

НЕПРЕРЫВНЫЕ ФИЛЬТРЫ ЧАСТИЦ И ИХ РЕАЛИЗАЦИЯ В РЕАЛЬНОМ МАСШТАБЕ ВРЕМЕНИ

К. А. Рыбаков, А. А. Ющенко

Московский авиационный институт (национальный исследовательский университет)

Поступила в редакцию 06.06.2018 г.

Аннотация. Основной целью является разработка и тестирование программного обеспечения для оптимального оценивания траекторий непрерывных стохастических систем по результатам измерений с помощью непрерывных фильтров частиц. Программное обеспечение разработано с применением технологии параллельного программирования OpenMP на базе среды разработки приложений Microsoft Visual Studio и пакета Intel Parallel Studio (язык программирования C/C++). Разработанное программное обеспечение реализует два варианта фильтров частиц для непрерывных систем наблюдения и оценивания. В качестве примера рассмотрена задача отслеживания координат и скоростей самолета, осуществляющего маневр в горизонтальной плоскости.

Ключевые слова: оптимальное оценивание, оптимальная фильтрация, параллельное программирование, стохастическая система, стохастические дифференциальные уравнения, фильтр частиц.

Annotation. The main goal is the development and testing of the optimal estimation software for continuous-time stochastic system trajectories. This estimation is based on the results of measurements using continuous-time particle filters. The software is developed with OpenMP, Microsoft Visual Studio and Intel Parallel Studio (C/C++ programming language). The developed software implements two variants of particle filters for continuous-time stochastic systems. The tracking for coordinates and speeds of an aircraft that executes a maneuver in the horizontal plane is considered as an example.

Keywords: optimal estimation, optimal filtering problem, parallel programming, stochastic system, stochastic differential equations, particle filter.

ВВЕДЕНИЕ

Многие задачи обработки информации в современном мире необходимо решать в реальном масштабе времени. Одна из таких задач – задача оптимальной фильтрации сигналов, а также близкие к ней задачи интерполяции и экстраполяции сигналов, возникающие при синтезе систем наблюдения и оценивания. Это накладывает дополнительные требования к алгоритмам решения перечисленных задач, а также к их программной реализации. В частности, для достижения хорошей точности получаемых оценок при программной реализации фильтров требуется использовать мощные вычислительные средства.

Статьи, в которых проводится сравнительный анализ различных алгоритмов фильтрации по точности оценивания, а иногда и по вычислительным затратам, появляются достаточно регулярно [1–6], однако они затрагивают в основном алгоритмы оптимальной фильтрации для дискретных стохастических систем. Отличие данной статьи состоит в том, что в ней рассматривается задача оптимальной фильтрации для непрерывных стохастических систем.

В качестве методов фильтрации выбраны два варианта непрерывных фильтров частиц. Первый вариант основан на вероятностном представлении решения уравнения Дункана – Мортенсена – Закаи [7, 8], он соответствует известному фильтру частиц для непрерывных стохастических систем [9]. Вторым вариантом представляет собой новый алгоритм,

предложенный в [10], он базируется на вероятностном представлении решения робастного уравнения Дункана – Мортенсена – Закаи. Оба алгоритма допускают эффективное распараллеливание. Показано, что при использовании современных вычислительных средств и технологий параллельного программирования задача оптимальной фильтрации может быть решена в реальном масштабе времени, несмотря на большой объем вычислений, характерный при применении фильтров частиц.

Ранее эти алгоритмы были апробированы на решении задачи оценивания траектории спускаемого аппарата на участке аэродинамического торможения в атмосфере Марса [11, 12], соответствующее программное обеспечение было разработано в системе компьютерной математики Mathcad. Здесь же в качестве тестовой задачи для апробации алгоритмов и нового программного обеспечения с применением технологии параллельного программирования OpenMP была выбрана задача отслеживания координат и скоростей самолета, осуществляющего маневр в горизонтальной плоскости [13–15].

