
ИНФОРМАТИКА, УПРАВЛЕНИЕ И СИСТЕМНЫЙ АНАЛИЗ

УДК 269.7.537

EDN TXAJAT

РАЗРАБОТКА АЛГОРИТМА ОПЕРАТИВНОЙ ОБРАБОТКИ СПУТНИКОВЫХ И ИНЕРЦИАЛЬНЫХ НАВИГАЦИОННЫХ ДАННЫХ С ПРИМЕНЕНИЕМ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

М.Н. Гусев

ORCID: 0009-0005-6366-292X e-mail: 123456789080-5@bk.ru

Арзамасское научно-производственное предприятие «ТЕМП-АВИА»

Арзамас, Нижегородская область, Россия

Представлен алгоритм обработки навигационной информации, предназначенный для интеграции данных от инерциальных навигационных систем (ИНС) и глобальных навигационных спутниковых навигационных систем (СНС). Для всесторонней оценки эффективности разработанного алгоритма проведено комплексное исследование, включающее как статистический анализ, так и полунатурное моделирование на синтезированных данных. Итоги численных экспериментов продемонстрировали повышенную точность предложенного алгоритма для определения местоположения навигационного объекта в сравнении с традиционным методом, основанным на использовании расширенного фильтра Калмана. Зафиксировано снижение среднеквадратической ошибки определения положения в диапазоне 8.3-14 % в условиях длительного (до 100 сек) отказа сигнала от СНС, что критически важно для обеспечения надежной навигации в условиях помех. Работоспособность предложенного алгоритма подтверждена в ходе полунатурного моделирования на вычислителе, используемом в навигационных изделиях. Предложенный подход базируется на применении машины экстремального обучения (Extreme Learning Machine – ELM), что в перспективе позволит существенно улучшить определение навигационных параметров.

Ключевые слова: навигационные системы; инерциальная навигация; спутниковая навигация; машинное обучение; машины экстремального обучения (Extreme Learning Machine – ELM); расширенный фильтр Калмана.

ДЛЯ ЦИТИРОВАНИЯ: Гусев, М.Н. Разработка алгоритма оперативной обработки спутниковых и инерциальных навигационных данных с применением методов машинного обучения // Труды НГТУ им. Р.Е. Алексеева. 2026. № 3. С. 7-15. EDN: TXAJAT

DEVELOPMENT OF ALGORITHM FOR REAL-TIME PROCESSING OF SATELLITE AND INERTIAL NAVIGATIONAL DATA USING MACHINE LEARNING METHODS

M.N. Gusev

ORCID: 0009-0005-6366-292X e-mail: 123456789080-5@bk.ru

JSC «Arzamas Research and Production Enterprise «TEMP-AVIA»

Arzamas, Nizhny Novgorod region, Russia

Abstract. This paper presents an algorithm for processing navigation information designed to integrate data from inertial navigation systems (INS) and global navigation satellite systems (GNSS). A comprehensive investigation, including both statistical analysis and hardware-in-the-loop (HIL) simulation using synthesized data, was conducted to thoroughly evaluate the effectiveness of the developed algorithm. The numerical experiments demonstrate that the proposed algorithm achieves higher accuracy in determining the location of a navigation object compared to the traditional approach based on the extended Kalman filter (EKF). A reduction in root mean square error of position determination in the range of 8.3-14 % was recorded under prolonged (up to 100 sec) GNSS signal outages, which is critical for ensuring reliable navigation in interference conditions. The functionality of the developed algorithm was confirmed through HIL simulation on a computer used in navigation products. This work proposes approach based on the application of an Extreme Learning Machine (ELM), which promises to significantly improve the determination of navigational parameters.

Key words: navigation systems; inertial navigation; satellite navigation; machine learning; extreme learning machine; extended Kalman filter.

