$  [12].

При использовании такого способа формирования антитела обеспечивается наглядность

представления структуры СБД в виде объединения поддеревьев, а само антитело легко интерпретируется в аналитическую зависимость.

В частности, антитело, сформированное на основе СБД, приведенного на рисунке 1, кодируется строкой символов:

$$L \cdot S / SeSdC - S + EaCbEa, \quad (17)$$

которая легко преобразуется к аналитической зависимости:

$$f(a, b, c, d) = \ln(\cos(\sin(\exp(a) + \cos(b))) - \exp(c)) \cdot \sin(\sin(d) / \sin(a)). \quad (18)$$

В случае разработки модели прогнозирования  $k$ -го порядка при  $k = 4$  с учетом упорядоченности символов  $a, b, c, d$  в алфавите терминалов *Terminal* аналитическая зависимость (18) может быть записана как:

$$f(d^{j-1}, d^{j-2}, d^{j-3}, d^{j-4}) = \ln(\cos(\sin(\exp(d^{j-1}) + \cos(d^{j-2})) - \exp(d^{j-3})) \cdot \sin(\sin(d^{j-4}) / \sin(d^{j-1}))). \quad (19)$$

При интерпретировании антител в аналитические зависимости целесообразно использовать рекурсивную процедуру интерпретирования [8].

МАКО, применяемый для поиска «лучшего» антитела, определяющего «лучшую» аналитическую зависимость, включает в себя подготовительную часть, реализующую формирование начальной популяции антител, и итерационную часть, предполагающую: упорядочение антител по возрастанию значений аффинитета *Aff*; отбор и клонирование части «лучших» антител, характеризующихся наименьшими значениями аффинитета *Aff*; гипермутацию клонов антител; самоуничтожение клонов антител, «похожих» на другие клоны и антитела текущей популяции; вычисление аффинитета клонов антител и формирование новой популяции антител; супрессию полученной популяции; генерацию новых антител и добавление их к текущей популяции до получения ее исходного размера; проверку условия завершения работы МАКО. Подробное описание МАКО дано в работе [8].

Можно утверждать, что использование общих моделей прогнозирования, построенных для ВР-центроидов кластеров, позволяет обеспечить индивидуальность результатов прогнозирования для всех частных ВР, отнесенных к кластеру, поскольку при вычислении прогнозных значений для частных ВР на основе аналитических зависимостей, определяющих модели прогнозирования для ВР-центроидов кластеров, в качестве значений переменных выступают значения элементов частных ВР [15].

С учетом вышеизложенного подход к прогнозированию групп ВР, обеспечивающий получение прогноза с приемлемыми временными затратами, может быть реализован при выполнении следующих этапов.

**Этап 1.** Нормализация всех ВР группы относительно их среднего уровня, выраженного

BP-центроидом  $S$ .

**Этап 2.** Кластеризация нормализованных BP наперед заданное количество кластеров  $c$ .

**Этап 3.** Построение общих моделей прогнозирования для BP-центроидов кластеров

на основе СБД и МАКО.

**Этап 4.** Прогнозирование исходных частных BP с использованием общих моделей прогнозирования для BP-центроидов кластеров.

#### Результаты прогнозирования макропоказателей

№ п/п	Наименование показателя	Единица измерения	AFER, %	2012		2013		2014		Ошибка, %
				прогноз	факт	прогноз	факт	прогноз	факт	
1-й кластер										
1	Потребление энергии	Кг.нефт.эquiv.	1,70	4740	5105,56	5022	4733,09	5114	5013,35	5,09
2	Потребление электроэнергии	кВт*ч/чел.	0,74	6279	6476,5	6457	6269,51	6485	6447,64	2,21
3	Валовой национальный доход на душу населения по методу Атласа	\$	5,02	9642	12730,34	10406	9633,48	11740	10397,98	15,06
4	Валовой национальный доход на душу населения по паритету покупательной способности	\$	3,25	19373	22700,58	20861	19363,23	22279	20852,1	9,75
2-й кластер										
5	Добавленная стоимость в сфере услуг	% от ВВП	0,36	59,59	58,2	59,82	59,2	59,52	59,59	1,18
6	Экспорт товаров и услуг	% от ВВП	0,61	29,04	30,3	29,03	29,6	29,05	29,04	2,04
7	Импорт товаров и услуг	% от ВВП	0,29	21,9	21,7	21,87	22,3	21,89	21,9	0,97
8	Валовое накопление капитала	% от ВВП	0,23	24,95	25	24,91	24,5	24,84	24,95	0,77
9	Доходы (за исключением грантов)	% от ВВП	0,97	29,04	31,3	29,14	29,8	28,96	29,04	3,24
10	Коэффициент подростковой фертильности	Число рождений/1000 женщин	2,26	28,97	26,4	29	25,7	28,98	28,97	7,54
11	Добавленная стоимость в промышленности	% от ВВП	2,91	34,15	37,4	35,12	36,8	36,16	34,15	6,38
3-й кластер										
12	Запуск процедур для регистрации бизнеса	Кол.	1,48	7,95	8	7	8	7,82	7,95	4,92
13	Экспорт высоких технологий	%	1,36	8,27	8	8,95	8,4	8,57	8,27	4,52
14	Добавленная стоимость в сельском хозяйстве	% от ВВП	6,03	13,6	11,2	13,47	10,6	11,99	13,6	20,11
15	Смертность в возрасте до 5 лет	%	2,99	3,92	4,36	3,08	3,79	3,93	3,89	9,95
4-й кластер										
16	Количество деторождений, происшедших при помощи квалифицированного медицинского персонала	%	0,24	99,54	98,96	99,49	98,86	99,82	98,68	0,79
17	Иммунизация против кори	%	0,06	97,68	97,43	97,68	97,71	97,72	97,39	0,21
18	Процент населения, имеющего начальное образование	%	0,27	96,4	96,69	97,09	97,94	95,19	96,69	0,91
19	Соотношение девочек и мальчиков в системе начального и среднего образования	%	0,34	98,28	97,71	98,79	98,02	99,29	97,29	1,14
20	Улучшенные источники воды	%	0,21	96,91	97,38	96,74	97,46	96,66	97,51	0,70
21	Ожидаемая продолжительность жизни при рождении	Число лет	0,72	72,02	70,68	72,83	70,73	73,13	71,48	2,39
22	Улучшенные средства санитарии	%	1,21	74,96	71,38	74,86	70,08	74,65	74,45	4,03

Следует отметить, что прогнозирование ВР с использованием общих моделей прогнозирования не ведет к получению общих для кластера (подгруппы) ВР прогнозов. Общая модель прогнозирования определяет лишь математический закон изменения значений элементов ВР посредством формируемой с применением МАКО аналитической зависимости.

Сами же прогнозные значения для каждого частного ВР будут индивидуальными, поскольку будут вычисляться при подстановке в общую модель прогнозирования известных значений элементов частного ВР.

**Экспериментальные исследования.** Апробация предлагаемого подхода к прогнозированию групп ВР была выполнена с использованием ВР 22 макроэкономических показателей Российской Федерации, взятых с сайта World DataBank за период с 1999 г. по 2014 г. (<http://databank.worldbank.org/data/views/reports/tableview.aspx?isshared=true#>).

С использованием алгоритма четких  $s$ -средних, в котором расстояния между ВР вычислялись в соответствии с (8), все показатели были разделены на 4 кластера (подгруппы ВР), информация о содержимом которых представлена в первом столбце таблицы.

Модели прогнозирования для частных ВР каждого кластера были определены на основе антител, сопоставленных ВР-центроидам кластеров:

$_{-}C \cdot _{-} _{-}C \cdot E/E - Q/_{-}bLhCf0iSeCaCbQiQ'$ ;  
 $L - C \cdot Q/_{-}C \cdot C - E/Q - EcSeS?QcQ?QfEiCdQc$ ;  
 $L + _{-}C - _{-}S - Q - C \cdot C \cdot _{-}iEcEdQhSeEbCbQ?Sg$ ;  
 $L + Q + Q \cdot E + C \cdot C/Q + S \cdot Si \_{-}bE?Ca \_{-}fC?QcEfCe$  ,  
 в форме соответственно аналитических зависимостей вида:

$$f_1(d^{j-1}, d^{j-2}, d^{j-3}, d^{j-4}, d^{j-5}, d^{j-6}) =$$

$$= \cos(\exp(\sqrt{\sqrt{d^{j-3}}}/\sqrt{d^{j-1}}) - \cos(d^{j-5})) -$$

$$- \exp(\cos(d^{j-6})/\sin(d^{j-4})) \cdot \cos(d^{j-1} \cdot \cos(d^{j-3})) -$$

$$- \ln(d^{j-2}) - d^{j-5};$$

$$f_2(d^{j-1}, d^{j-2}, d^{j-3}, d^{j-4}, d^{j-5}) =$$

$$= \ln(\cos(\sqrt{\sqrt{\sqrt{d^{j-5}}}} -$$

$$- \cos(d^{j-4}))/\exp(d^{j-1}) - \cos(\sqrt{d^{j-2}} - \sqrt{25,87})) \cdot$$

$$\cdot \cos(\sqrt{d^{j-5}} \cdot \sin(-103,17))/\sin(d^{j-3}) - \exp(d^{j-5});$$

$$f_3(d^{j-1}, d^{j-2}, d^{j-3}, d^{j-4}, d^{j-5}, d^{j-6}) =$$

$$= \ln(\sqrt{\sqrt{\sqrt{\sin(\cos(d^{j-3})) \cdot \exp(d^{j-2}))}} +$$

$$+ \sqrt{d^{j-4}}) + \cos(\cos(2,14)/d^{j-2}) + \cos(\cos(d^{j-6}) \cdot$$

$$\cdot \exp(224,07)) \cdot \exp(d^{j-5} + \sin(d^{j-1}));$$

$$f_4(d^{j-1}, d^{j-2}, d^{j-3}, d^{j-4}, d^{j-5}) =$$

$$= \ln(\exp(\exp(\exp(\sin(\ln(d^{j-4})) \cdot \sin(d^{j-3}))/\exp(129,43)) -$$

$$- \cos(\ln(d^{j-3}) + \sqrt{d^{j-2}})) \cdot \cos(\sqrt{d^{j-5}} +$$

$$+ d^{j-1})) \cdot \sqrt{d^{j-4}} - \exp(d^{j-1})).$$

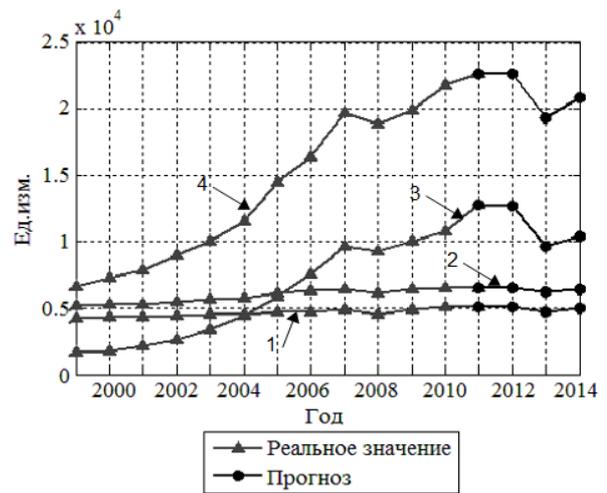


Рисунок 2 – Результаты прогнозирования для временных рядов 1-го кластера

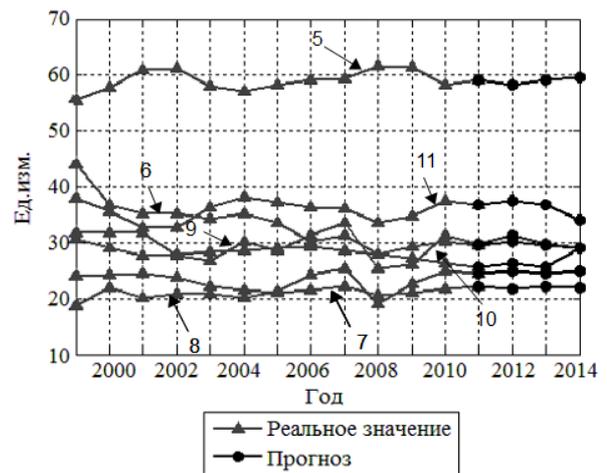


Рисунок 3 – Результаты прогнозирования для временных рядов 2-го кластера

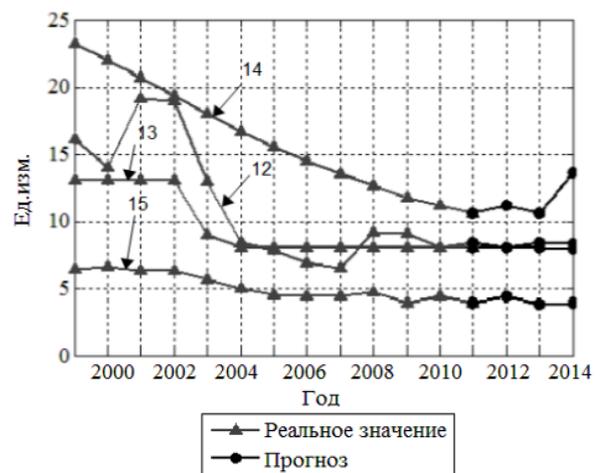
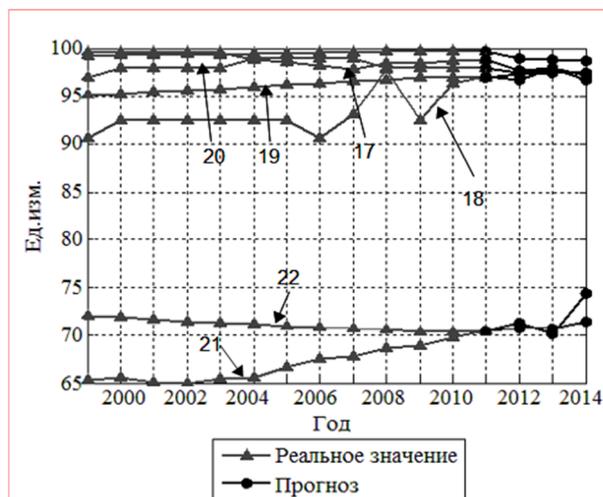


Рисунок 4 – Результаты прогнозирования для временных рядов 3-го кластера

При этом данные за период с 1999 г. по 2011 г. были использованы для разработки моделей прогно-

зирования, а данные за период с 2011 г. по 2014 г. – для прогнозирования частных ВР на 3 шага.

Результаты прогнозирования частных ВР на 3 шага вперед, значения средних относительных ошибок прогнозирования на 3 шага вперед, а также значения *AFER* (16) (за период с 1999 г. по 2011 г.) приведены в таблице.



**Рисунок 5 – Результаты прогнозирования для временных рядов 4-го кластера**

Усредненное по всем частным ВР значение *AFER* (16) составило 1,43 %, а усредненное по всем частным ВР значение средней относительной ошибки прогнозирования на 3 шага вперед оказалось равным 4,48 %, что незначительно больше, чем в случае построения индивидуальной модели прогнозирования для каждого частного ВР.

При этом использование предлагаемого подхода к прогнозированию групп ВР позволило с точностью 99,33 % определить будущее значение валового накопления капитала (в % от ВВП) и с точностью 99,09 % определить процент населения, имеющего начальное образование (за период с 2011 г. по 2014 г.).

Необходимо отметить, что с приемлемой точностью были определены значения прогноза для частных ВР, принадлежащих одному кластеру, но имеющих существенные различия относительно среднего уровня значений своих элементов. При этом за счет использования формулы (8) для вычисления расстояния между ВР была снижена актуальность значений элементов ВР, наиболее удаленных от момента разработки моделей прогнозирования.

На рисунках 2 – 5 представлены графические зависимости для значений элементов частных ВР, отнесенных соответственно к кластерам 1, 2, 3 и 4 (в том числе отображены значения прогноза на 3 шага вперед). Все ВР пронумерованы в соответствии с нумерацией, приведенной в таблице, а под обозначением «Ед. изм.» по верти-

кальной оси на рисунках следует понимать единицу измерения соответствующего частного ВР (третий столбец таблицы).

При использовании предлагаемого подхода к прогнозированию групп ВР в приведенном выше примере временные затраты на непосредственное построение моделей прогнозирования были снижены в 5,5 раза (поскольку вместо 22 моделей были построены всего 4). Даже с учетом того, что при реализации данного подхода имеют место дополнительные временные затраты, обусловленные необходимостью выполнения процедуры кластеризации ВР при разном количестве кластеров  $s$  с целью нахождения оптимального разбиения, определяемого по минимуму значения целевой функции (6) алгоритма кластеризации, можно сделать вывод о существенном снижении временных затрат на прогнозирование групп ВР.

**Заключение.** Предлагаемый подход к прогнозированию групп ВР реализует последовательное применение алгоритма четких  $s$ -средних для кластеризации группы анализируемых ВР и принципов разработки моделей прогнозирования на основе СБД и МАКО для построения общих моделей прогнозирования для ВР-центроидов кластеров. При этом обеспечивается получение индивидуальных прогнозных значений для всех ВР группы с приемлемыми временными затратами.

Результаты экспериментальных исследований, полученные в ходе прогнозирования макроэкономических показателей Российской Федерации, подтверждают перспективность применения и дальнейшего развития предлагаемого подхода.

Использование технологий кластерного анализа позволяет образовывать кластеры (подгруппы) связанных ВР, имеющих сходные законы изменения значений своих элементов, и обеспечивает увеличение скорости прогнозирования ВР. Применение общих моделей прогнозирования, сформированных для ВР-центроидов кластеров, для прогнозирования отдельных частей ВР, входящих в соответствующие кластеры (подгруппы), не приводит к существенному снижению точности прогнозирования. Требуемая точность прогноза для частного ВР может быть достигнута в результате уточнения общей модели прогнозирования с применением МАКО.

#### **Библиографический список**

1. Андерсен Т. Статистический анализ временных рядов. М.: Мир, 1976. 756 с.
2. Паклин Н.Б., Орешков В.И. Бизнес-аналитика от данных к знаниям. СПб.: Питер, 2013. 704 с.
3. Демидова Л.А., Кураковский В.В., Пылькин А.Н. Принятие решений в условиях неопределенности. М.: Горячая линия-Телеком, 2012. 288 с.