1. ПОСТАНОВКА ЗАДАЧИ ФИЛЬТРАЦИИ

Задача оптимальной фильтрации сигналов для непрерывных стохастических систем состоит в оценивании траектории случайного процесса $X(t)$ по измерениям траектории случайного процесса $Y(t)$ наилучшим образом в смысле заданного критерия качества [7–9, 15–17]. При этом случайные процессы $X(t)$ и $Y(t)$ удовлетворяют системе стохастических дифференциальных уравнений, понимаемых в смысле Ито:

$$dX(t) = f(t, X(t))dt + \sigma(t, X(t))dW(t), \quad (1)$$

$$X(t_0) = X_0,$$

$$dY(t) = c(t, X(t))dt + \zeta(t)dV(t), \quad (2)$$

$$Y(t_0) = Y_0 = 0,$$

где X – n -мерный вектор, Y – m -мерный вектор, $W(t)$ и $V(t)$ – s -мерный и d -мерный независимые винеровские процессы, $f(t, x)$, $\sigma(t, x)$, $c(t, x)$ и $\zeta(t)$ – заданные векторные и

матричные функции соответствующих размеров. Распределение вектора X_0 (начальное распределение) считается заданным. Уравнение (1) – это уравнение объекта наблюдения, а уравнение (2) – уравнение измерительной системы.

Далее при записи алгоритмов решения задачи оптимальной фильтрации уравнение измерительной системы (2) удобно переписать в форме Ланжевена, т. е.

$$Z(t) = \dot{Y}(t) = c(t, X(t)) + \zeta(t)N(t), \quad (3)$$

где $N(t)$ – гауссовский белый шум, соответствующий винеровскому процессу $V(t)$. Следовательно, в задаче оптимальной фильтрации оценивание траектории $X(t)$ осуществляется и по измерениям $Z(t)$.

2. ПРИМЕНЯЕМЫЕ МЕТОДЫ И АЛГОРИТМЫ

В основе алгоритмов, описанных далее, лежит моделирование ансамбля траекторий непрерывной стохастической системы. Для первого алгоритма это исходная система (1), а для второго – вспомогательная система, математическая модель которой строится на основе модели системы (1) и коэффициентов уравнений (2) или (3). Каждой траектории ставится в соответствие весовая функция, значения которой вычисляются на основе получаемых измерений. Оценивание траектории осуществляется с помощью статистической обработки результатов моделирования, в данном случае с помощью нахождения взвешенного среднего по ансамблю траекторий, но по этим же результатам моделирования можно оценить апостериорную функцию распределения или плотность вероятности. Приближенность получаемой оценки обусловлена только тем, что для моделирования ансамбля траекторий и соответствующих весовых функций должны использоваться методы численного решения стохастических дифференциальных уравнений, а также то, что количество таких траекторий ограничено. Подобные методы получили название фильтров частиц. Поведение частицы определяется упорядоченной парой «траектория

движения + весовая функция». Более подробно фильтр частиц описан в [7, 9].

Для численного решения стохастического дифференциального уравнения в смысле Ито в приведенных ниже алгоритмах можно применить стохастический метод Эйлера [18–20]. Разностная схема стохастического метода Эйлера на примере уравнения (1) имеет вид

$$X_{k+1} = X_k + hf(t_k, X_k) + \sqrt{h}\sigma(t_k, X_k)\Delta W_k, \quad (4)$$

$$k = 0, 1, \dots,$$

где элементы вектора ΔW_k представляют собой систему независимых гауссовских случайных величин с нулевым математическим ожиданием и единичной дисперсией. Таким образом, значение вектора состояния X будем искать в дискретные моменты времени $t_k = t_0 + kh$, где h – достаточно малый шаг интегрирования (для некоторого k момент времени t_k совпадает с T).

Более точным является метод Хьюна, при этом его реализация не намного сложнее [8]:

$$X_{k+1} = X_k + \frac{h}{2}(a(t_k, X_k) + a(t_{k+1}, X_k^p)) + \frac{\sqrt{h}}{2}(\sigma(t_k, X_k) + \sigma(t_{k+1}, X_k^p))\Delta W_k, \quad (5)$$

$$X_k^p = X_k + hf(t_k, X_k) + \sqrt{h}\sigma(t_k, X_k)\Delta W_k,$$

где

$$a(t, x) = f(t, x) - \frac{1}{2} \sum_{l=1}^s \frac{\partial \sigma_{*l}(t, x)}{\partial x} \sigma_{*l}(t, x).$$

В последней формуле $\sigma_{*l}(t, x)$ – столбец матричной функции $\sigma(t, x)$ с номером l , $l = 1, 2, \dots, s$.