FOR CITATION: Gusev M.N. Development of algorithm for real-time processing of satellite and inertial navigational data using machine learning methods. Transactions of NNSTU n.a. R.E. Alekseev. 2026. № 3. Pp. 7-15. EDN: TXAJAT

Введение

Комплексная обработка навигационной информации, получаемой от инерциальных и спутниковых навигационных систем, обычно осуществляется с помощью расширенного фильтра Калмана (ЕКФ). Его эффективность обусловлена возможностью оперативной оценки ошибок инерциальных навигационных систем (ИНС), базирующейся на анализе расхождений с данными спутниковых навигационных систем (СНС) [1]. Однако корректная работа ЕКФ требует математической модели системы, создание которой не всегда представляется возможным, что ограничивает его применение. Недостаточная точность оценок ЕКФ в условиях нестабильного обмена с СНС привела к необходимости поиска альтернативных подходов к интеграции данных, в частности, к использованию методов на основе искусственного интеллекта (ИИ) [2-4]. В рамках данного исследования предложен алгоритм на основе машины экстремального обучения [5], задачей которого является оценка ошибок ИНС.

В общем виде данная задача (e) сводится к минимизации разницы между истинным значением ($x(k)$) в момент времени k и его оценки ($\hat{x}(k)$), которая была получена в результате комплексной обработки данных в тот же момент времени k :

$$e = x(k) - \hat{x}(k) \rightarrow 0$$

1. Алгоритм машины экстремального обучения

Машина экстремального обучения (Extreme Learning Machine – ELM) – метод построения нейронных сетей прямого распространения. Спектр задач, которые способны решить навигационные системы (НС) на данной основе, широк: разреженная аппроксимация, регрессия, кластеризация и др. Главное преимущество ELM заключается в простоте обучения: параметры скрытых слоев не требуют настройки и могут быть заданы случайным образом или унаследованы от предыдущих слоев. Обучение выходных весов происходит за один шаг, что значительно упрощает и ускоряет процесс [6].

На рис. 1 представлена архитектура используемой нейронной сети.

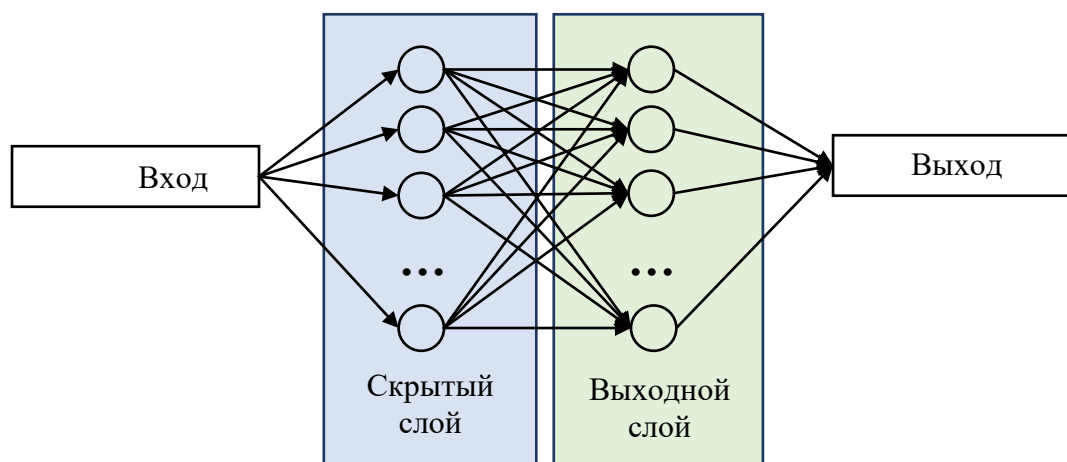


Рис. 1. Архитектура НС на основе ELM

Fig. 1. Architecture of the neural network based on ELM

Вид классического алгоритма обучения НС на основе ELM представлен ниже.

Алгоритм 1 ELM

Вход: $\Delta S_{m \times n}$ – набор обучающей информации, $\Delta X_{m \times p}$ – желаемый результат.

Выход: $Wout_{h+n \times p}$ – веса выходного слоя НС.

1. Вычисление отклика входного слоя $H_{m \times h} = F(\Delta S_{m \times n} \cdot Win_{n \times h})$;
2. При наличии функциональной зависимости ΔS от ΔX дополним матрицу $H_{m \times h+n} = [H_{m \times h} \Delta S_{m \times n}]$;
3. Находим псевдообратную матрицу $(H_{m \times h+n})^\dagger$;
4. Вычисляем веса выходного слоя $Wout_{h+n \times p} = (H_{m \times h+n})^\dagger \cdot \Delta X_{m \times p}$.