4. Чубукова И.А. Data Mining: учеб. пособие. М.: Интернет-университет информационных технологий: БИНОМ: Лаборатория знаний, 2006. 382 с.
5. Штовба С.Д. Проектирование нечетких систем средствами Matlab. М.: Горячая линия – Телеком, 2007. 288 с.
6. Воронцов К.В., Потапенко А.А. Модификации EM-алгоритма для вероятностного тематического моделирования // Машинное обучение и анализ данных. 2013. Т. 1. № 6. С. 657-686.
7. Демидова Л.А., Корячко А.В., Скворцова Т.С. Модифицированный алгоритм клонального отбора для анализа временных рядов с короткой длиной актуальной части // Системы управления и информационные технологии, 2010. Т. 42. № 4.1. С. 131-136.
8. Демидова Л.А. Модели прогнозирования временных рядов с короткой актуальной частью на основе модифицированного алгоритма клонального отбора // Вестник Рязанского государственного радиотехнического университета. 2012. № 39-2. С. 64-71.
9. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы, нечеткие системы / пер. с польск. И.Д. Рудинского. М.: Горячая линия-Телеком. 2004. 452 с.
10. Демидова Л.А. Разработка однофакторных нечетких моделей для анализа тенденций временных рядов с использованием генетического алгоритма // Научно-технические ведомости СПбГПУ. 2007. № 52-2. С. 156-164.
11. Демидова Л.А. Прогнозирование тенденций рынка труда на основе однофакторных нечетких временных рядов // Системы управления и информационные технологии. 2007. Т. 29. № 3.2. С. 241-246.
12. Астахова Н.Н., Демидова Л.А. Использование почти полных строго бинарных деревьев и модифицированного алгоритма клонального отбора при разработке моделей прогнозирования временных рядов с короткой актуальной частью // Вестник Рязанского государственного радиотехнического университета. 2013. № 4-2 (46). С. 89-96.
13. Демидова Л.А. Оценка качества моделей прогнозирования на основе строго бинарных деревьев и модифицированного алгоритма клонального отбора // Cloud of Science. 2014. Т. 1. № 2. С. 202-222.
14. Demidova L.A. Time series forecasting models on the base of modified clonal selection algorithm // В сборнике: 2014 International conference on computer technologies in physical and engineering applications (ICCTPEA) Editor: E. I. Veremey. Санкт-Петербургский государственный университет; IEEE (IEEE Catalog number CFP14BDA-USB). 2014. С. 33-34.
15. Astakhova N.N. Time series forecasting by means of modified clonal selection algorithm and clustering mechanisms // В сборнике: Modern informatization problems in simulation and social technologies Proceedings of the XX-th International Open Science Conference (Yelm, WA, USA, January 2015). Editor in Chief Dr. Sci., Prof. O.Ja. Kravets. Yelm, WA, USA, 2015. С. 146-150.

УДК 517.977.58, 519.668

*А.И. Бобиков, Д.М. Грушин*

## ИНСТРУМЕНТАРИЙ, ПРОЕКТИРУЮЩИЙ ЗСУР РЕГУЛЯТОР ДЛЯ КОНЕЧНОГО ГОРИЗОНТА ВРЕМЕНИ В СРЕДЕ GUI СИСТЕМЫ MATLAB

*Рассматривается задача, целью которой является разработка инструментария для синтеза регулятора нелинейных систем, представленных моделью «вход-состояние-выход». При проектировании регулятора применяется метод, основанный на решении зависящего от состояния уравнения Риккати (ЗСУР) для конечного горизонта времени. Применение ЗСУР регулятора позволяет решать задачу слежения. В инструментарии реализована возможность проектирования системы управления с наблюдателем состояния, который основан на решении уравнения ЗСУР для конечного горизонта времени.*

**Ключевые слова:** задача слежения, ЗСУР регулятор для конечного горизонта времени, наблюдатель ЗСУР для конечного горизонта времени.

**Введение.** Модели, которые адекватно описывают свойства объекта, почти всегда являются нелинейными.

Существует множество нелинейных методов управления: бэкстеппинг, метод функций управ-

ления Ляпунова, линеаризация обратной связью и др. [1]. Область применения большинства из них достаточно узка [1].

Среди методов нелинейного управления быстро развивающимся и перспективным является

ся субоптимальный метод ЗСУР.

Он стал интересным инструментом для систематической разработки нелинейных регуляторов. Метод ЗСУР позволяет компенсировать нелинейности в системе [2]. Отметим следующие достоинства метода ЗСУР:

- ✓ субоптимальность управления при относительно малых вычислительных затратах [2];
- ✓ простота настройки регулятора;
- ✓ устойчивость;
- ✓ робастность [3, 4, 5].

В статье [5] описан большой потенциал метода ЗСУР, с помощью которого проектируется субоптимальное управление для бесконечного горизонта времени. В статье [6] представлен модифицированный метод ЗСУР, позволяющий решать задачу слежения. Основная идея подхода – решить уравнение ЗСУР для конечного горизонта времени. Применение замены переменной [7] в статье [8] позволяет конвертировать дифференциальное уравнение Риккати (ДУР) в линейное дифференциальное уравнение Ляпунова [9]. Преимущества проектирования ЗСУР управления для конечного горизонта времени перед проектированием ЗСУР управления для бесконечного горизонта времени заключаются в уменьшении вычислительных затрат, исчезновении необходимости рассматривать свойства управляемости и наблюдаемости на бесконечном горизонте времени, решении проблемы сингулярности.

В связи с трудностью поиска ЗСУР управления для конечного горизонта времени в аналитическом виде для систем высокого порядка возникает необходимость создания инструментария, в котором применяется численный метод. В результате использования численного метода получаем набор значений сигнала управления в дискретные моменты времени. Следующая причина разработки инструментария – желание избавить проектировщика от необходимости создавать для каждой нелинейной системы отдельную имитационную модель в среде SIMULINK. Также причиной разработки приложения послужило желание позволить проектировщику сохранять и загружать результаты работы.

Идея создания инструментария для проектирования различных регуляторов, основанных на методах теории автоматического управления, не нова. Например, в [10] разрабатывается инструментарий для синтеза ЗСУР управления для бесконечного горизонта времени. Актуальность работы заключается в том, что на данный момент не существует приложения, позволяющего синтезировать ЗСУР регулятор, решающий задачу слежения для конечного горизонта времени без

применения наблюдателя состояния и с использованием наблюдателя состояния.

*Цель работы* – реализовать инструментарий в пакете MATLAB, который позволит проектировать ЗСУР управление для конечного горизонта времени без наблюдателя состояния и ЗСУР управление для конечного горизонта времени с наблюдателем состояния, а также проводить имитационное моделирование замкнутых систем.

**Теоретические сведения о методе ЗСУР, решающем задачу слежения.** Рассмотрим следующую задачу нелинейного оптимального управления, где модель автономной, нелинейной по состоянию и аффинной по управлению системы представлена в виде [3]:

$$\dot{x}(t) = f(x) + g(x)u(t), \quad x_0 = x(0), \quad (1)$$

$$f(0) = 0, \quad g(x) \neq 0,$$

с вектором состояния  $x = [x_1, \dots, x_n]^T \in R^n$ , сигналом управления  $u \in R^m$ , матрицами  $f(x) \in R^n$  и  $g(x) \in R^{n \times m}$ .

В результате расширенной линеаризации [3] модель нелинейной системы (1) можно представить как

$$\dot{x}(t) = A(x)x + B(x)u, \quad x_0 = x(0). \quad (2)$$

Уравнение (2) имеет линейную структуру с матрицами  $A(x)$  и  $B(x)$  такими, что

$$f(x) = A(x)x, \quad g(x) = B(x). \quad (3)$$

Для систем выше первого порядка непрерывные нелинейные матрицы  $A \in R^{n \times n}$  и  $B \in R^{n \times m}$  неоднозначно определяются с помощью расширенной линеаризации [4].

Фактором, влияющим на качество работы замкнутой системы, является выбор весовых матриц  $Q$  и  $R$ , входящих в уравнение Риккати и являющихся параметрами проектирования [12].

Пусть  $v(t)$  – задающее воздействие. Цель состоит в том, чтобы найти закон управления обратной связи по состоянию, обеспечивающий равенство  $v(t) = y(t)$ , где выход

$$y = C(x)x, \quad (4)$$

а  $C(x) \in R^{p \times n}$  – матрица выхода. Для синтеза такого закона управления минимизируется функция стоимости, определяемая формулой:

$$J(x, u) = \frac{1}{2} (e'(t_f) F e(t_f) + \int_{t_0}^{t_f} [e'(t) Q(x) e(t) + u'(x) R(x) u(x)]), \quad (5)$$

где  $t_0$  – начальное значение времени,  $t_f$  – конечное значение времени, ошибка слежения  $e(t) = v(t) - y(t)$ , весовая матрица  $F$  является положительной полуопределенной симметричной, также как матрица  $Q(x)$ , а весовая матрица  $R(x)$  – симметричной положительно определенной. Кроме того, подинтегральное слагаемое  $e'(t)Q(x)e(t)$  является мерой точности управления, а  $u'(x)R(x)u(x)$  является мерой величины сигнала управления [6].

Прежде чем приступить к решению задачи слежения, рассмотрим 3 условия, выполнение которых необходимо для синтеза ЗСУР управления.

*Условие 1.*  $f(x)$  является непрерывно дифференцируемым вектором, зависящим от  $x$ .

*Условие 2.* Состояние  $x = 0$  является точкой равновесия системы, такой, что  $f(0) = 0$  при  $u = 0$ .

*Условие 3.* Пара  $\{A(x), B(x)\}$  является поточечно управляемой или стабилизируемой для  $\forall x$  [4].

Для решения задачи слежения требуется минимизировать функцию стоимости (5) и закон управления при этом должен иметь вид [6]:

$$u(x) = -R^{-1}(x)B'(x)[P_c(x)x(t) - g(x)], \quad (6)$$

где при выполнении условий 1-3  $P_c(x)$  – симметричное положительно определенное единственное решение зависящего от состояния дифференциального уравнения Риккати

$$\begin{aligned} -\dot{P}_c(x) &= P_c(x)A(x) + A'(x)P_c(x) - \\ &- P_c(x)B(x)R^{-1}(x)B'(x)P_c(x) + \\ &+ C'(x)Q(x)C(x). \end{aligned} \quad (7)$$

Конечное значение решения уравнения (7) определяется как

$$P_c(x(t_f)) = C'(x(t_f))FC(x(t_f)). \quad (8)$$

В результате ЗСУР управления для конечного горизонта времени траектория вектора состояния становится решением дифференциального уравнения

$$\begin{aligned} \dot{x} &= [A(x) - B(x)R^{-1}(x)B'(x)P_c(x)] \\ &x + B(x)R^{-1}(x)B'(x)g(x), \end{aligned} \quad (9)$$

где  $g(x)$  является решением зависящего от состояния неоднородного вектора дифференциальных уравнений

$$\begin{aligned} \dot{g}(x) &= -[A(x) - B(x)R^{-1}(x)B'(x)P_c(x)]' \\ &g(x) - C'(x)Q(x)v(t), \end{aligned} \quad (10)$$

с конечным значением

$$g(x(t_f)) = C'(x(t_f))Fv(x(t_f)). \quad (11)$$

Решение уравнения (7) с конечным значением (8) не может быть вычислено с помощью обратного интегрирования от  $t_f$  к  $t_0$ , так как неизвестно значение вектора состояния в будущие моменты времени. Для преодоления этой проблемы применяется приближенная аналитическая аппроксимация [7, 9], которая преобразует исходное нелинейное дифференциальное уравнение Риккати в дифференциальное уравнение Ляпунова. В каждый момент времени может быть решено уравнение Ляпунова в замкнутой форме [7].

Для того чтобы решить дифференциальное уравнение Риккати (7) и неоднородное дифференциальное уравнение (10), необходимо осуществить следующие шаги.

1. Решим алгебраическое уравнение Риккати

$$\begin{aligned} P_{ss}(x)A(x) + A'(x)P_{ss}(x) - \\ - P_{ss}(x)B(x)R^{-1}(x)B'(x)P_{ss}(x) + \\ + Q(x) = 0, \end{aligned} \quad (12)$$

чтобы вычислить значение устойчивого состояния  $P_{ss}(x)$ .

2. Вычислим значение матрицы замкнутой системы

$$A_{cl}(x) = A(x) - B(x)R^{-1}(x)B'(x)P_{ss}(x). \quad (13)$$

3. Решим алгебраическое уравнение Ляпунова относительно  $D(x)$  [12]

$$\begin{aligned} A_{cl}(x)D(x) + D(x)A'_{cl}(x) - \\ - B(x)R^{-1}(x)B'(x) = 0. \end{aligned} \quad (14)$$

4. Используем технологию замены переменных и предположим, что

$$K(x) = [P_c(x) - P_{ss}(x)]^{-1}. \quad (15)$$

5. Найдем  $K(x)$  с помощью дифференциального уравнения Ляпунова

$$\begin{aligned} \dot{K}(x) &= K(x)A'_{cl}(x) + A_{cl}(x)K(x) - \\ &- B(x)R^{-1}(x)B'(x). \end{aligned} \quad (16)$$

Решение уравнения (16) имеет следующий вид

$$\begin{aligned} K(x, t) &= \exp(A_{cl}(x)(t - t_f)) \cdot [K(x, t_f) - \\ &- D(x)] \cdot \exp(A'_{cl}(x)(t - t_f)) + D(x), \end{aligned} \quad (17)$$

где  $K(x, t_f) = [P_c(x, t_f) - P_{ss}(x)]^{-1}$ . Для вычисления  $K(x, t_f)$  необходимо рассчитать

$P_c(x, t_f)$  по формуле (8).

6. Вычислим  $P_c(x)$  из уравнения (15) как

$$P_c(x) = K^{-1}(x) + P_{ss}(x). \quad (18)$$

7. Определим установившееся состояние  $g_{ss}(x)$ . Для этого примем  $\dot{g}(x) = 0$  из уравнения (10) и выразим  $g(x)$

$$g_{ss}(x) = -[A(x) - B(x)R^{-1}(x)B'(x)P_c(x)]^{-1} \cdot C'(x)Q(x)v(t). \quad (19)$$

8. Используем технологию замены переменных и примем, что

$$K_g(x) = [g(x) - g_{ss}(x)]. \quad (20)$$

9. Используем значение  $g(x, t_f)$ , рассчитанное по формуле (11), и вычислим

$$K_g(x, t) = \exp(-(A(x) - B(x)R^{-1}(x)B'(x)P_c(x))(t - t_f)) \cdot (g(x, t_f) - g_{ss}(x)). \quad (21)$$

10. Вычислим  $g(x)$  из уравнения (20)

$$g(x) = K_g(x) + g_{ss}(x). \quad (22)$$

11. Теперь имеются все данные для вычисления закона управления. Остается найти его по формуле (6).

Рассмотренные шаги можно графически представить с помощью рисунка 1.

В [13] указано, что если  $t_0 \ll t_f$ , то решение ДУР (7) сходится к решению алгебраического уравнения Риккати (12) и матрица  $K(x)$ , которая является инверсией выражения  $P_c(x) - P_{ss}(x)$ , может стать сингулярной. Избежать проблемы сингулярности можно с помощью вычисления отрицательно определенного решения уравнения (12). Отметим, что в этом случае выражение  $P_c(x) - P_{ss}(x)$  гарантированно будет положительно определенным, следовательно, существует его инверсия [13].

Для вычисления отрицательно определенного решения алгебраического уравнения Риккати (12) достаточно изменить знак матрицы  $A(x)$  на противоположный текущему и решить алгебраическое уравнение Риккати (12). Теперь отрицательно определенное решение уравнения (12) будет равно  $-P_{ss}(x)$  [6]. Условие (3) необходимо изменить так, чтобы пара  $\{-A(x), B(x)\}$  вместо пары  $\{A(x), B(x)\}$  является поточно управляемой или стабилизируемой для  $\forall x$ .

Нелинейные системы с ЗСУР управлением – локально асимптотически устойчивые, что доказано в работах Банка и Клутье [4, 5].

Метод ЗСУР для конечного горизонта времени может быть использован для линейных систем.

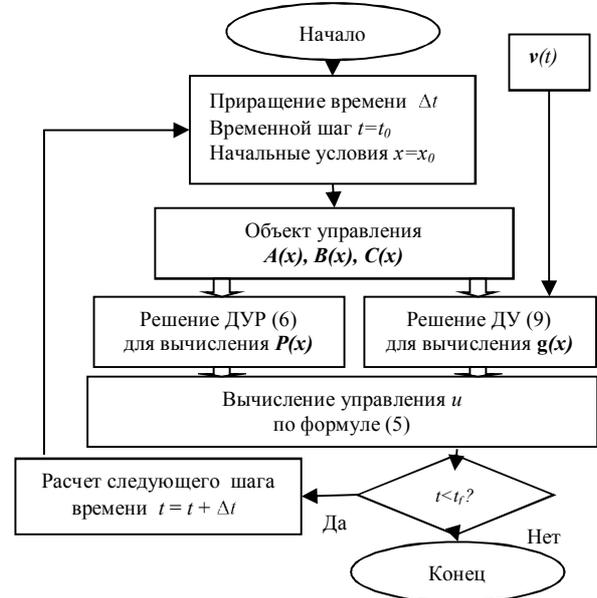


Рисунок 1 – Решение задачи слежения с помощью метода ЗСУР для конечного горизонта времени

### Методология проектирования наблюдателя ЗСУР для конечного горизонта времени.

Проблема проектирования наблюдателя возникает, когда появляется необходимость во внутренней информации о системе при доступных внешних измерениях. Использование датчиков для получения информации сталкивается с различными сложностями: стоимость, технологические ограничения [14].

Наблюдатель ЗСУР для конечного горизонта времени основывается на решении двойственной (дуальной) задачи, основанной на технологии построения нелинейных регуляторов. При этом наблюдатель имеет такую же структуру, как расширенный наблюдатель Калмана (РНК) [15]. В отличие от РНК, который использует якобиан нелинейностей в динамике системы, наблюдатель ЗСУР для конечного горизонта времени основан на расширенной линеаризации, которая приводит нелинейную систему к линейной структуре с зависящими от состояния параметрами [14]. Этот метод создает дополнительные степени свободы, которые можно использовать, чтобы преодолеть ограничения традиционных методов оценивания вектора состояния, такие как низкая эффективность, сингулярность и потеря свойства наблюдаемости [16].

Рассмотрим расширенную линеаризацию модели нелинейной системы (2 – 4). Дифференциальное уравнение для оценки вектора состо-

яния определяется как

$$\dot{\hat{x}} = A(\hat{x})\hat{x} + B(\hat{x})u(\hat{x}) + \Psi \left\{ \hat{x}, [C(x)x - C(\hat{x})\hat{x}] \right\}, \quad (23)$$

где  $\Psi$  – дополнительное слагаемое наблюдателя, благодаря которому вектор оценок состояния  $\hat{x} \rightarrow x$  при  $t \rightarrow t_f$ . Пусть ошибка оценивания для системы определяется, как  $e_{oc.} = x - \hat{x}$  и  $u$  – управление.

Пара  $\{A(x), C(x)\}$  должна быть поточечно наблюдаемой для  $\forall x$ .

Наблюдатель ЗСУР для конечного горизонта времени с уравнением (23) является субоптимальным наблюдателем, так как приближенно минимизирует функционал стоимости

$$J(\hat{x}, u) = \frac{1}{2} (e_{oc.}(t_f))' \bar{F} e_{oc.}(t_f) + \int_{t_0}^{t_f} [e_{oc.}'(t) \bar{Q}(\hat{x}) e_{oc.}(t) + u'(\hat{x}) \bar{R}(\hat{x}) u(\hat{x})], \quad (24)$$

где  $\hat{x} \in R^n$ ,  $u \in R^m$  и  $\bar{Q} \in R^{n \times n}$  – симметричная положительно полуопределенная матрица, а  $\bar{R} \in R^{m \times m}$  – симметричная положительно определенная матрица. Предлагаемый в статье нелинейный наблюдатель строится, имитируя наблюдатель для линейных систем.

Чтобы найти субоптимальный наблюдатель, минимизируем функционал стоимости (24), решая уравнение

$$\begin{aligned} -\dot{P}_c(\hat{x}) &= P_c(\hat{x})A'(\hat{x}) + A(\hat{x})P_c(\hat{x}) - \\ &- P_c(\hat{x})C'(\hat{x})\bar{R}^{-1}(\hat{x})C(\hat{x})P_c(\hat{x}) + \bar{Q}(\hat{x}). \end{aligned} \quad (25)$$

Конечное значение решения уравнения (25) определяется как

$$P_c(\hat{x}(t_f)) = B(\hat{x}(t_f))\bar{F}B'(\hat{x}(t_f)). \quad (26)$$

Для того чтобы решить уравнение (25), необходимо пройти следующие шаги.

1. Решить алгебраическое уравнение Риккати

$$\begin{aligned} P_{ss}(\hat{x})A'(\hat{x}) + A(\hat{x})P_{ss}(\hat{x}) - \\ - P_{ss}(\hat{x})C'(\hat{x})\bar{R}^{-1}(\hat{x})C(\hat{x})P_{ss}(\hat{x}) + \bar{Q}(\hat{x}) = 0, \end{aligned} \quad (27)$$

для того чтобы вычислить значение устойчивого состояния  $P_{ss}(\hat{x})$ .

2. Рассчитать значение матрицы замкнутой системы

$$A_{cl}(\hat{x}) = A'(\hat{x}) - C'(\hat{x})\bar{R}^{-1}(\hat{x})C(\hat{x})P_{ss}(\hat{x}). \quad (28)$$

3. Решить алгебраическое уравнение Ляпунова относительно  $D(\hat{x})$

$$\begin{aligned} A_{cl}(\hat{x})D(\hat{x}) + D(\hat{x})A'_{cl}(\hat{x}) - \\ - C'(\hat{x})R^{-1}(\hat{x})C(\hat{x}) = 0. \end{aligned} \quad (29)$$

4. Используя технику замены переменных, предположим, что

$$K(\hat{x}) = [P_c(\hat{x}) - P_{ss}(\hat{x})]^{-1}. \quad (30)$$

5. Найти  $K(\hat{x})$  с помощью дифференциального уравнения Ляпунова

$$\begin{aligned} \dot{K}(\hat{x}) &= K(\hat{x})A'_{cl}(\hat{x}) + A_{cl}(\hat{x})K(\hat{x}) - \\ &- C'(\hat{x})R^{-1}(\hat{x})C(\hat{x}). \end{aligned} \quad (31)$$

Решение уравнения (31) имеет следующий вид

$$\begin{aligned} K(\hat{x}, t) &= \exp(A_{cl}(\hat{x})(t - t_f)) \cdot (K(\hat{x}, t_f) - \\ &- D(\hat{x})) \cdot \exp(A'_{cl}(\hat{x})(t - t_f)) + D(\hat{x}), \end{aligned} \quad (32)$$

где  $K(\hat{x}, t_f) = [P_c(\hat{x}, t_f) - P_{ss}(\hat{x})]^{-1}$ . Для вычисления  $K(\hat{x}, t_f)$  необходимо рассчитать  $P_c(\hat{x}, t_f)$  по формуле (26).

6. Вычислить  $P_c(\hat{x})$  из уравнения (30) как

$$P_c(\hat{x}) = K^{-1}(\hat{x}) + P_{ss}(\hat{x}). \quad (33)$$

Во избежание проблем с сингулярностью вычисляется отрицательно определенное решение уравнения (27). Тогда пара  $\{-A(x), C(x)\}$  должна быть поточечно наблюдаемой для  $\forall x$ .

Далее выбираем дополнительное слагаемое наблюдателя  $\Psi$

$$\begin{aligned} \Psi \left\{ \hat{x}, [C(x)x - C(\hat{x})\hat{x}] \right\} &= \\ &= L(\hat{x})(C(x)x - C(\hat{x})\hat{x}), \end{aligned} \quad (34)$$

где

$$L(\hat{x}) = P_c(\hat{x})C'(\hat{x})\bar{R}^{-1} \quad (35)$$

является параметром наблюдателя.

Матрицы  $\bar{Q}$ ,  $\bar{R}$  и  $\bar{F}$  и позволяют настраивать характеристики наблюдателя. Часто эти матрицы выбираются равными матрицам  $Q$ ,  $R$  и  $F$ . После всех преобразований получаем следующее уравнение наблюдателя ЗСУР для конечного горизонта времени

$$\dot{\hat{x}} = A(\hat{x})\hat{x} + B(\hat{x})u + L(\hat{x})[C(x)x - C(\hat{x})\hat{x}]. \quad (36)$$

Анализируя рассмотренные уравнения, можно сделать вывод, что задача построения наблюдателя ЗСУР для конечного горизонта времени дуальна, другими словами, эквивалентна задаче проектирования ЗСУР регулятора для конечного горизонта времени.

**Описание работы инструментария.** Для разработки инструментария с «дружелюбным» графическим интерфейсом используется специализированная среда GUIDE, входящая в состав системы MATLAB [17]. Проектировщик осу-

ществляет работу с инструментарием через два окна, созданных средствами среды GUIDE.

Главное окно, с помощью которого синтези-

руется ЗСУР регулятор, решающий задачу слежения для конечного горизонта времени (рисунок 2).

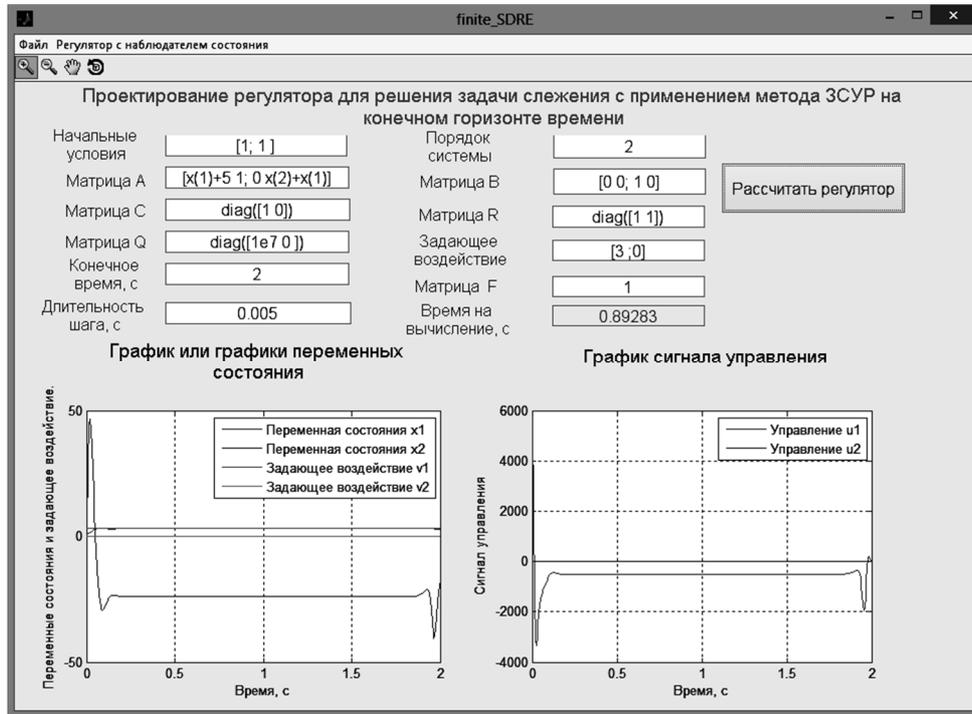


Рисунок 2 – Главное окно инструментария

Все, что требуется для работы – это ввести в текстовые поля матрицы модели системы, весовые матрицы, начальные условия, порядок системы, конечное время моделирования, длительность шага моделирования и задающее воздействие. Ввод осуществляется с помощью текстовых полей для ввода. Если хотя бы одно из полей для ввода заполнено неверно, например, неверная размерность матриц, отрицательные значения времени моделирования, то при нажатии на кнопку «Рассчитать регулятор» выдается ошибка с соответствующим описанием и расчет регулятора не производится.

В текстовых полях для ввода начального состояния, порядка системы, конечного времени моделирования и длительности шага моделирования вбиваются только числа, а в остальных полях для ввода наряду с числами используются символы, чтобы ввести элементы вектора состояния или функцию, зависящую от времени  $t$ . Для иллюстрации поведения вектора состояния и управления на оси выводятся их графики. В неактивное текстовое поле выводится время, затраченное на вычисление алгоритма.

В главном окне присутствует панель инструментов. Все элементы панели инструментов предназначены для работы с графиками: лупа для увеличения масштаба графиков, лупа для уменьшения масштаба графиков, панорамирова-

ние, используемое для перемещения области просмотра [17].

После запуска инструментария текстовые поля ввода уже заполнены для демонстрационного примера, чтобы пользователь разобрался с правилами ввода, и чтобы продемонстрировать работоспособность инструментария. Графики изменения вектора состояния и управления имеют подписи осей координат. У всех графиков есть легенда, так как на них может присутствовать большое количество элементов и без легенды будет не понятно, где на графике изображен тот или иной элемент.

В главном окне реализовано меню с вкладками «Файл» и «Регулятор с наблюдателем состояния». С помощью подпунктов вкладки «Файл» можно сохранить изображение окна инструментария в виде bmp-файла, сохранить в txt-файл данные, находящиеся в текстовых полях ввода, а также есть возможность загрузить сохраненные ранее данные из txt-файла, что значительно упрощает работу с инструментарием, так как отпадает необходимость вбивать заново уже сохраненный проект, и есть возможность закрыть инструментарий.

Вспомогательное окно для синтеза ЗСУР регулятора, решающего задачу слежения с применением наблюдателя ЗСУР для конечного горизонта времени, которое запускается при нажатии

на вкладку меню «Регулятор с наблюдателем состояния» (рисунок 3).

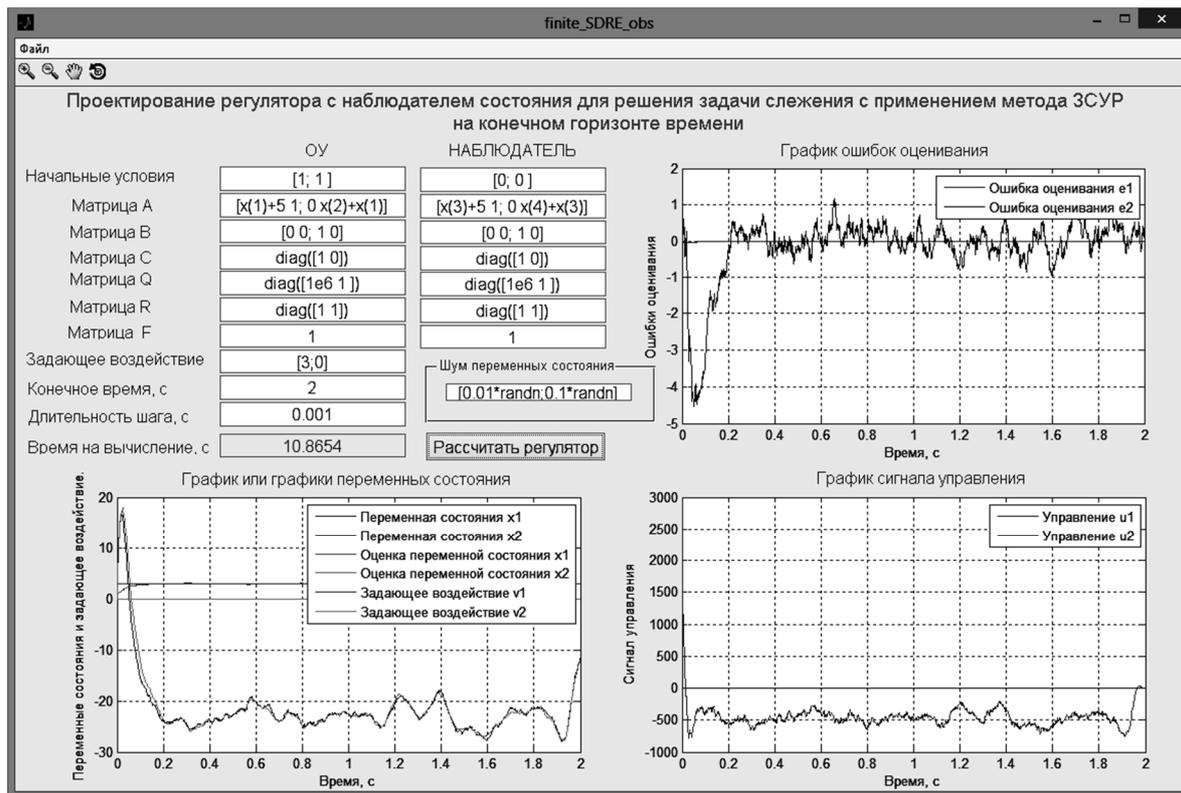


Рисунок 3 – Вспомогательное окно для синтеза регулятора с применением наблюдателя

Наблюдатель состояния ЗСУР используется при проектировании ЗСУР управления, так как для вычисления закона ЗСУР управления необходимо знать вектор состояния, который не всегда может быть измерен с помощью датчиков (использование датчиков не всегда является целесообразным). После запуска открывается новое окно, на котором расположены одна кнопка для синтеза регулятора с наблюдателем, поля для ввода начальных условий, матриц  $A$ ,  $B$ ,  $C$ ,  $Q$ ,  $R$ ,  $F$  ОУ и наблюдателя, поля для ввода конечного времени моделирования, длительности шага моделирования, вектора задающего воздействия и вектора шума. Если хотя бы одно из полей для ввода заполнено неверно, то при нажатии на кнопку «Рассчитать регулятор» выдается ошибка с соответствующим описанием и расчет регулятора не производится.

Также присутствуют три оси, на которые выводятся графики изменения вектора состояния и вектора оценок состояния, графики изменения ошибок оценивания и график изменения вектора управления. Все графики имеют подписи осей координат. У всех графиков есть легенда. В неактивное текстовое поле выводится время, затраченное на вычисление алгоритма.

Текстовые поля ввода уже заполнены для демонстрационного примера.

На вспомогательном окне расположена па-

нель инструментов. Все элементы панели аналогичны элементам панели инструментов для основного окна.

Реализовано меню с вкладкой «Файл», подпункты которой аналогичны подпунктам одноименной вкладки меню главного окна.

**Пример для иллюстрации работы инструментария.** Для иллюстрации работоспособности разработанного инструментария используем демонстрационный пример.

В нем модель ОУ описывается ДУ:

$$\begin{aligned} \dot{x}_1 &= x_1^2 + 5x_1 + x_2, \\ \dot{x}_2 &= x_2^2 + x_1x_2 + u, \\ x_1(0) &= x_2(0) = 1. \end{aligned} \quad (37)$$

- синтез ЗСУР регулятора.

Весовые матрицы представлены следующим образом:

$$Q = \begin{bmatrix} 1000000 & 0 \\ 0 & 0 \end{bmatrix}, \quad R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad F = 1.$$

Задающее воздействие  $v = [3; 0]$ . Конечное время равно 2 с. Длительность шага 0.005 с, что позволяет добиться работы системы управления в режиме реального времени (время, затраченное на вычисление алгоритма, приближенно равно 0.89 с). Матрица выхода

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},$$

поэтому лишь  $x_1 = v_1$ .

Из графиков на рисунке 1 видно, что  $x_1 = v_1 = 3$  после 0.1 с. Ошибка слежения равна в установившемся режиме 0.008.

- Синтез ЗСУР регулятора с применением наблюдателя.

Весовые матрицы представлены следующим образом:

$$Q = \begin{bmatrix} 1000000 & 0 \\ 0 & 1 \end{bmatrix}, R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, F = 1.$$

Задающее воздействие  $v = [3; 0]$ . Конечное время равно 2 с. Длительность шага 0.001 с. Увеличение шага приводит к тому, что алгоритм не сходится. С таким шагом время, затраченное на вычисление алгоритма, приближенно равно 15.87 с. Матрица выхода

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},$$

поэтому лишь  $x_1 = v_1$ .

Из графиков на рисунке 2 видно, что  $x_1 \approx v_1 = 3$  после 0.15 с. Максимальная ошибка слежения равна 0.4 из-за наличия шума. На выход наложен шум с нулевым матожиданием и дисперсией 0.1. На переменную состояния  $x_2$  наложен шум с нулевым математическим ожиданием и дисперсией 0.32.

Из рисунка 2 видно, что несмотря на шум ошибка оценивания переменной состояния  $x_1$  стремится к нулю, а ошибка оценивания переменной состояния  $x_2$  колеблется от -1.1 до 1.1, ее пиковое значение 4.5 (это обусловлено выбором матрицы  $Q$ ).

Из результатов симуляций для демонстрационного примера можно сделать вывод, что инструментарий успешно производит синтез регуляторов.

**Выводы.** Испытания инструментария доказали его работоспособность.

Стоит отметить, что разработано работоспособное универсальное приложение для синтеза ЗСУР регуляторов, решающих задачу слежения на конечном горизонте времени, с простым и понятным пользователю интерфейсом. Синтез регулятора осуществляется для линейных и нелинейных аффинных по входу систем. Предложенный в статье инструментарий позволяет значительно упростить работу проектировщика.

#### Библиографический список

1. Бобиков А.И. Анализ и проектирование нелинейных систем управления. РГРТУ. Рязань, 2013. – 220 с.

2. Çimen T. Survey of State-Dependent Riccati Equation in Nonlinear Optimal Feedback Control Synthesis. AIAA Journal of Guidance, Control, and Dynamics, Vol. 35, No. 4, 2012. P. 1025–1047.

3. Erdem E.B. Analysis and real-time implementation of state-dependent riccati equation controlled systems, Thesis for the degree of Doctor of Philosophy University of Illinois at Urbana-Champaign, 2001.