где n – количество входов НС; m – количество измерений одного обучающего окна; p – количество выходов НС; h – количество нейронов в скрытом слое НС; $F(\cdot)$ – сигмоидальная функция активации нелинейного слоя.

Использование всего массива входных данных ΔS при каждом шаге обучения приводит к значительной вычислительной нагрузке, особенно на этапах перемножения матриц и вычисления псевдообратной матрицы размером m . С ростом объема обучающей выборки (m) эта проблема становится критичной и может исключить возможность реализации алгоритма в режиме реального времени. Уменьшить затраты на вычисления накопительный режим обучения: обновление происходит лишь по факту формирования группы данных размером w . В данной работе для этих целей используется метод неперекрывающегося перемещаемого окна (рис. 2) [7]. Таким образом, коррекция весовых значений выходного блока производится по результатам обработки каждого отдельного временного отрезка.



Рис. 2. Схема обновления НС неперекрывающимся перемещаемым окном

Fig. 2. Scheme of neural network updating with a non-overlapping sliding window

Вычисление псевдообратной матрицы в уравнении может быть ресурсоемким и численно неустойчивым, что критично для выполнения в реальном времени. Для решения этой проблемы предложено свести задачу к стандартной задаче наименьших квадратов с использованием QR-факторизации [8]. Алгоритм реализован в виде следующей последовательности действий.

Алгоритм 2 ELM с QR-факторизацией

Вход: $\Delta S_{m \times n}$ – набор обучающей информации, $\Delta X_{m \times p}$ – желаемый результат.

Выход: $Wout_{h+n \times p}$ – веса выходного слоя НС.

При первом обучении алгоритм имеет следующий вид.

1. Заполнение буфера информации для обучения $\Delta S_{m \times n}$ и $\Delta X_{m \times p}$.
2. Вычисление отклика входного слоя $H_{m \times h} = F(\Delta S_{m \times n} \cdot Win_{n \times h})$.
3. При наличии функциональной зависимости ΔS от ΔX дополним матрицу $H_{m \times h+n} = [H_{m \times h} \Delta S_{m \times n}]$.
4. Формируем матрицу $HX_{w \times h+n+p} = [H_{w \times h+n} \Delta X_{w \times p}]$.
5. Находим матрицы $Q_{w \times h+n+p}$ и $R_{h+p \times h+n+p}$ через QR-факторизацию

$$\begin{bmatrix} Q_{w \times h+n+p} & R_{h+p \times h+n+p} \end{bmatrix} = qr(HX, 0).$$

6. Разложим матрицу $R_{h+n+p \times h+n+p}$ на три составляющих

$$\begin{bmatrix} Xt_{h+n \times h+n} & St_{h+n \times p} \\ 0 & Yt \end{bmatrix} = R_{h+n+p \times h+n+p}.$$

7. Находим обратную матрицу $Xt_{h+n \times h+n}^{-1}$.
8. Перемножим матрицы $Wout_{h+n \times p} = Xt_{h+n \times h+n}^{-1} \cdot St_{h+n \times p}$.

При повторном обучении алгоритм выглядит следующим образом.

1. Проведем конкатенацию $R_{h+p+w \times h+p} = \begin{bmatrix} R_{h+p \times h+p} \\ [H_{w \times h} \Delta X_{w \times p}] \end{bmatrix}$.
2. Находим матрицы $Q_{2w \times h+n+p}$ и $R_{2h+p \times h+n+p}$ через QR-факторизацию $[Q_2 \ R_2] = qr(R)$.
3. Разложим матрицу $R_{2h+n+p \times h+n+p}$ на три составляющих

$$\begin{bmatrix} Xt_{h+n \times h+n} & St_{h+n \times p} \\ 0 & Yt \end{bmatrix} = R_{2(h+n+p \times h+n+p)}.$$

4. Подготовка к следующему обучению $R = R_2$.
5. Находим обратную матрицу $Xt_{h+n \times h+n}^{-1}$.
5. Перемножим матрицы $Wout_{h+n \times p} = Xt_{h+n \times h+n}^{-1} \cdot St_{h+n \times p}$

где $qr(\)$ – QR-факторизация.