4. Banks S.P. and Manha K.J. Optimal Control and Stabilization for Nonlinear Systems, IMA Journal of Mathematical Control and Information, vol. 9. P. 179-196, 1992.

5. Cloutier J.R. State-dependent Riccati equation techniques: an overview, Proceedings of the American Control Conference, vol.2. P. 932-936, 1997.

6. Khamis A. and Naidu D S. Nonlinear optimal tracking with incomplete state information using finite-horizon state dependent Riccati equation (SDRE), American Control Conference (ACC), 2014.

7. Nguyen, T., and Gajic, Z. Solving the matrix differential Riccati equation: a Lyapunov equation approach, IEEE Trans. Automatic Control, Vol. 55, No. 1, 2010. P. 191-194.

8. Heydari A., Balakrishnan S.N. Optimal online path planning for approach and landing guidance, AIAA Atmospheric Flight Mechanics Conference 08 - 11 August, 2011.

9. Nazarzadeh, J., Razzaghi, M., and Nikravesh, K. Solution of the matrix Riccati equation for the linear quadratic control problems, Mathematical and computer modelling, Vol. 27, No. 7. 1998. P. 51-55.

10. Бобиков А.И., Грушин Д.М. Инструментарий для синтеза нелинейного регулятора, основанного на решении зависящего от состояния уравнения Риккати в среде GUI системы MATLAB // Вестник Рязанского государственного радиотехнического университета. 2014. № 50-1. С. 22–28.

11. Çimen T. Systematic and effective design of nonlinear feedback controllers via the State-dependent Riccati equation (SDRE) method, Annual Reviews in Control. 2010. P. 32–51.

12. Gajic Z. and Qureshi M. The Lyapunov matrix equation in system stability and control. New York: Dover Publications, 2008.

13. Heydari A., Balakrishnan S.N. Closed-form solution to finite-horizon suboptimal control of nonlinear systems, Int. J. Robust Nonlinear Control, 2014.

14. Jaganath C., Ridley A. and Bernstein D.S. A SDRE-based asymptotic observer for nonlinear discrete-time systems, American Control Conference. 2005. P. 3630-3635.

15. Haessig D., Friedland B. State dependent differential Riccati equation for nonlinear estimation and control, Proceedings of the 15th World Congress The International Federation of Automatic Control Barcelona, Spain. 202. P. 227-232.

16. Mracek C.P., Cloutier J.R. and D'Souza C.A. A new technique for nonlinear estimation, Proceedings of the International Conference on Nonlinear Problems in Aviation and Aerospace. 1996. P. 387–396.

17. Дьяконов В. П. MATLAB 7.\*/R2006/R2007. Самоучитель. Программные средства математических

вычислений. Операции с векторами и матрицами. Использование обычной и специальной графики. Об-

зор расширений в MATLAB. Визуальное программирование GUI. – М.: ДМК-Пресс. 2008. — 768 с.

УДК 681.515

*А.И. Бобиков, И.И. Сурков*

## НЕЙРОСЕТЕВОЕ УПРАВЛЕНИЕ СКОРОСТЬЮ ДВИГАТЕЛЯ ПОСТОЯННОГО ТОКА

*Предложен новый алгоритм управления скоростью двигателя постоянного тока с использованием нейросетевого регулятора, который позволяет исключить влияние нелинейностей двигателя на его выходной сигнал. В качестве примера рассмотрена нелинейная модель двигателя постоянного тока, построенная с помощью инструмента Simulink в пакете прикладных программ MATLAB. Показано преимущество нового метода настройки регулятора над классическими методами.*

**Ключевые слова:** *двигатель постоянного тока, нелинейные параметры, ПИД-регулятор, адаптивный регулятор, нейросетевой регулятор.*

**Введение.** Двигатели постоянного тока (ДПТ) используются для обеспечения вращательного или прямолинейного движения в различных электромеханических устройствах и следящих системах [1]. Они находят своё применение во многих отраслях производства (робототехника [2], станки низкой и средней мощности [3, 4] и т.д.).

Двигатель постоянного тока – это один из основных компонентов системы управления. У любой системы управления должен быть модуль привода, который фактически обеспечивает выполнение возложенных на неё функций [5].

Однако, несмотря на интенсивные разработки в области ДПТ, всё ещё не удаётся избавиться от присущих им нелинейностей. ДПТ до сих пор обладают нелинейными свойствами и имеют параметры, влияющие на их качество и требующие разработки более сложных систем управления или в ряде случаев являющиеся причиной потребления большого количества энергии и возникновения неисправностей.

Идея разработки регулятора для борьбы с нелинейностями ДПТ не нова. Например, в [6] разработана нелинейная модель ДПТ, однако решение проблемы ослабления влияния нелинейностей осуществляется путём введения регулятора в токовую цепь двигателя, что представляет трудности с точки зрения реализации.

Также имеется ряд работ, посвященных моделированию нелинейных систем управления двигателем постоянного тока [7, 8, 9]. В них рассматриваются основные, наиболее распространённые виды нелинейностей ДПТ.

Первым важным нелинейным свойством является зона нечувствительности, из-за которой двигатель не начнет вращаться, пока входное напряжение не достигнет определенного минимального значения. Это замедляет реакцию системы и требует большего сигнала управления для достижения требуемой скорости [18].

Второе важное нелинейное свойство двигателей постоянного тока – эффект насыщения, при котором выходной сигнал двигателя не может достигнуть требуемого значения. Например, когда мы хотим достигнуть угловой скорости, равной 100 об/мин, подавая входное напряжение, равное 12 вольт, то двигатель достигает только 90 об/мин при этом напряжении. Эффект насыщения очень распространен почти во всех ДПТ.

Третьим математическим описанием нелинейного эффекта, имеющего место в двигателях постоянного тока, является воздушный зазор (люфт) в редукторах двигателя. Некоторые ДПТ используют внутренние соединения, осуществляемые с помощью редукторов, чтобы улучшить свои механические и скоростные характеристики, однако это улучшение оказывает негативный эффект на характеристики выходной скорости [9].

*Цель работы* – спроектировать нейросетевой регулятор, который будет способен устранить или ослабить влияние перечисленных нелинейностей на выходной сигнал двигателя постоянного тока.

**Теоретические сведения о ДПТ.** Двигатель постоянного тока имеет два основных компо-

нента. Первый – электрический компонент, состоящий из сопротивления  $R$ , индуктивности  $L$ , входного напряжения  $V$  и противоэлектродвижущей силы  $E$ . Второй – механический компонент, благодаря которому создается полезное механическое вращательное движение на валу. А именно – вал привода, момент инерции привода и инерции нагрузки  $J$  и затухание  $b$ .

На рисунке 1 изображена типичная модель ДПТ. Через  $\omega$  обозначена угловая скорость вала.

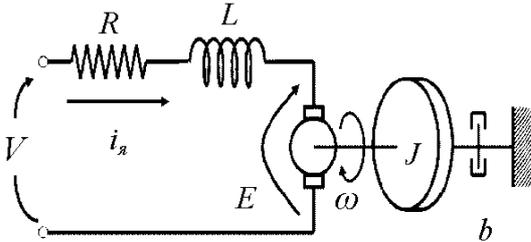


Рисунок 1 – Принципиальная схема ДПТ

**Построение модели ДПТ.** С помощью законов Ньютона и Кирхгофа и рисунка 1 получаем следующие уравнения математической модели ДПТ:

$$V = Ri_a + L \frac{di_a}{dt} + E, \tag{1}$$

$$E = K_e \omega, \tag{2}$$

$$T_m = K_t i_a, \tag{3}$$

$$T_m = J \dot{\omega} + b\omega + T_n, \tag{4}$$

- где  $R$  – сопротивление;
- $L$  – индуктивность;
- $V$  – приложенное напряжение;
- $E$  – противо-ЭДС;
- $K_t$  – постоянная вращающего момента;
- $J$  – момент инерции;
- $b$  – коэффициент демпфирования;
- $i_a$  – ток якоря;
- $K_e$  – постоянная электродвижущей силы;
- $T_m$  – вращающий момент;
- $\omega$  – скорость двигателя;
- $T_n$  – момент нагрузки.

Применяя преобразование Лапласа к уравнениям (1) – (4), получаем передаточную функцию, связывающую входное напряжение  $V$  и угловую скорость двигателя  $\omega$ :

$$\frac{\omega(s)}{V(s)} = \frac{k}{(R + Ls)(J + b) + k^2}. \tag{5}$$

Здесь  $k = K_t = K_e$ .

На рисунке 2 показана структурная схема, описывающая ДПТ.

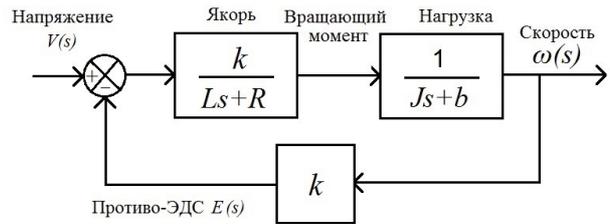


Рисунок 2 – Структурная схема ДПТ

С использованием передаточной функции ДПТ (5) была построена модель ДПТ в Simulink (рисунок 3). Также в нее были добавлены нелинейности, присущие двигателю постоянного тока и рассмотренные в работе [6]. Значение зоны нечувствительности составляет 1.5 В, верхняя граница эффекта насыщения равняется 0.5 В, нижняя – -0.5 В, параметр люфта равен 1.

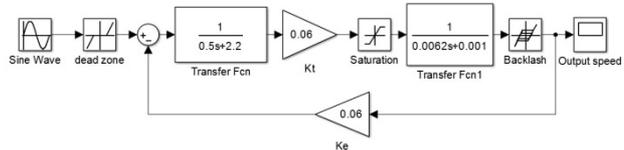


Рисунок 3 – Модель ДПТ в Simulink

Результат динамического моделирования ДПТ в течение 10 секунд с фиксированным шагом, равным 0.02 с, представлен на рисунках 4 – 6. Значения параметров, используемых в модели Simulink, приведены в таблице 1.

Таблица 1– Значения параметров модели ДПТ

Параметр	Значение
$J$	0.0062 Н·м·с <sup>2</sup> /рад
$b$	0.001 Н·м·с/рад
$K_t$	0.06 Н·м/А
$K_e$	0.06 В·с/рад
$R$	2.2 Ом
$L$	0.5 Гн

Рисунок 4 показывает влияние нелинейностей на реакцию двигателя постоянного тока при входном гармоническом сигнале с амплитудой 12 В и частотой 2.6 рад/с. Как видим, имеют место большие отклонения сигнала действительной скорости от сигнала желаемой скорости.

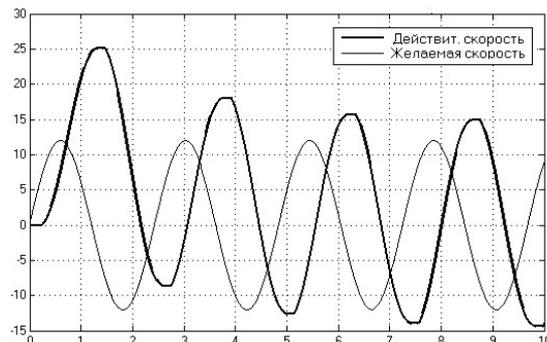


Рисунок 4 – Действительная и желаемая скорость

Чтобы получить более четкое представление о влиянии нелинейных элементов, которые содержит ДПТ, был также использован ступенчатый входной сигнал с максимальным значением, равным 12 вольт, что является максимальным напряжением для используемого ДПТ. На рисунке 5 видно, как элемент типа зона нечувствительности ограничивает сигнал желаемой скорости до 10.5 вольт. Вследствие этого наблюдается более медленная реакция ДПТ, представленная сигналом действительной скорости (рисунок 6).

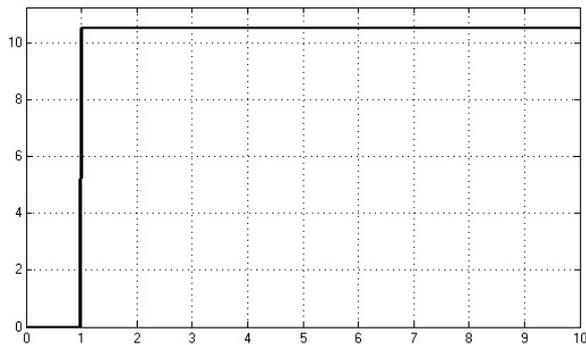


Рисунок 5 – Влияние элемента типа зона нечувствительности на желаемую скорость

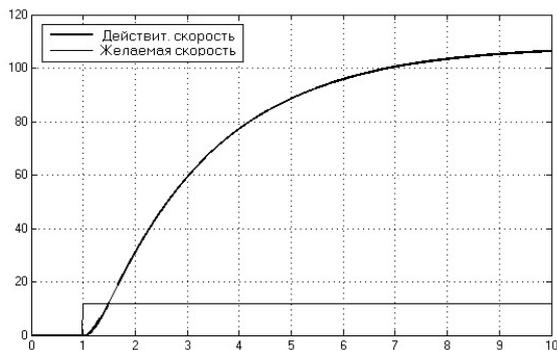


Рисунок 6 – Действительная и желаемая скорость

**Проектирование ПИД-регулятора.** Для расчета параметров ПИД-регулятора с законом управления (6) воспользуемся первым методом Циглера-Николса [13]. Популярность этого метода обусловлена отсутствием сложных математических расчётов и тем, что адаптивные контроллеры зарубежных и отечественных производителей, как правило, реализуют метод Циглера-Николса [14]. В данном методе используются всего два параметра:  $L$  и  $T$ , определяемые по переходной характеристике объекта управления, как показано на рисунке 7. Формулы для расчета коэффициентов ПИД-регулятора приведены в таблице 2.

Имеем следующий закон управления:

$$u(t) = K_p \left( 1 + \frac{1}{T_i} \int e d\tau + T_d \frac{de}{dt} \right), \quad (6)$$

где  $K_p$  – пропорциональный коэффициент усиления,  $T_i$  – постоянная интегрирования,  $T_d$  – постоянная дифференцирования.

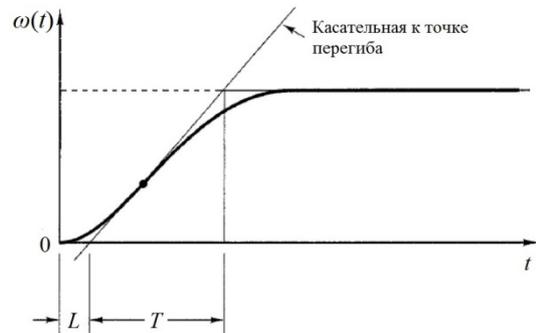


Рисунок 7 – Переходная характеристика ДПТ

Таблица 2 – Формулы для расчета коэффициентов регулятора по методу Циглера-Николса

$K_p$	$T_i$	$T_d$
$1.2 \frac{T}{L}$	$2L$	$0.5L$

Для рассматриваемой модели ДПТ были определены параметры  $L=0.166$ с и  $T=2.834$ с. Используя формулы из таблицы 2, определяем параметры ПИД-регулятора:  $K_p=20.486$ ,  $T_i=0.332$ ,  $T_d=0.083$ .

На рисунке 8 представлена модель системы управления ДПТ с ПИД-регулятором, реакция системы с обратной связью на гармонический и ступенчатый входной сигнал изображена соответственно на рисунках 9 и 10.

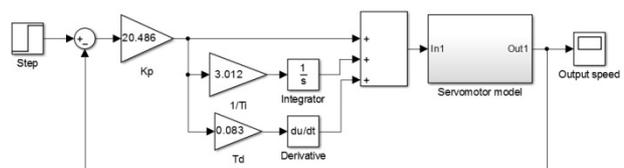


Рисунок 8 – Модель ДПТ с ПИД-регулятором

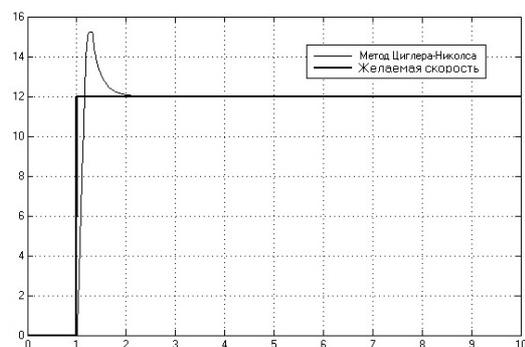
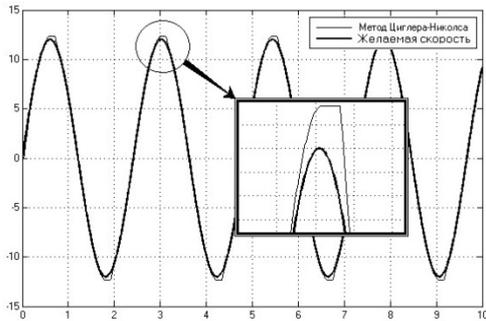


Рисунок 9 – Реакция системы при ступенчатом входном сигнале



**Рисунок 10 – Реакция системы при гармоническом входном сигнале**

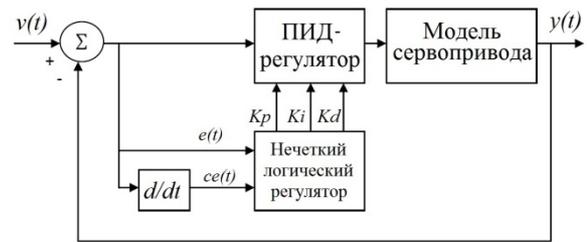
По рисункам 9, 10 отчетливо видно, что классический ПИД-регулятор не только не в состоянии справиться с нелинейностями ДПТ, но и обладает большим перерегулированием, что весьма нежелательно для рассматриваемой модели. Для повышения качества процесса управления имеет смысл прибегнуть к алгоритму нечеткой самонастройки ПИД-регулятора.

**Разработка нечеткого самонастраивающегося ПИД-регулятора.** В последние годы широкое распространение находят нечеткие модели и алгоритмы управления [15, 16]. Как известно, основой нечеткого управления является использование не столько аналитических или теоретических моделей, сколько практическое применение знаний квалифицированных специалистов, представленных в форме лингвистических баз правил [11]. Нечеткое управление эффективно в случаях неопределенности параметров объектов, когда используется опыт экспертов по управлению и настройке системы управления. Теория нечеткой логики позволяет применить знания специалистов – наладчиков с целью улучшения качества процессов управления и оказания помощи (супервизорный режим) по настройке типовых ПИД-регуляторов [12].

Принцип работы нечеткого самонастраивающегося ПИД-регулятора [17] прежде всего заключается в том, чтобы определить нечеткое отношение между тремя параметрами регулятора  $K_p$ ,  $K_i = 1/T_i$ ,  $K_d = K_p T_d$  и ошибкой управления  $e(t) = v(t) - y(t)$  и скоростью изменения ошибки  $ce(t) = de(t)/dt$ . Затем технология нечеткого логического вывода модифицирует три параметра с целью выполнить требования, предъявляемые к системе управления, причем делает это онлайн посредством постоянного оценивания ошибки и скорости её изменения. Такой регулятор имеет важное преимущество перед регулятором, настроенным по методу Циглера-Николса. При использовании обыкновенного ПИД-регулятора его основные параметры ( $K_p$ ,  $K_i$ ,  $K_d$ ) задаются как постоянные вели-

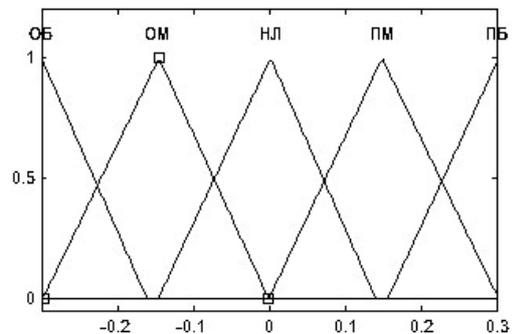
чины и настраиваются вручную для повышения качества процесса управления лишь путём многократной остановки работы системы. Напротив, нечёткий самонастраивающийся ПИД-регулятор автоматически изменяет свои параметры в процессе работы, основываясь на ошибке  $e$  и скорости её изменения  $ce$  [17].

Структурная схема нечеткого самонастраивающегося ПИД-регулятора представлена на рисунке 11.



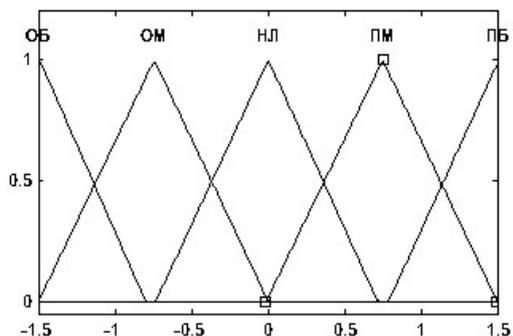
**Рисунок 11 – Структурная схема нечеткого самонастраивающегося ПИД-регулятора**

Для настройки нечеткого логического регулятора необходимо осуществить фаззификацию входных и выходных лингвистических переменных (ЛП). Фаззификация входных переменных (ошибки регулирования  $e$  и её производной  $ce$ ) производится согласно методике, описанной в работе [10]. Графики функций принадлежности входных ЛП представлены на рисунках 12 – 13.



**Рисунок 12 – Функции принадлежности ЛП «ошибка e»**

Здесь ОБ – отрицательная большая, ОМ – отрицательная малая, НЛ – нулевая, ПМ – положительная малая, ПБ – положительная большая.



**Рисунок 13 – Функции принадлежности ЛП «скорость изменения ошибки ce»**

Этап фаззификации выходных ЛП представлен на рисунке 14.

Функции принадлежности логических переменных  $K_i$  и  $K_d$  строятся аналогичным образом, с той лишь разницей, что ЛП  $K_i$  принадлежит универсуму [0 4], а ЛП  $K_d$  – универсуму [0 6].

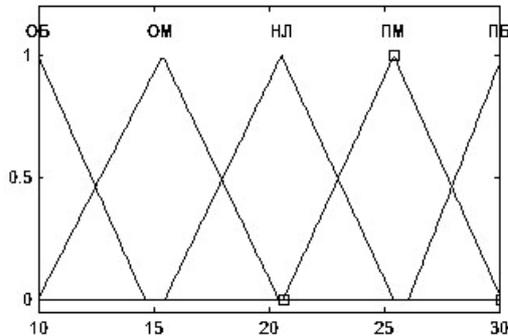


Рисунок 14 – Функции принадлежности ЛП «коэффициент усиления  $K_p$ »

Следующий шаг заключается в создании базы правил вида «ЕСЛИ ...ТО». Для простоты формулировки правила по каждой из трёх выходных логических переменных составлены с помощью таблицы 3.

Таблица 3– База правил для выходных ЛП

		$ce$				
		ОБ	ОМ	НЛ	ПМ	ПБ
$e$	ОБ	ПБ	ПБ	ПБ	ПМ	НЛ
	ОМ	ПБ	ПБ	ПМ	НЛ	НЛ
	НЛ	ПБ	ПМ	НЛ	ОМ	ОБ
	ПМ	НЛ	НЛ	ОМ	ОБ	ОБ
	ПБ	НЛ	ОМ	ОБ	ОБ	ОБ

Таким образом, получено 25 правил, первые 10 из которых представлены на рисунке 15 в окне редактора FIS программы Matlab.

1. If (e is ОБ) and (ce is ОБ) then (Kp is ПБ)(Ki is ПБ)(Kd is ПБ)
2. If (e is ОБ) and (ce is ОМ) then (Kp is ПБ)(Ki is ПБ)(Kd is ПБ)
3. If (e is ОБ) and (ce is НЛ) then (Kp is ПБ)(Ki is ПБ)(Kd is ПБ)
4. If (e is ОБ) and (ce is ПМ) then (Kp is ПМ)(Ki is ПМ)(Kd is ПМ)
5. If (e is ОБ) and (ce is ПБ) then (Kp is НЛ)(Ki is НЛ)(Kd is НЛ)
6. If (e is ОМ) and (ce is ОБ) then (Kp is ПБ)(Ki is ПБ)(Kd is ПБ)
7. If (e is ОМ) and (ce is ОМ) then (Kp is ПБ)(Ki is ПБ)(Kd is ПБ)
8. If (e is ОМ) and (ce is НЛ) then (Kp is ПМ)(Ki is ПМ)(Kd is ПМ)
9. If (e is ОМ) and (ce is ПМ) then (Kp is НЛ)(Ki is НЛ)(Kd is НЛ)
10. If (e is ОМ) and (ce is ПБ) then (Kp is НЛ)(Ki is НЛ)(Kd is НЛ)

Рисунок 15 – Окно редактора FIS

Схема модели ДПТ с нечетким самонастраивающимся ПИД-регулятором приведена на ри-

сунке 16. Реакции данной системы на ступенчатый и синусоидальный входной сигнал представлены соответственно на рисунках 17, 18.

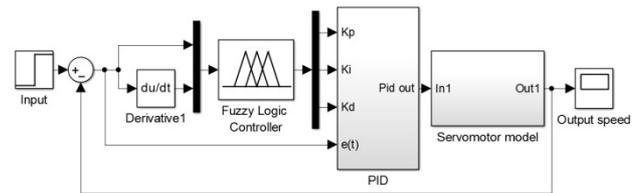


Рисунок 16 – Схема модели ДПТ в Simulink

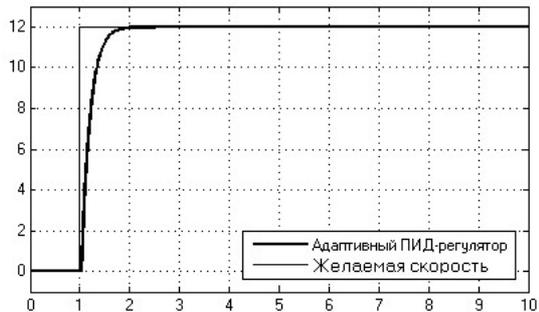


Рисунок 17 – Действительная и желаемая скорость системы с адаптивным нечетким регулятором для ступенчатого входного сигнала

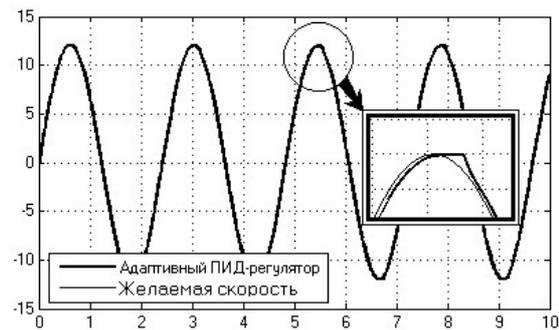


Рисунок 18 – Действительная и желаемая скорость системы с адаптивным нечетким регулятором для синусоидального входного сигнала

Если сравнить результат работы адаптивного регулятора с результатом работы ПИД-регулятора, настроенного по методу Циглера-Николса, невооруженным глазом видно, что нечеткий самонастраивающийся ПИД-регулятор гораздо лучше справляется с воспроизведением задающего воздействия.

Таким образом, использование адаптивного нечеткого ПИД-регулятора позволяет эффективно управлять технологическим процессом без его остановки, а дальнейшее переоборудование адаптивного нечеткого регулятора в нейросетевой регулятор приводит к самообучающейся адаптации по определению оптимальных настроек без корректировки базы правил.

**Переоборудование нечеткого самонастраивающегося ПИД-регулятора в нейросетевой регулятор.** Сам по себе вопрос проектирования нейросетевого регулятора является общеизвест-

ным. Существует целый ряд работ, в которых так или иначе затрагивается тема проектирования нейросетевого регулятора с целью улучшить качество процесса управления [6, 7, 8, 17]. Однако во всех этих работах используется один и тот же подход к обучению нейронной сети, который не всегда может обеспечить желаемую точность обучения. Суть этого подхода заключается в том, что нейронная сеть обучается на примере классического ПИД-регулятора, настроенного вручную для достижения желаемого качества процесса управления.

Для создания модели нейросетевого регулятора в Matlab был использован инструмент Neural Network Fitting Tool, который вызывается с помощью ввода в командную строку Matlab команды *nftool*. Данный инструмент позволяет создать и обучить нейронную сеть с использованием входных и выходных данных ПИД-регулятора, записанных из модели Simulink в рабочее пространство Matlab с помощью блоков *ToWorkspace*.

В качестве входных данных для обучения нейронной сети используется входной сигнал ПИД-регулятора, записанный в рабочее пространство Matlab в виде двумерной матрицы *pid\_in*, первый столбец которой представляет собой значение сигнала, а второй – значение времени. Аналогичным образом в рабочее пространство записывается выходной сигнал ПИД-регулятора в виде двумерной матрицы *pid\_out*, которая представляет собой целевые данные для процесса обучения. Далее осуществляется обучение нейронной сети с помощью инструмента Neural Network Fitting, число нейронов в скрытом слое равно 15. Процесс обучения занял 59 эпох, график регрессии представлен на рисунке 20.

Для дальнейшего использования спроектированной нейронной сети необходимо сохранить её в виде модели Simulink. Для этого в окне Deploy Solution инструмента Neural Network Fitting Tool был выбран вариант развёртывания нейронной сети как схемы Simulink. В результате была сгенерирована модель Simulink, представляющая собой спроектированный нейросетевой регулятор (рисунок 19).

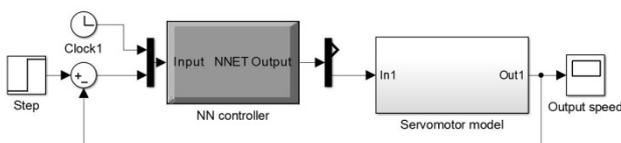


Рисунок 19 – Модель ДПТ с нейросетевым регулятором, обученным по традиционному методу

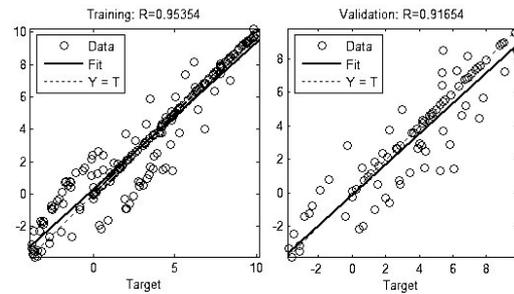


Рисунок 20 – График регрессии процесса обучения нейронной сети по традиционному методу

Реакция системы с нейросетевым регулятором, обученным по традиционному методу, на ступенчатый и синусоидальный входной сигнал изображена соответственно на рисунках 21, 22.

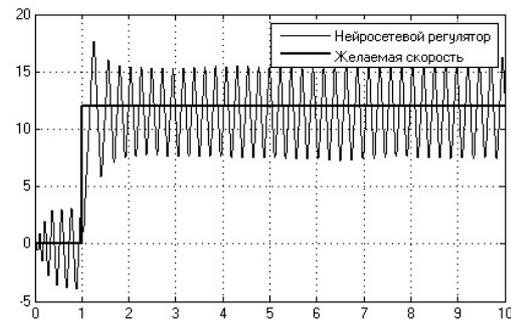


Рисунок 21 – Реакция системы с нейросетевым регулятором, обученным по традиционному методу на ступенчатый входной сигнал

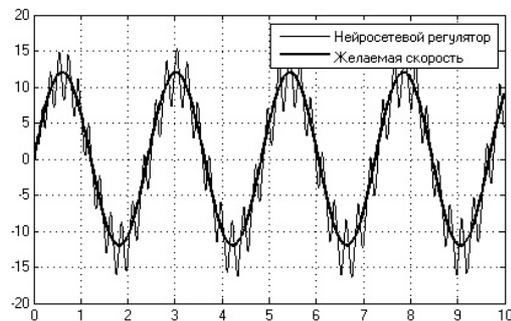


Рисунок 22 – Реакция системы с нейросетевым регулятором, обученным по традиционному методу на гармонический входной сигнал

Хорошо видно, что реакция системы управления с нейросетевым регулятором, обученным по традиционному методу, абсолютно не удовлетворяет предъявляемым к ней требованиям. Следовательно, такой регулятор не может быть использован для управления ДПТ с нелинейностями.

Однако нейросетевой регулятор по-прежнему обладает высоким потенциалом за счёт свойства обобщения нейронной сети. Поэтому был разработан новый метод обучения нейросетевого регулятора, предлагающий обучать нейросетевой регулятор на примере адаптивного нечеткого ПИД-регулятора. Входные

значения адаптивного нечеткого регулятора, а именно ошибка  $e$  и скорость изменения ошибки  $ce$ , были записаны соответственно в векторы  $E$  и  $CE$ , а затем объединены в двумерную матрицу  $input\_data$ . Управляющее воздействие с выхода адаптивного регулятора записано в вектор  $output\_data$ . Далее нейронная сеть была обучена с помощью инструмента Neural Network Fitting Tool, где входными данными была указана матрица  $input\_data$ , а целевыми данными – вектор  $output\_data$ , число нейронов в скрытом слое равно 15. Процесс обучения занял 118 эпох, график регрессии представлен на рисунке 23.

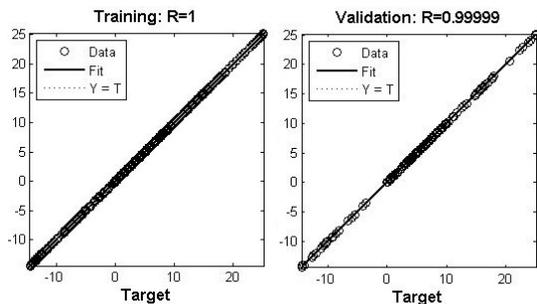


Рисунок 23 – График регрессии процесса обучения нейронной сети

По графику регрессии видно, что все данные расположены вдоль линии регрессии. Это является свидетельством успешного обучения.

Схема модели системы управления с нейросетевым регулятором представлена на рисунке 24. Реакция данной системы на ступенчатый и синусоидальный входной сигнал изображена соответственно на рисунках 25 – 26, по которым видно, что рассмотренный подход позволяет добиться высокой точности воспроизведения системой управления задающего воздействия и достаточно сильно ослабить влияние нелинейностей ДПТ.

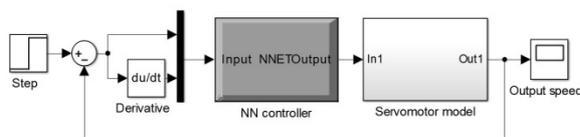


Рисунок 24 – Модель ДПТ с нейросетевым регулятором

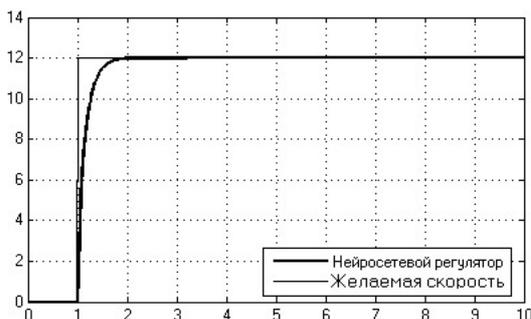


Рисунок 25 – Действительная и желаемая скорость системы с нейросетевым регулятором

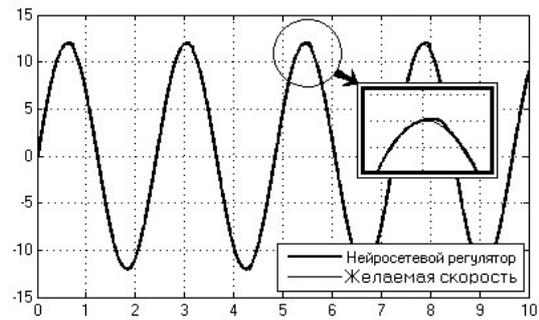


Рисунок 26 – Действительная и желаемая скорость системы с нейросетевым регулятором

Следует отметить, что рассматриваемый подход к обучению нейросетевому регулятору является инновационным и обеспечивает высокую точность обучения нейронной сети по сравнению с подходом, изложенным в работе [6]. Инновационность представленного подхода обусловлена выбором в качестве входных данных процесса обучения значений ошибки и скорости её изменения.

**Сравнительный анализ спроектированных регуляторов.** На рисунке 27 представлена реакция ДПТ на ступенчатый входной сигнал с каждым из спроектированных в данной работе регуляторов. Сравнительные характеристики переходного процесса приведены в таблице 4.

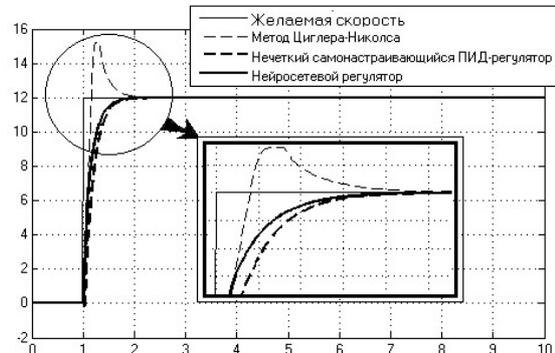


Рисунок 27 – Реакция ДПТ на ступенчатый входной сигнал для каждого из спроектированных регуляторов

Таблица 4– Характеристики переходного процесса

	Метод Циглера-Николса	Нечеткий самонастраивающийся ПИД-регулятор	Нейросетевой регулятор
$\sigma, \%$	26.61	0	0
$t_p, c$	2.05	1.94	1.87

Как видно из результатов моделирования, наилучшим вариантом регулятора для системы управления ДПТ с нелинейностями является нейросетевой регулятор, самым важным пре-

имуществом которого является способность свести к минимуму влияние этих нелинейностей на скорость ДПТ и, кроме того, обеспечить переходный процесс без перерегулирования.