Первичный анализ показывает, что использование QR-факторизации для расчета обратной матрицы вносит погрешность, которая в контексте текущей работы не является значительной.

2. Реализация НС

В качестве входных данных используются измерения датчиков первичной информации (ДПИ, ΔS), доступные в режиме реального времени. Входной вектор состоит из шести значений (n): три показания от акселерометра (АКС) и три от датчиков угловой скорости (ДУС). Выходными данными являются значения коррекции вектора состояния ИНС (ΔX), также из шести значений: три по координатам и три по скоростям. Функционирование НС происходит в двух режимах. В случае отсутствия корректной информации от СНС, НС предсказывает коррекцию вектора состояния. При наличии таковой НС обучается, используя СНС в качестве источника эталонной информации. Для обучения НС был выбран метод перекрывающегося окна, с размером в 12 измерений. Оптимальное количество нейронов на скрытом слое определим посредством численного моделирования.

3. Численное моделирование

Численное моделирование осуществлялось на цифровом комплексе математического моделирования (ЦКМ) в режиме статистического моделирования и на комплексе имитационного моделирования (КИМ) в качестве полунатурных испытаний.

3.1. Моделирование на ЦКМ

ЦКМ – программно-математический комплекс, предназначенный для проверки алгоритмических блоков в контексте модели объекта. Он применяется не только для отработки кода, но и для проведения сопутствующего моделирования в ходе всего цикла экспериментов, начиная с ранних стадий испытаний.

Моделирование проводилось для нейронных сетей с различным количеством нейронов в скрытом слое – 12, 24, 48, 96, 192 и 384. Для каждой конфигурации алгоритма использовался один набор из 10000 уникальных сценариев. В каждом из них начальные условия были одинаковыми: высота, дальность и имитация длительного отказа СНС на 100 сек через 300 сек после начала автономного полета (рис. 3). До момента отказа СНС стабильно выдает эталонные измерения.

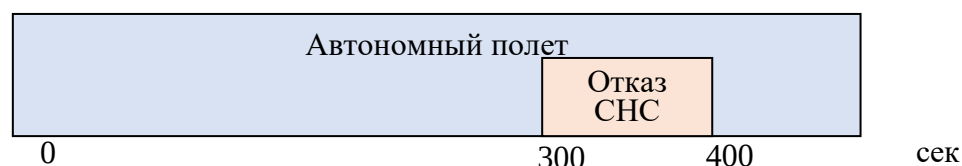


Рис. 3. Отказы СНС в сценарии

Fig. 3. Failures of the GNSS in the scenario

Сценарии различались углом ракурса (задавался от -180° до 180°) и погрешностями датчиков первичной информации, задаваемыми согласно потребительскому [9] классу точности (табл. 1). Результаты статистического моделирования (табл. 2) показывают, что чем больше нейронов на скрытом слое, тем большую точность имеет алгоритм на основе ELM. Алгоритм с 48 нейронами показал сравнимую погрешность с алгоритмом на основе EKF. Алгоритмы, имеющие 96 нейронов и более, оказались точнее алгоритма на основе EKF и имеют СКО погрешности определения местоположения на 8.3-14 % (для 96 и 386 нейронов соответственно). Однако эти тезисы справедливы для данных условий при достаточно длительном времени обучения (периода достоверной информации от СНС).

Таблица 1.
Погрешности ДПИ

Table 1.
Errors of primary information sensors

Характеристика погрешности ДПИ	Значение	Единицы измерения
Нулевой сигнал АКС	0.01	м/с ²
Нулевой сигнал ДУС	120	град/ч
Неортогональность блока АКС	2	угл.мин
Неортогональность блока ДУС	2	угл.мин
Масштабный коэффициент АКС	0.2	%
Масштабный коэффициент ДУС	0.5	%

Таблица 2.
Результаты статистического моделирования

Table 2.
Results of statistical simulation

Алгоритм	СКО ошибки определения алгоритмом параметра на конец отказа СНС						Доля испытаний, в которых ЕКФ имел сферическую ошибку по координатам больше, чем ELM [%]
	По координатам [м]			По скоростям [м/с]			
	dB	dL	dH	dVx	dVy	dVz	
EKF	32.75	32.78	0.65	0.81	0.04	0.81	–
ELM_12	49.02	49.34	1.55	1.13	0.04	1.14	5.70
ELM_24	43.13	43.29	1.59	1.01	0.04	1.02	10.18
ELM_48	33.58	33.75	1.35	0.82	0.04	0.83	45.19
ELM_96	29.98	30.10	1.22	0.75	0.04	0.76	73.55
ELM_192	30.21	30.44	1.08	0.76	0.04	0.76	76.08
ELM_384	28.07	28.27	1.07	0.71	0.04	0.72	85.61

3.2. Моделирование на КИМ

Для проведения полунатурного моделирования использовался комплекс имитационного моделирования (КИМ). Отличием данного моделирования от предыдущего в том, что алгоритмы комплексирования выполняются непосредственно на микроконтроллере. Модель измерителей и внешние воздействия моделируются на ПК. Принципиальная схема функционирования КИМ представлена на рис. 4. В качестве вычислительной платформы алгоритмов использовался микроконтроллер STM32H743VIT6 с тактовой частотой 480 МГц и объемом оперативной памяти 512 КБ.

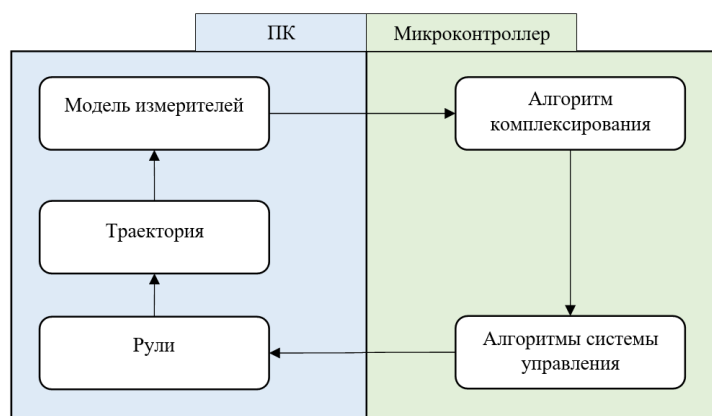


Рис. 4. Схема моделирования на КИМ

Fig. 4. Simulation scheme using hardware-in-the-loop

Сценарий моделирования был унифицирован с одним из сценариев, использованных при проведении статистического моделирования, для обеспечения сопоставимости результатов. В ходе полунатурного тестирования алгоритмов на КИМ было установлено, что доступные вычислительные ресурсы обеспечивают достаточный уровень производительности для всех исследуемых реализаций алгоритмов на основе ELM. Этот результат подтверждает эффективность проведенной модернизации алгоритма. Однако при реализации алгоритма с архитектурой скрытого слоя, содержащей 384 нейрона, оперативной памяти микроконтроллера оказалось недостаточно. Поэтому для проведения полунатурного моделирования на КИМ диапазон исследуемых ELM был ограничен количеством нейронов в скрытом слое до 192.

Результаты, представленные на рис. 5-7, демонстрируют, что алгоритмы ELM, использующие 48 и более нейронов на скрытом слое, обеспечивают точность не ниже, чем у алгоритма, реализованного на основе ЕKF. Следовательно, проведенное полунатурное моделирование подтверждает преимущество предложенного алгоритма, основанного на ELM, в задачах обработки навигационной информации при условии соблюдения ограничений по объему оперативной памяти.