**Заключение.** В статье предложен новый алгоритм нейросетевого управления нелинейной моделью двигателя постоянного тока. Показано, что с помощью интеллектуальных методов настройки регулятора можно устранить влияние нелинейных параметров на выходной сигнал системы. Примеры симуляции модели ДПТ демонстрируют эффективность предлагаемого метода настройки регулятора.

#### **Библиографический список**

1. Гусев Н.В. Алгоритмическое обеспечение систем управления следящими электроприводами // Научно-технический и учебно-образовательный журнал: Известия высших учебных заведений. Электромеханика. – 2006. № 3. – С. 57-60.
2. Конструкции промышленных роботов: учеб. пособие для СПТУ/ Е.М.Канаев, Ю.Г. Козырев, Б.И. Черпаков, В.И. Царенко. М.: Высш. шк., 1987. 95 с.
3. Босинзон М.А. Автоматизированные мехатронные модули линейных и вращательных перемещений металлообрабатывающих станков // Приводная техника. – 2002. № 1. – С. 10-19.
4. Демидов С.В., Авдушев С.А., Дубников А.М. и др. Электромеханические системы управления тяжелыми металлорежущими станками. – Л.: Машиностроение, 1986. – 236 с.
5. Шегал Г.Л. Электрические исполнительные механизмы в системах управления. – М.: Энергия, 1968. – 162 с.
6. E.E. Ezema, I.I. Eneh. Improving the Functional Limitations of DC Motor using Neural Network Controller // International Journal of Computer Technology and Electronics Engineering (IJCTEE), Volume 3, Issue 5, October 2013.
7. Lyshevski S.E. Nonlinear control of mechatronic systems with permanent-magnet DC motors // Mechatronics, Volume 9, Issue 5, August 1999.
8. Kara T., Eker I. Identification of Nonlinear Systems for Feedback Control. In: Proceedings of the Fourth GAP Engineering Congress 2002.
9. Nordin M., Gutman P. Controlling mechanical systems with backlash—a survey // Automatica, Volume 38, Issue 10, October 2002.
10. Бобиков А.И. Интеллектуальные системы управления (Проектирование нечетких ПИД-контроллеров и нечеткой обратной связи, нейронные сети). – Рязань: РГРТУ, 2008.
11. Бобиков А.И., Князева У.Ю., Лимонова А.С. Нечеткая следящая система с супервизорным управлением для моделей объектов типа Такаги-Сугено // Вестник Рязанского государственного радиотехнического университета. 2012. № 39-1. С. 63-70.
12. Бобиков А.И., Сивов С.А. Нечеткое супервизорное управление объектами первого порядка // Вестник Рязанского государственного радиотехнического университета. 2010. № 33. С. 97-103.
13. Денисенко В. ПИД-регуляторы: вопросы реализации // СТА. – 2008. № 1. – С. 86-99.
14. Михайленко В. С. Методы настройки нечёткого адаптивного ПИД-регулятора / В. С. Михайленко, В. Ф. Ложечников // ААЭКС. – 2009. – № 2 (24).
15. Леоненков А. Ю. Нечеткое моделирование в среде Matlab и fuzzyTech. – СПб.: БХВ, 2003. – 720 с.
16. Мелихов А.Н. и др. Ситуационные советующие системы с нечеткой логикой. – М.: Наука, 1990. – 272 с.
17. K.V.M. Krishna. Fuzzy PID Control for Networked Control System of DC Motor with Random Design // Acharya Nagarjuna University. International Journal of Computer Applications (0975 – 8887). – Volume 52. – № 7. August 2012.
18. Зазыкин В.Г., Чернышев А.П. О влиянии "зоны нечувствительности" в характеристике объекта управления на качество выполнения слежения // Вопросы психологии. 1980. – № 2. – С. 132–135.

УДК 004.896:621.865.8

**С.С. Лукаш**

## **АЛГОРИТМ ПОСТРОЕНИЯ КАРТЫ ОКРУЖАЮЩЕГО ПРОСТРАНСТВА МОБИЛЬНОЙ РОБОТОТЕХНИЧЕСКОЙ ПЛАТФОРМЫ В РЕАЛЬНОМ ВРЕМЕНИ**

*Рассматривается модификация итеративного алгоритма ближайших точек, разработанного для использования на графических процессорах и позволяющего осуществлять оценку перемещения в режиме реального времени. Проводится сравнительная характеристика временных показателей этапов итеративного алгоритма ближайших точек, реализованного для использования на центральном и графическом процессоре.*

**Ключевые слова:** лазерные сканеры, навигация, локализация, мобильный робот, цифровая карта, параллельные вычисления.

**Введение.** Функционирование мобильной робототехнической платформы (МРП) неразрывно связано с определением своего местоположения как локального, так и глобального, а также с получением информации об объектах пространства функционирования (ПФ). В настоящее время имеется множество различных методов определения относительного местоположения МРП: информация с одометрии, датчиков инерциальной навигационной системы (ИНС), глобальной спутниковой системы позиционирования (GNSS), датчиков магнитного поля Земли и др. Однако крайне важной является группа методов, основанная на оценке смещения объектов ПФ относительно МРП в ходе её перемещения. Эти методы используют различные типы датчиков, получающих информацию об объектах ПФ: видеосенсоры, в т.ч. и стереовидеосенсоры, лазерные сканеры, ультразвуковые дальномеры, сенсоры глубины (RGB-D камеры), а также радиолокационные методы. В большинстве случаев этих типов датчиков или их комбинаций достаточно для получения определённой информации об окружающей среде и оценке перемещения МРП в пространстве. Однако объём поступающих данных крайне велик и вычислительной мощности бортовых систем недостаточно для выполнения столь затратных вычислений в режиме реального времени.

Решением этой проблемы может быть организация распределённых вычислений, когда данные передаются с МРП на центральное вычислительное ядро и затем результат вычисления передаётся обратно. Однако сбои при передаче информации вследствие сложных условий окружающей среды или применения средств радиоэлектронной борьбы могут нарушить процесс функционирования МРП. Размещение мощного вычислительного модуля на самой МРП, как правило, невыгодно с точки зрения энергопотребления. Поэтому задача эффективной обработки данных с сенсоров МРП в реальном времени для обеспечения её автономного функционирования является крайне актуальной.

**Цель работы** – разработка алгоритма оценки перемещения МРП на основе данных с сенсоров, предоставляющих информацию об объектах ПФ, работающего в режиме реального времени.

**Теоретическая часть.** В области робототехнических систем принято выделять группу методов, которая носит название SLAM (Simultaneously Localization and Mapping – Одновременная навигация и картографирование). Суть методов состоит в получении актуальной информации о ПФ в различных представлениях

(главным образом, цифровая карта), осуществляя при этом оценку текущего положения МРП. Качество построения карт при этом зависит как от качества построения локальных сцен используемым сенсором, так и от точности оценки положения МРП, а, следовательно, и оценки её перемещения. Ввиду приемлемых характеристик современных сенсоров актуальным становится второй вопрос, который ориентируется на алгоритмы сопоставления частных данных.

Одним из наиболее распространённых алгоритмов для сопоставления частных сцен ПФ является итеративный алгоритм ближайших точек (*ICP – Iterative Closest Point*) [1]. Разработанный изначально для сопоставления частных сцен, получаемых со стационарного 3D сканера, этот алгоритм получил широкое распространение в робототехнике. Однако, имея итеративную природу, он достаточно ресурсоёмок, особенно при больших объёмах данных. Поэтому его использование в режиме реального времени крайне затруднительно, что приводит к необходимости другого подхода к его реализации.

В современных вычислительных системах широкое распространение получили графические процессоры, главной особенностью которых является наличие множества ядер, что позволяет выполнять операции с высокой степенью параллельности. Наибольшая эффективность достигается при применении одинакового набора инструкций на множество однотипных данных, что позволяет выполнять эти инструкции в различных потоках.

Таким образом, главная задача этой работы – выделение этапов алгоритма ICP и их распараллеливание для достижения такого времени вычисления, при котором алгоритм может использоваться в реальном времени.

Для получения данных об объектах ПФ, как было сказано выше, могут быть использованы различные сенсоры. В этой работе будут использованы данные с лидара Velodyne HDL-32E, который позволяет получать трёхмерную сцену окружающего пространства в виде массива точек с частотой до 10 Гц.

Итеративный алгоритм ближайших точек предложен в работах [2] и [3], а в дальнейшем подвергался ряду изменений, в результате чего возникли его различные варианты [4]. Пусть  $M$  и  $D$  – два массива точек в трёхмерном пространстве,  $m_i$  и  $d_i$  – пара соответствующих точек,  $m_i \in M$  и  $d_i \in D$ , а  $R$  и  $t$  – искомые трансформации вращения и перемещения. Задача алгоритма ICP заключается в минимизации весовой функции вида

$$E(R, t) = \frac{1}{N} \sum_{i=1}^N \|m_i - (Rd_i + t)\|^2, \quad (1)$$

где  $\|x\|$  – евклидово расстояние,  $N$  – число соответствующих пар точек.

Предлагается использовать концепцию алгоритма ICP, описанную в работе [6]. Можно выделить два ключевых этапа:

- нахождение ближайших точек на двух имеющихся сканах;
- нахождение оптимального преобразования  $(R, t)$ , при котором выражение (1) имеет минимальное значение.

Решение второго этапа может быть осуществлено с использованием сингулярного разложения [5]. Пусть  $\mu_M$  и  $\mu_D$  – центры масс массивов точек  $M$  и  $D$  соответственно, тогда

$$\mu_M = \frac{1}{N_M} \sum_{i=1}^{N_M} m_i, \quad (2)$$

$$\mu_D = \frac{1}{N_D} \sum_{i=1}^{N_D} d_i, \quad (3)$$

где  $N_M$  и  $N_D$  – количество точек в массивах  $M$  и  $D$  соответственно. Далее вычисляем  $m'_i = m_i - \mu_M$  и  $d'_i = d_i - \mu_D$ .

Пусть  $M'$  и  $D'$  – конкатенации точек  $m'_i$  и  $d'_i$ . Вычислим матрицу  $W$ :

$$W = M'D'^T, \quad (4)$$

и осуществим её сингулярное разложение

$$W = U\Lambda V^T. \quad (5)$$

Можно показать, что оптимальным решением  $R$  и  $t$ , которое минимизирует весовую функцию (1), будет  $R = UV^T$  и  $t = \mu_M - R\mu_D$  [5].

Этапы алгоритма ICP итеративно повторяются до достижения определённого условия: заданного количества итераций, величины суммарной ошибки и т.д. Выделим следующие функциональные этапы в каждой итерации.

1. Определение соответствующих точек из двух массивов точек. На каждой итерации для заданной точки необходимо найти ближайшую соседнюю точку из второго массива при условии, что расстояние между этими точками будет меньше заданного порога.

2. Вычисление центров масс двух облаков точек.

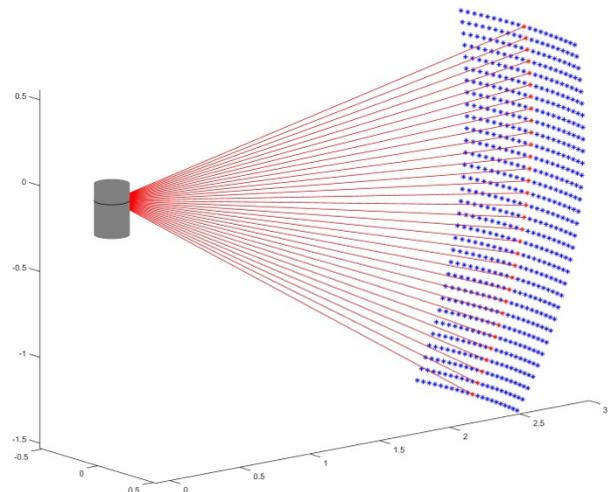
3. Вычисление матрицы ковариации  $W$ .

4. Осуществление сингулярного разложения матрицы ковариации  $W$ .

5. Применение найденных трансформаций к массиву точек.

Рассмотрим поэтапно возможности распа-

раллеливания алгоритма. Первый шаг может быть реализован несколькими альтернативными подходами:  $k$ -мерное дерево, грубый перебор, поиск внутри сегмента [6]. Кроме этого, предлагается использовать естественную структуру данных с сенсора, формируемую в ходе их получения. Так, для лидара Velodyne HDL-32E формирование массива данных происходит последовательно по мере вращения головной части, по 32 измерения в каждой из позиций. Таким образом, предлагается осуществлять поиск ближайшей соответствующей точки во множестве измерений, полученных в некотором сегменте до и после линейки измерений, которой принадлежит текущая точка (рисунок 1). Поиск в полученном ограниченном множестве можно осуществлять также разными методами, причём вычислительные затраты на сегментацию могут быть выше, чем сразу искать грубым перебором. Контроль за выполнением условия по расстоянию осуществляется в ходе поиска, и если найденная ближайшая точка не удовлетворяет условию по расстоянию, она не учитывается.



**Рисунок 1 – Схематическое изображение пространства поиска ближайших точек: для каждой из точек текущего среза выделяется некоторый сегмент до и после среза, в котором ищется ближайшая точка**

Распараллеливание вычисления центров масс осуществляется за счёт вычисления частных сумм в различных потоках, а затем вычисление общей суммы в итоге. Количество слагаемых в частных суммах зависит от возможностей графического процессора.

На третьем этапе в стандартной реализации алгоритма ICP вычисление матрицы ковариации осуществляется последовательно:

$$W_i = W_{i-1} + M'_i D_i'^T, \quad (6)$$

где  $M'_i$  и  $D'_i$  – конкатенации  $i$ -го количества

точек, а  $W_i$  – матрица на соответствующем этапе вычисления. Для параллельного выполнения этого вычисления необходимо определить выражение для каждого элемента матрицы  $W$ , размерность которой ввиду рассматриваемого трехмерного пространства будет  $(3 \times 3)$ . Поскольку

$$W = \sum_{i=1}^N W_i, \quad (7)$$

принимая, что  $w_{k,m}$  – элемент матрицы  $W$ , где  $k$  – номер строки,  $m$  – номер столбца матрицы,  $m'_{i_x}, m'_{i_y}, m'_{i_z}$  и  $d'_{i_x}, d'_{i_y}, d'_{i_z}$  – элементы одного столбца матриц  $M'$  и  $D'$  соответственно, которые являются координатами точки в трехмерном пространстве, получаем

$$\begin{cases} w_{1,1} = \sum_{i=1}^N m'_{1,i} \cdot d'_{1,i} \\ w_{1,2} = \sum_{i=1}^N m'_{1,i} \cdot d'_{2,i} \\ \dots \\ w_{3,3} = \sum_{i=1}^N m'_{3,i} \cdot d'_{3,i} \end{cases} \quad (8)$$

В современных реализациях алгоритма сингулярного разложения количество требуемых вычислительных ресурсов напрямую зависит от размера матрицы [7]. Поскольку её размер мал, то существенных затрат на вычисление не требуется. Поэтому необходимости в переносе вычисления сингулярного разложения на графический процессор нет.

Распараллеливание последнего этапа (применения найденной трансформации) является наиболее эффективным, поскольку к одинаковому типу данных (координаты точек) применяется одинаковый набор инструкций (применение перемещения как суммирование и умножение на матрицу поворота).

Таким образом, особенностями предложенной модификации алгоритма ИСР являются метод нахождения ближайших точек на основе исходной структуры данных с сенсора, а также параллельное вычисление отдельных этапов (вычисление матрицы ковариации и центроидов массивов точек) алгоритма.

**Экспериментальная часть.** Предложенная модификация алгоритма была реализована в программной среде *Matlab*, где проверялась его эффективность на смоделированных данных в сравнении с обычной реализацией алгоритма, а также на языке *C++* с применением программно-аппаратного комплекса *Nvidia CUDA*.

Рассмотрим сначала временные характеристики параллельной и последовательной реализации алгоритма на смоделированных данных в программной среде *Matlab*. При реализации параллельной версии алгоритма был использован пакет *Parallel Computing Toolbox*.

Общий вид смоделированных поверхностей представлен на рисунке 2. Поверхность имеет синусоидальную форму и состоит из 62750 точек – примерно столько же, сколько содержит один массив данных с лидара *Velodyne* после одного полного вращения.

Важным аспектом является определение приемлемого количества итераций алгоритма. Очевидно, что время выполнения сопоставления пропорционально количеству итераций, а с другой стороны, чем больше итераций алгоритма, тем выше точность сопоставления. Для экспериментального определения этого параметра построим зависимость среднего расстояния между соответствующими точками (априорная информация о том, какие точки являются действительно соответствующими, у нас имеется) от количества итераций алгоритма (рисунок 3).

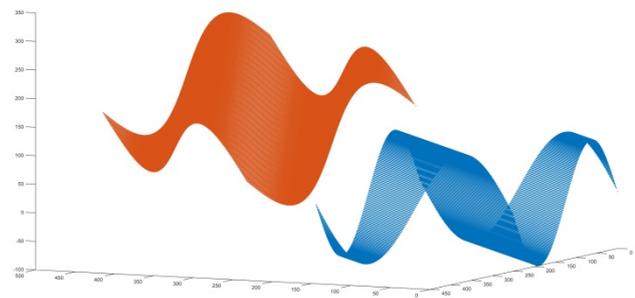


Рисунок 2 – Общий вид смоделированных поверхностей для сопоставления алгоритмом ИСР

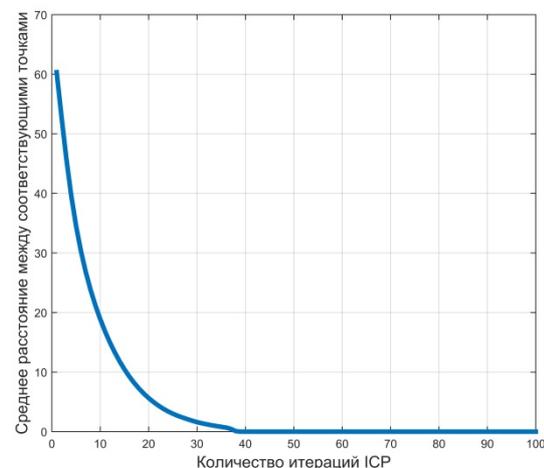
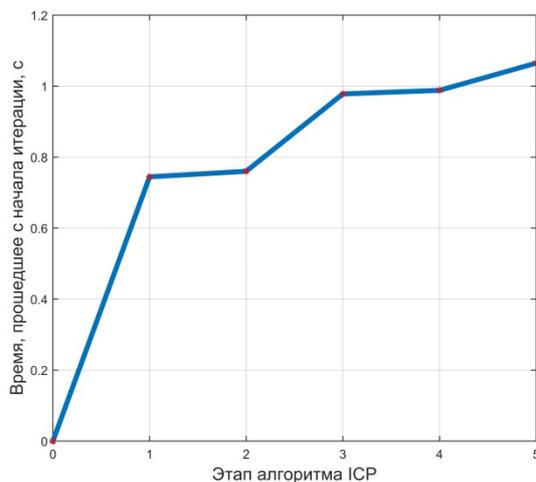


Рисунок 3 – График зависимости среднего расстояния между соответствующими точками от количества итераций алгоритма ИСР

Как видно из полученного графика, на уровне 38 итераций темп уменьшения расстоя-

ния резко падает, в результате чего среднее расстояние изменяется незначительно. На основе этого можно сделать вывод, что приемлемое количество итераций находится в пределах от 30 до 40. В ходе экспериментов нами будет использоваться количество итераций, равное 40.