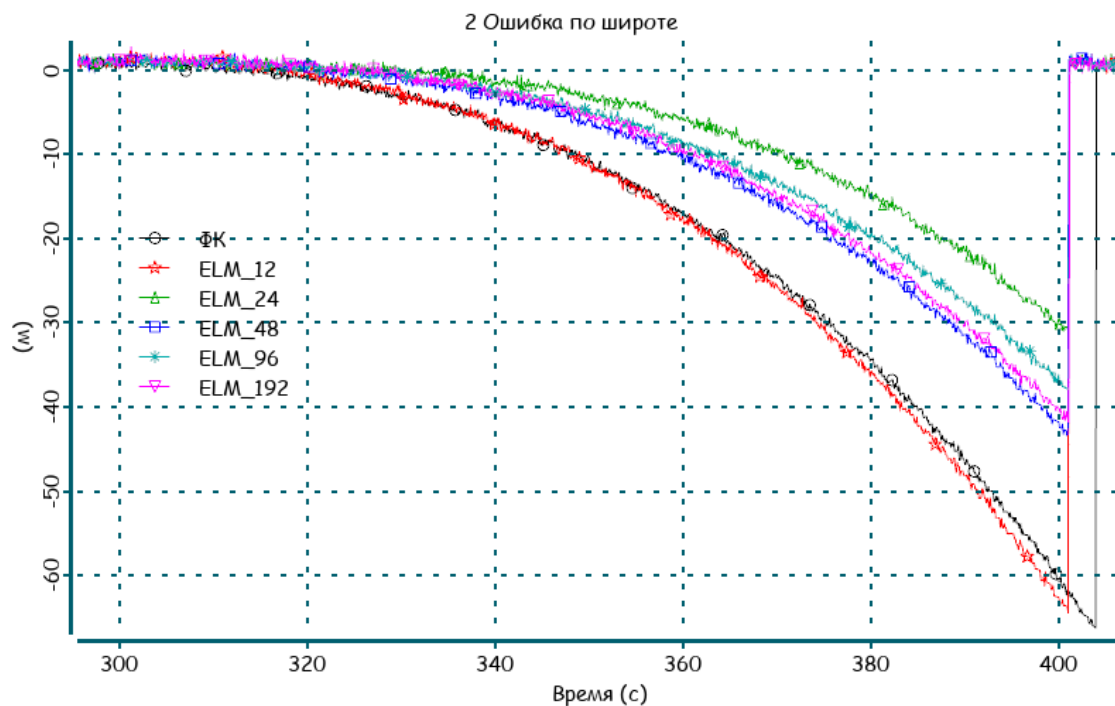


Рис. 5. Ошибки по широте алгоритмов комплексирования

Fig. 5. Latitude errors of the integration algorithms

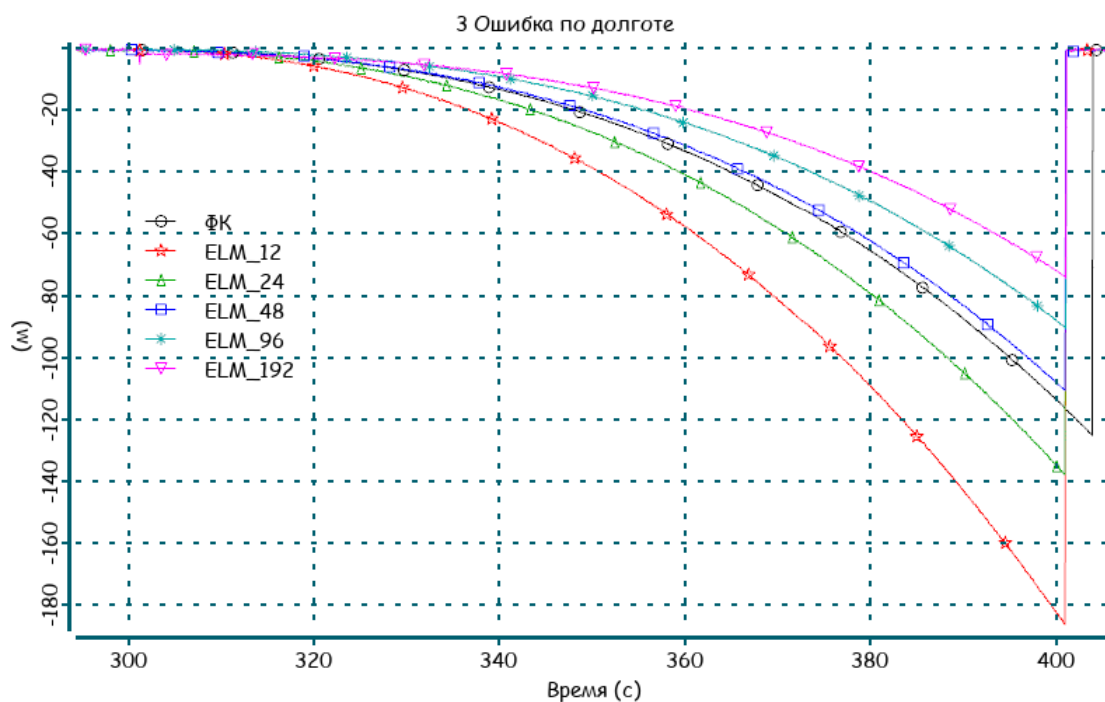


Рис. 6. Ошибки по долготе алгоритмов комплексирования

Fig. 6. Longitude errors of the integration algorithms

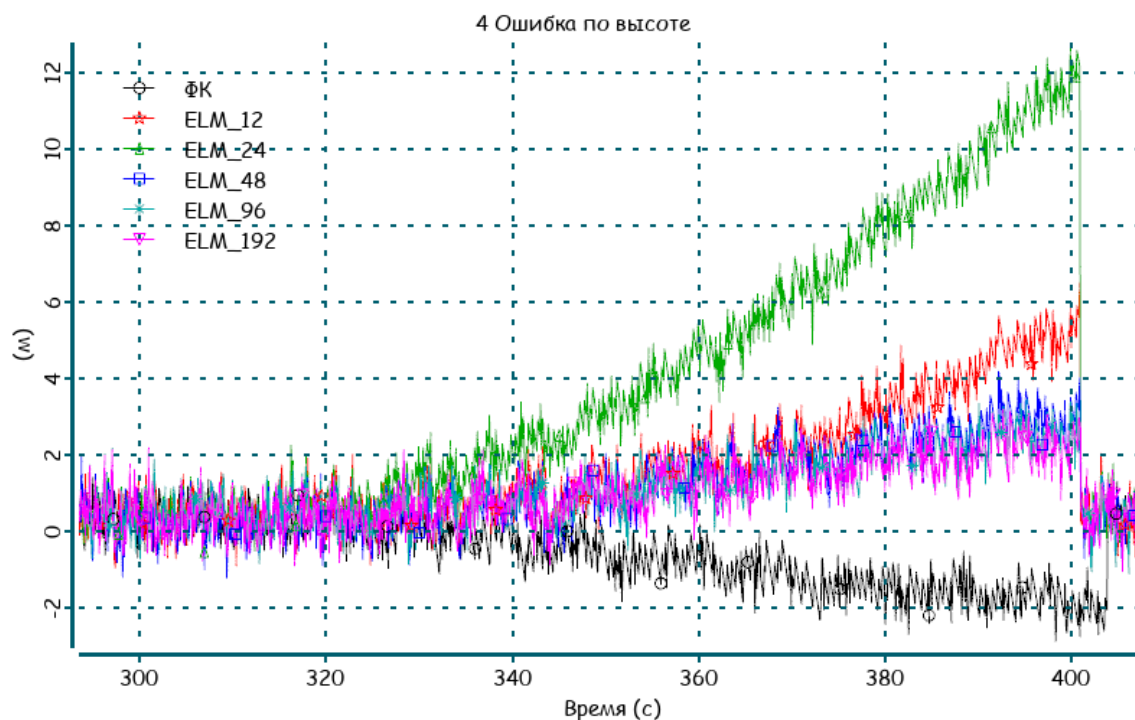


Рис. 7. Ошибки по высоте алгоритмов комплексирования

Fig. 7. Altitude errors of the integration algorithms

Заключение

Настоящая работа посвящена созданию алгоритма комплексной обработки навигационной информации, интегрирующего данные от ИНС и СНС через использование метода машинного обучения. Для адаптации под ограниченные вычислительные ресурсы была выполнена его оптимизация. Это позволило обеспечить работу системы в режиме реального времени на вычислителе, используемом на навигационных изделиях.