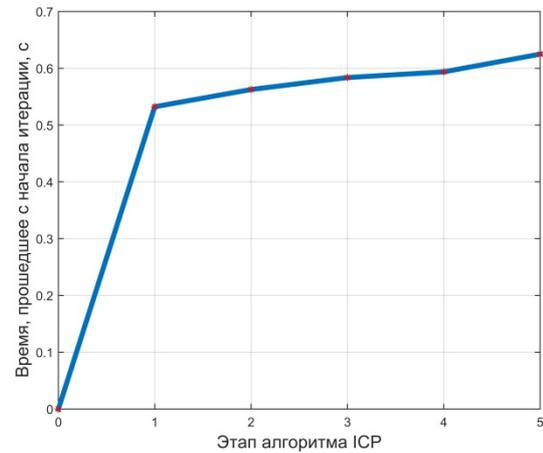
Построим временные диаграммы итерации алгоритма ICP, чтобы проанализировать временные затраты на выполнение каждого этапа. Временная диаграмма для стандартной реализации алгоритма (на центральном процессоре) представлена на рисунке 4. Из графика видно, что более половины всего времени было затрачено на поиск соответствующих точек. Поскольку в данном эксперименте используются смоделированные данные, то применение подхода выделения пространства поиска в структурированных данных не представляется необходимым и вместо этого используется  $k$ -мерное дерево. На рисунке 5 представлена временная диаграмма итерации алгоритма ICP, реализованного с использованием параллельных вычислений. Распараллеливанию подверглись второй, третий, пятый, а также частично первый этапы. Ввиду низкой эффективности распараллеливания этап сингулярного разложения остался без изменения, при этом на первом этапе фильтр по расстоянию стал параллельным.



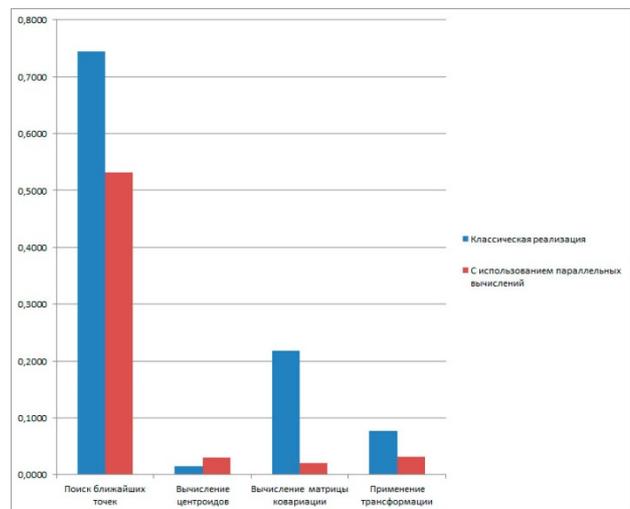
**Рисунок 4 – Временная диаграмма итерации алгоритма ICP в классической реализации. Нумерация этапов соответствует тексту**

Как видно из графиков, время выполнения одной итерации значительно сокращено (около 70%) при том, что наиболее ресурсоёмкий этап подвергся распараллеливанию лишь частично. Для сравнительного анализа времени выполнения этапа для двух реализаций алгоритма на рисунке 6 представлена диаграмма. Из неё видно, что применение параллельных вычислений привело к уменьшению времени выполнения этапов поиска ближайших точек, вычисления матрицы ковариации, вычисления центроидов, а также применения трансформации.

ковариации, а также применения перемещения (от полутора до десяти раз). Этап вычисления центроидов имеет несколько худшие показатели в параллельном исполнении, причины чего сложно объяснить ввиду закрытой реализации функции в среде Matlab.



**Рисунок 5 – Временная диаграмма итерации алгоритма ICP с использованием параллельных вычислений нумерация этапов соответствует тексту**



**Рисунок 6 – Сравнительная диаграмма времени выполнения этапов алгоритма ICP для классической реализации и реализации с использованием параллельных вычислений**

Перейдём теперь к реализации на языке C++, в ходе которой была создана программа сбора, сопоставления и визуализации полученных данных, а также программный компонент для системы ROS (*Robot Operation System* – Операционная система для робота). Для написания кода использовалась среда разработки Qt Creator на операционной системе Ubuntu.

Эксперименты на производительность проводились на персональном компьютере с процессором Intel Core-i7 4930K с 32 Гб оперативной памяти и графическим ускорителем Nvidia

GeForce 780 Ti. Параметры производительности функций, исполняющихся на графическом процессоре, оценивались с помощью утилиты Nvidia Visual Profiler. В таблице 1 приведено время выполнения функций, а также доля затрат на вычисление функции от общих вычислительных затрат графического процессора при выполнении алгоритма. Ввиду особенности реализации некоторые этапы алгоритма выполнялись в несколько функций. Как видно из таблицы, наиболее затратным оказался этап поиска ближайших точек. Он напрямую зависит от количества точек в исследуемом массиве, поэтому перед запуском алгоритма необходимо провести фильтрацию с целью удаления всех точек, которые могут помешать сопоставлению (точки земли, точки вдали от МРП и т.д.). В суммарное количество вычислительных затрат в таблице 1 не включены операции по копированию и выделению памяти, которые составляют небольшую долю от общих вычислений. Тем не менее, их оптимизации стоит уделять важное внимание, поскольку для выполнения операций на графическом процессоре необходимо скопировать все данные на память графического ускорителя, а при больших объемах данных это может быть затратным.

**Таблица 1 – Параметры производительности функций итеративного алгоритма ближайших точек в параллельной реализации**

Название функции	Выполняемое действие	Среднее время выполнения, мс	Процент от общих вычислительных затрат
<i>NNsearchBF</i>	Поиск ближайших точек с фильтром по расстоянию	3,431	91,4
<i>centroidsDual- Calc</i>	Вычисление частичных сумм центроидов	0,048	2,6
<i>centroidsDual- Final</i>	Вычисление центроидов из частичных сумм	0,027	1,5
<i>matrixSVDCalc</i>	Вычисление частичных сумм матрицы ковариации	0,053	1,4
<i>matrixSVDFinal</i>	Вычисление матрицы ковариации из частичных сумм	0,015	1,4
<i>rotateCloud</i>	Применение вращения к массиву точек	0,026	0,7
<i>applyTranslation</i>	Применение перемещения к массиву точек	0,013	0,4
<b>Всего</b>		<b>3,613</b>	<b>99,4</b>

Результаты таблицы 1 показывают, что суммарное время выполнения одной итерации алгоритма не превышает 4 мс. При общем количестве итераций 20-30 время одного сопоставления не будет превышать 100 мс, что является хорошим результатом при условии, что частота получения массивов точек с лидара Velodyne HDL-32E составляет 10 Гц. Помимо непосредственного сопоставления, вычислительные ресурсы также потребуются на первичную обработку данных, фильтрацию и т.д., что, в конечном счёте, несколько снизит временные показатели. Однако в таком случае возможно использовать не каждый массив с лидара, а пропуская некоторое их число, что, в итоге, незначительно скажется на общем объёме информации. Важно отметить, что при исследовании в настоящей работе используются лишь данные с лидара без дополнительной информации о перемещении. Поэтому при наличии альтернативной адекватной оценки перемещения возможно использование начального приближения для алгоритма ICP, что позволит уменьшить количество итераций при сохранении качественных показателей. Такой подход наиболее приемлем в современных робототехнических системах, где необходимо использовать всю доступную информацию

Ранее было отмечено, что важным аспектом при использовании современных бортовых вычислительных комплексов является их пониженное энергопотребление. Ввиду того, что размещение обычного персонального компьютера типа того, что использовался при тестовых экспериментах на МРП не всегда возможно, предложенный алгоритм был реализован для использования в системе ROS и размещён на мобильной вычислительной платформе Nvidia Jetson TK1 (рисунок 7). На этой платформе имеется четырёхядерный процессор с интегрированным графическим ядром, которое, в отличие от других аналогичных платформ, позволяет использовать параллельные вычисления на базе архитектуры CUDA. Такой подход позволяет не только размещать вычислительный комплекс для обработки большого объёма информации непосредственно на самой МРП, но и масштабировать вычислительные модули и распределять обработку данных между ними.

Конечным результатом работы алгоритма является массив координат, в которых были получены соответствующие массивы данных. Применяя трансформации к массивам данных в соответствии с их местоположением, нами формируется единый массив, который представляет собой карту окружающего пространства в формате множества точек.



**Рисунок 7 – Вычислительный модуль с интегрированным графическим ядром Nvidia Jetson TK1**

**Выводы.** Разработанный алгоритм является важным элементом в системах построения карт и анализа изменения местоположения МРП на основе сопоставления данных с лидара. Он позволил значительно сократить время обработки получаемых массивов данных с лазерного сканера и сделал доступным дополнительный источник оценки перемещения.

Тем не менее, по ряду направлений требуется провести дополнительные исследования с целью повышения точности и качества получаемых интегральных представлений о ПФ. Во-первых, требуется объективная оценка способов представления карт: множество точек, воксельное пространство, карта высот или какой-либо другой подход. Во-вторых, любой сенсор имеет ограниченные возможности по обзору, поэтому необходимо создать подход для эффективной аппроксимации данных с сенсора. В-третьих, любая измерительная система имеет погрешность, поэтому имеет смысл говорить не о наличии некоторой точки с определёнными координатами, а о наличии в некотором локальном пространстве объекта с определённой вероятностью. И, наконец, для объективной оценки различных подходов в области навигации и локализации МРП требуется методика испытаний и анализа результатов.

татами, а о наличии в некотором локальном пространстве объекта с определённой вероятностью. И, наконец, для объективной оценки различных подходов в области навигации и локализации МРП требуется методика испытаний и анализа результатов.

#### **Библиографический список**

1. M.Li, W.Li, J.Wang, Q.Li, A.Nüchter. Dynamic VeloSLAM – Preliminary Report on 3D Mapping of Dynamic Environments. 2012 IEEE Intelligent Vehicles Symposium, Workshop on Navigation, Perception, Accurate Positioning and Mapping for Intelligent Vehicles. Madrid, Spain, 2012.
2. P.J.Besl, N.D.McKay. A Method for Registration of 3-D Shapes. IEEE Trans. on Pattern Analysis and Machine Intelligence (Los Alamitos, CA, USA: IEEE Computer Society) 14 (2), 1992, P. 239–256.
3. C.Yang, G.Medioni. Object modelling by registration of multiple range images. Image Vision Comput. (Newton, MA, USA: Butterworth-Heinemann), 1991, P. 145–155.
4. S. Rusinkiewicz, M. Levoy. Efficient Variants of the ICP Algorithm. Third International Conference on 3-D Digital Imaging and Modeling, 2001. P.145.
5. Борисов А.Г., Голь С.А., Лукаш С.С. Анализ эффективности алгоритмов обработки трёхмерных лазерных сканов в задаче построения карт для навигации мобильных роботов // Вестник Рязанского государственного радиотехнического университета. 2013. № 4-3 (46). С. 34-42.
6. A. Segal, D. Haehnel, S. Thrun. Generalized-icp. Proceedings of Robotics: Science and Systems (Seattle, USA), 2009.
7. Yamamoto Y., Fukaya T., Uneyama T., Takata M., Kimura K., Iwasaki M., Nakamura Y. Accelerating the Singular Value Decomposition of Rectangular Matrices with the CSX600 and the Integrable SVD. LNCS, 2007. Vol. 4671/2007. Springer Berlin.

УДК 681.3.06: 519.7

**О.А. Козелков**

## **ЧАСТНЫЕ И ОБОБЩЕННЫЕ ОЦЕНКИ РЕАЛИЗУЕМОСТИ ПРОЕКТА ПРОИЗВОДСТВА НОВОЙ ТЕХНИКИ**

*Рассматриваются основные принципы оценивания реализуемости проектов производства новой техники. Основной является системная концепция, в которой уделяется внимание структурно-функциональному подходу с иерархической структурой показателей. Предлагается способ получения обобщенных оценок реализуемости проекта на основе частных показателей функционирования подсистем предприятия. Определена структура частных и обобщенных оценок реализуемости проекта производства новой техники и описан механизм их интеграции. Предложенный подход применен в системе принятия решений по реализации проектов производства новой техники.*

**Ключевые слова:** оценка реализуемости, проект производства новой техники, факторы производства, свертка, интегральный показатель.

**Введение.** Для достижения целей стабильного промышленного развития России необходимо формирование комплекса высокотехнологичных производств, а также развертывание и модернизация глобально ориентированных специальных производств. В связи с этим для поддержания конкурентоспособности производственных предприятий производится разработка комплексных программ инновационного развития. Однако сопоставление перспективных планов развития и результатов промышленного производства позволяет сделать вывод, что цели, поставленные в программах развития, не выполняются [1]. Следовательно, необходимо на прединвестиционном этапе осуществлять оценку реализуемости инновационных проектов, выявлять потенциальные причины их невыполнения и принимать решения, способствующие достижению целей развития.

**Цель работы.** Общей задачей при формировании инновационных проектов производства является оценка реализуемости целей с учетом выделенных объемов инвестиций и сроков необходимой модернизации. Как правило, реализация этих задач заключается в проведении работ в рамках предварительного технико-экономического обоснования в соответствии с рекомендациями «Руководства по подготовке промышленных технико-экономических исследований» [2]. Объектом исследования в данной работе являются проекты производства новой техники, которые реализуют стратегии развития в области инновации продукта. При этом требуются научно-обоснованные оценки производственного предприятия как исполнителя проекта по отдельным функциональным составляющим (технико-технологической, кадровой, логистической) [3].

На сегодняшний день оценка реализуемости производилась в основном с точки зрения ресурсного подхода на основе экономических моделей, при этом основным показателем реализуемости являлись характеристики финансового обеспечения [4-6]. Однако исследование реализуемости проекта производства новой техники ввиду комплексного характера решаемых задач требует применения системных принципов анализа причин возможного недостижения целей проекта [7]. Отсюда вытекает необходимость исследования качества процесса производства новой техники при всестороннем комплексном анализе условий и факторов, влияющих на этот процесс.

Реализуемость проекта следует рассматри-

вать как совокупность определяющих свойств, учитывая взаимосвязь с техническими и функциональными возможностями производства.

В рамках настоящей работы внимание уделяется этапу прединвестиционного исследования, что позволяет оценить целесообразность планирования дальнейших работ по таким характеристикам, как технологическая обеспеченность, реализуемость задач внутренней логистики, обеспеченность кадровым составом, которые влияют на основные требования проекта к срокам и стоимости его реализации. Таким образом, оценивание реализуемости требований проекта имеет особую значимость на прединвестиционной фазе, поскольку качество и достоверность последующих этапов зависит во многом от принятых решений и их интерпретации.

С помощью оценки проекта производства новой техники решается важная задача по выяснению и обоснованию технической возможности и экономической целесообразности его цели.

#### **Теоретические исследования**

Анализ и оценка проектов производства новой техники осуществляются по частным и комплексным показателям с последующим формированием обобщенной оценки.

Значения частных и комплексных показателей представляют собой характеристики текущего состояния и результатов работы объекта в виде количественных оценок или суждений. Одна из центральных операций оценивания сложных объектов состоит в сопоставлении показателей реализуемости исследуемого объекта с аналогичными по содержанию базовыми показателями и переводе абсолютных оценок в сравнительные, относительные. Сравнительные показатели представляют локальные оценки реализуемости применительно к определенным свойствам объекта. Они дают возможность судить о реализуемости в определенных частных аспектах и могут стать основой для последующих логических выводов. Поэтому должна быть сформирована база для последующего сравнения абсолютных значений показателей реализуемости объекта с базовыми значениями.

В дополнение к частным оценкам формируются обобщенные (комплексные, интегральные) в виде как числовых оценок, так и качественных суждений. Комплексные оценки реализуемости образуются путем объединения близких по содержанию единичных оценок в одну обобщенную, синтетическую или путем оценивания ряда признаков одним комплексным показателем. Интегральной принято называть оценку, выражен-

ную одним показателем, которая имеет общий характер и аккумулирует в себе все основные свойства объекта.

Формирование итогового суждения о реализуемости объекта требует проведения дополнительного анализа или дальнейшей обработки частных и обобщенных оценок. Завершающая оценка реализуемости проекта в наиболее общем случае формируется в виде комплекса качественных оценок (суждений), компоненты которого отражают оценки отдельных показателей.

Таким образом, суждение о реализуемости инновационного проекта должно быть комплексным, многосторонним, т.е. учитывать различные признаки и свойства. При принятии решений о целесообразности инвестиций не исключается возможность использования частных показателей и оценок реализуемости, однако такие оценки не способны заменить интегральную – более полную и представительную.

Требование комплексного оценивания реализуемости проектов является следствием системного подхода в планировании и необходимости соблюдения следующих условий:

- формирование плановых решений с учетом производственных факторов;
- взаимосвязь целей и ресурсов проекта;
- учет взаимосвязей между разными уровнями управления;
- рассмотрение в единстве натурально-вещественного и стоимостного измерения объемов производства, затрат;
- взаимосвязь краткосрочных проблем с долговременной стратегией развития.

Факторы реализуемости проектов производства новой техники представляют собой систему показателей, увязанных с характеристиками производственной деятельности предприятий.

Комплексный показатель реализуемости проекта производства новой техники характеризует возможность реализации его цели на основе анализа характеристик отдельных функциональных подсистем предприятия. Он рассчитывается с учетом частных показателей реализуемости:

$$W_{IP}^{(1)} = f_1 \left[ W_{FA_1}^{(1)}, \dots, W_{FA_k}^{(1)}, \dots, W_{FA_L}^{(1)} \right],$$

где  $W_{IP}^{(1)}$  – возможность реализации проекта в целом;

$W_{FA_k}^{(1)}$  – возможность реализации по различным функциональным аспектам ( $K = \overline{1, L}$ );

$f_1$  – функция объединения (свертка) показателей  $W_{FA_k}^{(1)}$ .