Проведенное статистическое моделирование, включающее сравнение алгоритмов на основе расширенного фильтра Калмана и ELM, показало, что при использовании 48 нейронов на скрытом слое точность работы алгоритмов сопоставима. Однако при увеличении количества нейронов алгоритмы на основе ELM демонстрируют более высокую точность определения навигационных параметров. Эффективность разработанного алгоритма была подтверждена в ходе полунатурного моделирования на КИМ, что свидетельствует о его практической применимости.

Следующие этапы работы планируется сосредоточить на практической проверке алгоритма в условиях, усложненных внешними факторами. Параллельно будет реализовано повышение производительности алгоритма ELM за счет более точной инициализации весов скрытого слоя и переноса основной вычислительной нагрузки на несколько временных итераций.

Библиографический список

1. **Браммер, К.** Фильтр Калмана-Бьюси / К. Браммер, Г. Зиффлинг. – М.: Наука, 1982. – 199 с.
2. **Потапов, А.С.** Технологии искусственного интеллекта / А.С. Потапов. – СПб.: СПбГУ ИТМО, 2010. – 218 с.
3. **Nilsson, N.J.** Principles of artificial intelligence / N.J. Nilsson. – Morgan Kaufmann, 2014. – 476 p.
4. **Russell, S.** Artificial intelligence: a modern approach / S. Russell, P. Norvig. – 2002. – 1132 p.
5. **Extreme learning machines** / E. Cambria, G.B. Huang, L.L.Ch. Kasun [et al] // IEEE Intelligent Systems. 2013. – Vol. 28, № 6. – P. 30-59.

6. **Брантон, С.Л.** Анализ данных в науке и технике / С.Л. Брантон, Дж.Н. Куц. – М.: ДМК Пресс, 2021. – 574 с.
7. **Noureldin A.** Optimizing neuro-fuzzy modules for data fusion of vehicular navigation systems using temporal cross-validation / A. Noureldin, A. El-Shafie, M.R. Taha // Engineering Applications of Artificial Intelligence. – 2007. – Vol. 20, iss. 1. – P. 49-61. – DOI: 10.1016/j.engappai.2006.03.002.
8. **Мэрфи, К.П.** Вероятностное машинное обучение: введение / К.П. Мэрфи. – М.: ДМК Пресс, 2022. – 940 с.
9. **Lefèvre H.C.** The fiber-optic gyroscope: The Artech House applied photonics series / H.C. Lefèvre. – 2nd ed. – Boston: Artech House, 2014. – 391 с.

References

1. **Brammer K.,** Ziffiling G. Fil'tr Kalmana-B'jusi [Kalman-Bucy filter]. Kolmanovskij V.B., translator. Moscow: Nauka; 1982. 199 p. (in Russian).
2. **Potapov A.S.** Tekhnologii iskusstvennogo intellekta [Artificial intelligence technologies]. Saint Petersburg: SPbGU ITMO; 2010. 218 p. (in Russian).
3. **Nilsson N.J.** Principles of artificial intelligence. Morgan Kaufmann; 2014. 476 p.
4. **Russell S.,** Norvig P. Artificial intelligence: a modern approach. 2002. 1132 p.
5. **Cambria E.,** Huang G.B., Kasun L.L.Ch. et al. Extreme learning machines. IEEE Intelligent Systems. 2013;28(6):30-59.
6. **Branton S.L.,** Kuts J.N. Analiz dannykh v nauke i tekhnike [Data analysis in science and engineering]. **Slinkin A.A.,** translator. Moscow: DМК Press; 2021. 574 p. (in Russian).
7. **Noureldin A.,** El-Shafie A., Taha M.R. Optimizing neuro-fuzzy modules for data fusion of vehicular navigation systems using temporal cross-validation. Engineering Applications of Artificial Intelligence. 2007;20(1):49-61. doi: 10.1016/j.engappai.2006.03.002.
8. **Murphy K.P.** Veroyatnostnoe mashinnoe obuchenie: vvedenie [Probabilistic machine learning: an introduction]. Slinkin A.A., translator. Moscow: DМК Press; 2022. 940 p. (in Russian).
9. **Lefèvre H.C.** The fiber-optic gyroscope. 2nd ed. Boston: Artech House; 2014. 391 p.

*Дата поступления
в редакцию: 28.05.2026*

*Дата принятия
к публикации: 17.07.2026*