В свою очередь,

$$W_{FA_k}^{(1)} = f_2 \left[ W_{P_{L,1}}^{(1)}, \dots, W_{P_{L,k}}^{(1)}, \dots, W_{P_{L,M}}^{(1)} \right],$$

где  $W_{P_{L,k}}^{(1)}$  – возможность реализации по определенному требованию проекта;

$f_2$  – свертка соответствующих показателей.

В результате анализа требований проекта вырабатывается набор обобщающих признаков, характеризующих различные функциональные аспекты предприятия, а именно:

- наличие производственных мощностей;
- достаточность ресурсов;
- возможный уровень технологических процессов;
- директивные сроки выполнения работ;
- динамика производства.

На рисунке 1 представлена укрупненная схема составляющих элементов оценки реализуемости выполнения предприятием работ проекта производства новой техники. Отметим, что характеристики финансовой подсистемы предприятия не участвуют в формировании оценки реализуемости, так как она является организационным элементом, а не производственным. Однако получаемые оценки реализуемости служат основой для принятия управляющих решений, в том числе и по финансированию проекта.

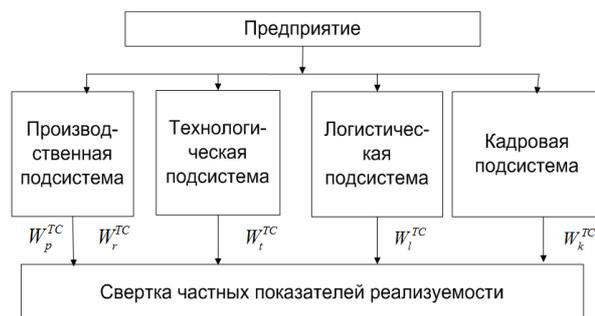


Рисунок 1 – Составляющие оценки реализуемости

Оценка реализуемости проводится по всем функциональным направлениям производства с учетом производственного опыта и управленческих аспектов. После этого формируется значение показателя  $W_{TC}^{(1)}$  как оценка возможности производства конкретного вида новой продукции (технической системы).

Возможность реализации проекта по производству отдельного вида продукции  $W_{TC}^{(1)}$  оценивается вектором следующих показателей:

$$W_{TC}^{(1)} = f_3 \left[ W_p^{TC}, W_r^{TC}, W_t^{TC}, W_l^{TC}, W_k^{TC} \right],$$

где  $W_p^{TC}$  – оценка реализуемости проекта с позиции производственных мощностей предприятия;

$W_r^{TC}$  – оценка реализуемости ТС с позиции достаточности ресурсного обеспечения;

$W_t^{TC}$  – возможность реализации ТС с пози-

ции технологического обеспечения;

$W_l^{TC}$  – реализуемость ТС с позиции логистического обеспечения;

$W_k^{TC}$  – реализуемость ТС с позиции кадрового обеспечения;

$f_3$  – функция объединения (свертка) частных показателей.

На заключительном этапе производится оценка реализуемости проекта производства новой техники в целом.

Комплексный показатель реализуемости проекта  $W_{PP}^{(2)}$  формируется в результате объединения следующих показателей:

$W_i^{TC}$  – реализуемость проекта производства  $i$ -го вида продукции ( $i = \overline{1, N}$ );

$W_T^{TC}$  – возможность реализации проекта с позиции обеспечения сроков;

$W_{OP}^{TC}$  – возможность реализации проекта с позиций планируемого объема производства.

Таким образом, формируется

$$W_{PP}^{(2)} = f \left[ W_1^{TC}, \dots, W_N^{TC}, W_T^{TC}, W_{OP}^{TC} \right].$$

На рисунке 2 представлена укрупненная схема сценария проведения оценки реализуемости проекта.

Оценивание реализуемости начинается с анализа возможностей производства по каждому виду продукции. Эта информация позволяет получить оценки в условиях планирования многономенклатурного производства.

Рассмотрим процедуру формирования интегрального показателя реализуемости проекта.

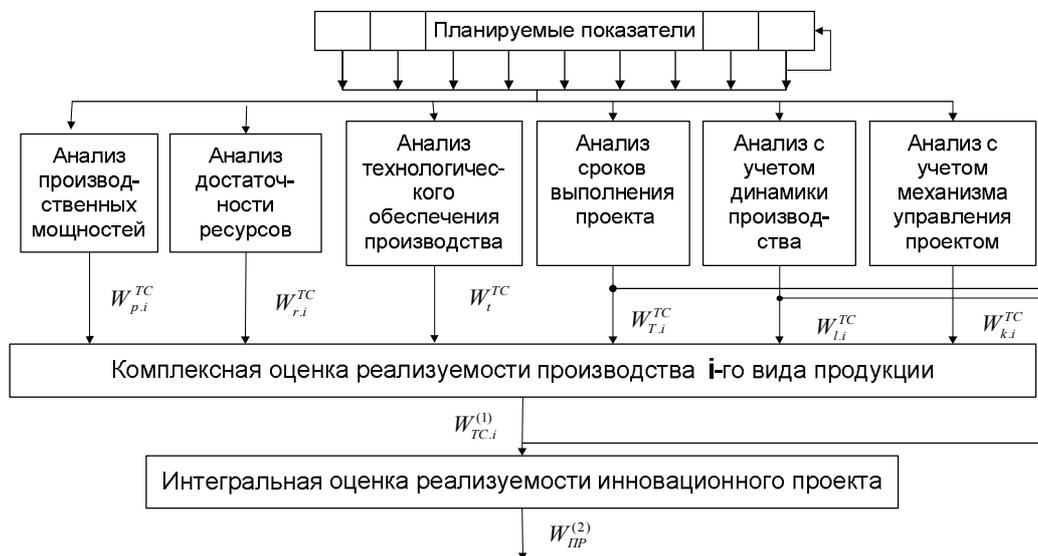


Рисунок 2 – Схема оценки реализуемости проекта производства новой техники

В результате анализа содержания проекта вырабатывается набор обобщающих признаков – требований к новой технике, характеризующихся различными аспектами, связанными с возможностями предприятия.

Формируется интегральная оценка проекта производства новой техники  $W_{интегр}$  в виде следующей функции:

$$W_{интегр} = f(W_i, i = \overline{1, N}). \quad (1)$$

Формирование  $W_{интегр}$  представляет собой нетривиальную задачу, поскольку эта функция должна отражать такие свойства, как важность того или иного фактора, приоритет различных элементов. Поэтому наиболее естественным представляется метод построения функции с использованием продукционной экспертной системы с базой знаний, отражающей влияние частных характеристик на обобщенную [8].

Для объединения частных показателей производства в интегральную оценку возможности

выполнения всего проекта целесообразно применять различные стратегии рассуждения – от самой пессимистической (реализация функции минимизации) до самой оптимистической (реализация функции максимизации). Использование пессимистической оценки позволяет выявить «узкие места» выполнения проекта.

Свертку по формуле (1) можно представить в виде лингвистических правил, отражающих качественное изменение  $W_{интегр}$  при задаваемых значениях важности, приоритета либо других критериев, список которых определяется пользователем при настройке системы. В качестве аппарата логического описания правил необходимо сформировать набор лингвистических переменных для описания значений оценок реализуемости и влияющих факторов (важность, приоритет, стоимость и т.д.).

Получение маргинальных возможностей выполнения проекта  $W_{интегр,i}^{марг}$  осуществляется с

помощью методов теории нечетких множеств. При этом используются алгоритмы фазификации частных значений реализуемости  $W_i$  и влияющих факторов. При необходимости для получения количественных значений используются методы дефазификации.

Формирование обобщенной оценки  $W_{интегр}$  реализуемости проекта производства новой техники осуществляется на основе маргинальных оценок  $W_{интегр.i}^{марг}$ :

$$W_{интегр.i}^{марг} = f(W_i, Imp_i),$$

где  $W_i$  – реализуемость элемента  $i$ -го уровня;

$Imp_i$  – важность элемента  $i$ -го уровня для реализации планов верхнего уровня).

Операция объединения определяется формулой

$$W_{интегр} = \bigcup_{i=1}^N W_{интегр.i}^{марг} \quad (2)$$

где  $N$  – общее количество правил.

Реализация отношения (2) зависит от настройки системы оценивания (пессимистическая, оптимистическая, средняя оценки). Различные взгляды лица, принимающего решение, реализованы с помощью логических операций.

Оптимистической операцией является дизъюнкция, то есть

$$W_{интегр} = \bigvee_{i=1}^N W_{интегр.i}^{марг} \quad (3)$$

Данная операция оценивает возможности проекта с учетом самого лучшего из своих маргинальных показателей.

В зависимости от типа логической системы выражение (2) можно реализовать различными способами, например:

$$W_{интегр} = \max \{W_{интегр.i}^{марг}, i = 1, \dots, N\}, \quad (4)$$

$$W_{интегр} = \min \{1, \sum W_{интегр.i}^{марг}, i = 1, \dots, N\}. \quad (5)$$

Формула (4) определяется логикой Клини-Дайнеса, а (5) – Лукасевича [9]. Наибольшее значение всегда дает соотношение (3), которое и является наиболее оптимистическим.

Самой пессимистической операцией, заменяющей объединение в формуле (2), является конъюнкция. Она оценивает возможность выполнения проекта с учетом самого худшего показателя реализуемости:

$$W_{интегр} = \bigwedge_{i=1}^N W_{интегр.i}^{марг} \quad (6)$$

В зависимости от типа логической системы выражение (6) можно реализовать различными способами, например:

$$W_{интегр} = \min \{W_{интегр.i}^{марг}, i = 1, \dots, N\}, \quad (7)$$

$$W_{интегр} = \max \{0, \sum W_{интегр.i}^{марг}, i = 1, \dots, N\}.$$

Наименьшее значение всегда дает формула (7), которая и является наиболее пессимистической.

Промежуточный уровень занимает операция импликации, которую также можно применить при реализации отношения (2):

$$\bigwedge_{i=1}^N W_{интегр.i}^{марг} \rightarrow W_{интегр}. \quad (8)$$

Данная операция оценивает возможности предприятия на основе маргинальных возможностей. В зависимости от типа логической системы формулу (8) можно реализовать следующим способом:

$$W_{интегр} = \max \left\{ 1 - \sum_{i=1}^N W_{интегр.i}^{марг}, i = 1, \dots, N \right\}.$$

Использование той или иной операции, а также логической системы задается эвристически и зависит от представлений лица, принимающего решение, обо всем проекте и отдельных его компонентах.

**Применение метода комплексной оценки реализуемости в СППР по реализации проекта производства новой техники.** Разработана структура и программные модули компьютерной системы поддержки принятия решений (СППР) по реализации проектов производства новой техники на конкретном предприятии.

СППР включает четыре подсистемы: «Планирование», «Технология», «Моделирование», «Кадры».

Каждая из подсистем СППР позволяет получить частные оценки реализуемости проекта с учетом характеристик соответствующих функциональных подсистем предприятия. Эта информация служит основой принятия решений о необходимости модернизации предприятия для реализуемости отдельных требований проекта производства новой техники.

Расчет показателей реализуемости производится путем сравнения оценок характеристик производства (полученных путем прогнозирования или моделирования – математического или имитационного [10]) с требованиями проекта. Приведем расчетные формулы для частных показателей реализуемости.

1. Показатель реализуемости объема производства отдельного вида продукции (с учетом планируемых ресурсов):

$$W_x = \sum_{i=1}^n \left( 1 - \frac{\Delta x_i}{x_p} \right),$$

где  $\Delta x_i = \begin{cases} x_p - x_i, & x_p > x_i \\ 0, & x_p \leq x_i \end{cases}$  – недостающий объем

производства  $i$ -го вида продукции, здесь  $x_p$  – требуемый объем производства новой продукции (по проекту),  $x_i$  – оценка возможного объема производства.

2. Показатель реализуемости суммарного объема производства (с учетом характеристик оборудования):

$$W_{N1} = 1 - \frac{\Delta B}{\sum_{i=1}^n x_i},$$

где  $\Delta B = \begin{cases} B_p - B_o, B_p > B_o \\ 0, B_p \leq B_o \end{cases}$  – недостающий объем

производства, здесь  $B_p$  – требуемый объем производства,  $B_o$  – оценка возможного объема производства.

3. Показатель реализуемости ресурсного обеспечения:

$$W_L = 1 - \frac{\Delta L}{L_p},$$

где  $\Delta L = \begin{cases} L_p - L_o, L_p > L_o \\ 0, L_p \leq L_o \end{cases}$  – недостающий объем

ресурсов (в стоимостном выражении), здесь  $L_p$  – планируемые затраты на материально-техническое обеспечение,  $L_o$  – оценка необходимых затрат ресурсов производства.

4. Показатели логистической реализуемости на основе оценки временных и объемных требований проекта. При этом рассматриваются варианты:

а) если  $N_m < N_p$  и  $T_m \leq T_p$  (здесь  $N_m$  – смоделированный объем производства,  $N_p$  – требуемый в проекте объем производства,  $T_m$  – смоделированное время производственного цикла,  $T_p$  – требуемое в проекте время производственного цикла):

$$W_{N2} = 1 - \frac{N_p - N_m}{N_p};$$

б) если  $N_m \geq N_p$  и  $T_m > T_p$ :

$$W_{T1} = 1 - \frac{T_m - T_p}{T_p}.$$

5. Показатели реализуемости временных и стоимостных требований проекта на основе оценок кадрового обеспечения проекта путем сравнения результатов расчетов и моделирования:

а) если  $C_m > C_p$  (здесь  $C_m$  – оптимальная стоимость работ с выбранным составом исполнителей,  $C_p$  – требуемая в проекте стоимость работ), то

$$W_c = 1 - \frac{C_m - C_p}{C_p},$$

б) если  $T_m > T_p$  (здесь  $T_m$  – смоделирован-

ное время производственного цикла,  $T_p$  – планируемое в проекте время производственного цикла), то реализуемость сроков

$$W_{T2} = 1 - \frac{T_m - T_p}{T_p}.$$

С использованием частных оценок реализуемости решается задача получения интегральной оценки, на основе которой инвестором принимается решение о принципиальной возможности производства новой продукции на данном предприятии.

Если частные оценки реализуемости характеризуют различные аспекты проекта, то интегрированный эффект рассчитывается путем их умножения. В противном случае, если частные оценки были определены в различных подсистемах, но выражаются одним требованием в проекте (например, количество производимой продукции), то комплексная оценка определяется операцией выбора минимального значения по пессимистическому критерию.

Сгруппируем частные оценки реализуемости по четырем аспектам - получим локальные оценки реализуемости (второго уровня):

– объемный

$$W_1 = \min(W_x, W_{N1}, W_{N2});$$

– технологический

$$W_2 = W_{TH};$$

– временной

$$W_3 = \min(W_{T1}, W_{T2});$$

– стоимостный

$$W_4 = \min(W_L, W_C).$$

Тогда интегральный показатель реализуемости проекта (обобщенная оценка третьего уровня)

$$W = W_1 W_2 W_3 W_4.$$

Таким образом, лицо, принимающее решение, может делать общие качественные выводы на основе интегральной оценки реализуемости. Граница принципиальной реализуемости проекта зависит от уровня допустимого риска ( $R$ ), который определяется соотношением ожидаемой прибыли (или важности проекта с точки зрения заинтересованных сторон или промышленной политики государства) и суммой необходимых инвестиций. Если интегральная оценка удовлетворяет условию:

$$W < 1 - R,$$

то проект считается принципиально реализуемым. Тогда на основе частных оценок реализуемости принимаются решения по модернизации отдельных функциональных подсистем произ-

водства или на основе локальных оценок реализуемости даются рекомендации по корректировке требований отдельных аспектов проекта.

**Выводы.** В данной статье основное внимание уделено методам расчета интегрального показателя реализуемости проекта производства новой техники. На основе показателя реализуемости можно судить о возможности выполнения технических требований проекта в условиях заданных ограничений по ресурсному обеспечению и уровню производства.

Для оценки возможности выполнения проекта производства новой техники предложены частные показатели реализуемости. Разработан метод формирования комплексной оценки реализуемости. Сформирована схема соотношения частных, локальных (комплексных) и интегральных показателей реализуемости проекта. Научная новизна состоит в разработке метода системной оценки реализуемости проектов производства новой техники на предприятии с использованием функционально-структурного подхода к анализу возможностей предприятия для получения локальных и интегральных оценок реализуемости, позволивший по отношению к существующим методам повысить достоверность решений прединвестиционного планирования.

Практическая значимость результатов состоит в том, что их применение в качестве методологической основы при разработке информационной технологии СППР по реализации инновационного проекта на предприятии позволит принимать научно-обоснованные решения по планированию развития производства и избежать затрат ресурсов на заведомо нереализуемые проекты.

### **Библиографический список**

1. Индикаторы инновационной деятельности: 2011: стат. сб. – М.: Национальный исследовательский университет «Высшая школа экономики». – 456 с.
2. Беренс В., Хавранек П.М. Руководство по подготовке промышленных технико-экономических исследований: пер. с англ. перераб. и дополн. изд. – М.: АОЗТ „Интерэксперт“, 1995. – 343 с.
3. Козелков О.А. Структурная и теоретико-множественная модели задачи оценивания технологических параметров в производственной системе // Вестник Рязанского государственного радиотехнического университета. 2014. № 47. С. 134 – 137.
4. Волков В.А., Орлов А.И. Организационно-экономические подходы к оценке реализуемости инновационно-инвестиционных проектов // Научный журнал КубГАУ, №97(03), 2014, электронный ресурс: <http://ej.kubagro.ru/2014/03/pdf/13.pdf>.
5. Бендигов М.А. Оценка реализуемости инновационного проекта // Менеджмент в России и за рубежом. 2001. № 2. С. 142 – 147.
6. Хрусталева Е.Ю., Хрусталева О.Е. Финансовая устойчивость наукоемкого предприятия как фактор оценки реализуемости инновационного проекта // Национальные интересы: приоритеты и безопасность. 2013. № 33(222). С. 16 – 23.
7. Балдин К.В., Быстров О.Ф., Передеряев И.И., Соколов М.М. Инвестиции. Системный анализ и управление. – М.: Дашков и Ко, 2009. – 288 с.
8. Barth S. Defining Knowledge Management // Knowledge Management Magazine, 2002, June. – P. 153 – 162.
9. Берштейн Л.С., Боженик А.В. Нечеткие модели принятия решений: дедукция, индукция, аналогия. – Таганрог: Изд-во ТРТУ, 2001. – 110 с.
10. Козелков О.А. Модели оценки деятельности операторов сложного технологического оборудования // Вестник Рязанского государственного радиотехнического университета. 2014. № 48. С. 126 – 132.