Если объект наблюдения описывается жесткой системой стохастических дифференциальных уравнений, то можно применить обобщенный метод типа Розенброка [18, 19]:

$$X_{k+1} = X_k + \left[E - \frac{h}{2} \frac{\partial a(t_k, X_k)}{\partial x} \right]^{-1} \times$$

$$\times \left[ha(t_k, X_k) + \sqrt{h}\sigma(t_k, X_k)\Delta W_k + \right. \quad (6)$$

$$\left. + \frac{h}{2} \frac{\partial \sigma(t_k, X_k)}{\partial x} \sigma(t_k, X_k)\Delta W_k^2 \right],$$

где E – единичная матрица порядка n ,

$$\frac{\partial \sigma(t, x)}{\partial x} \sigma(t, x)\Delta W_k^2 =$$

$$= \sum_{l,j=1}^s \frac{\partial \sigma_{*l}(t, x)}{\partial x} \sigma_{*j}(t, x)\Delta W_{kl}\Delta W_{kj}.$$

Далее с целью упрощения записи будем использовать соотношение

$$X_{k+1} = F(t_k, X_k, h),$$

в котором $F(t, X, h)$ – функция, ставящая в соответствие реализации вектора состояния $X = X_k$ в момент времени $t = t_k$ реализацию вектора состояния X_{k+1} в момент времени $t_{k+1} = t_k + h$. Эта функция определяется правой частью формул (4), (5) или (6) в зависимости от выбранного метода численного решения стохастических дифференциальных уравнений. Кроме того, введем обозначение для функции $C(t, X, Y, h)$, ставящей в соответствие реализации вектора состояния $X = X_k$ и вектору измерений $Y = Y_k$ в момент времени $t = t_k$ реализацию вектора измерений Y_{k+1} в момент времени $t_{k+1} = t_k + h$. Она необходима для моделирования измерений согласно уравнению (2), ее конкретный вид также определяется выбранным методом численного решения. В эту функцию должен входить вектор, координаты которого представляют собой систему независимых гауссовских случайных величин с нулевым математическим ожиданием и единичной дисперсией для моделирования приращений винеровского процесса $V(t)$.

АЛГОРИТМ ФИЛЬТРАЦИИ 1

Шаг 1. Задать M – число вспомогательных моделируемых траекторий (частиц); h – шаг численного интегрирования. Получить реализацию начальных векторов состояний X_0 и X_0^i согласно заданному начальному распределению, $i = 1, 2, \dots, M$.

Положить $k = 0$, $Y_0 = 0$, $\omega_0^i = 1$ (ω_k^i – весовой коэффициент), $i = 1, 2, \dots, M$.

Шаг 2. Положить

$$M_k = \sum_{i=1}^M \omega_k^i \quad (M_0 = M)$$

и найти статистические характеристики вектора состояния по реализациям $\mathbb{X}_k = \{X_k^i\}_{i=1,2,\dots,M}$

с учетом весовых коэффициентов $\mathbb{W}_k = \{\omega_k^i\}_{i=1,2,\dots,M}$: оценки вектора состояния и ковариационной матрицы ошибки оценивания по результатам измерений:

$$\hat{X}_k = \frac{1}{M_k} \sum_{i=1}^M \omega_k^i X_k^i,$$

$$\hat{R}_k = \frac{1}{M_k} \sum_{i=1}^M \omega_k^i (X_k^i - \hat{X}_k)(X_k^i - \hat{X}_k)^T.$$

Проверить условие $T - t_k = 0$. Если оно выполнено, то завершить процесс. Получить реализацию оцениваемого вектора состояния в следующем узле сетки $\{t_k\}$:

$$X_{k+1} = F(t_k, X_k, h), \quad Y_{k+1} = C(t_k, X_k, Y_k, h),$$

$$Z_k = \frac{Y_{k+1} - Y_k}{h}.$$

Шаг 3. Получить реализацию вектора состояния и весовой коэффициент в следующем узле сетки $\{t_k\}$:

$$X_{k+1}^i = F(t_k, X_k^i, h), \quad \omega_{k+1}^i = \omega_k^i e^{\mu(t_k, X_k^i, Z_k)h},$$

где

$$\mu(t, x, z) = c^T(t, x)q(t) \left(z - \frac{1}{2}c(t, x) \right),$$

$$q(t) = \eta^{-1}(t), \quad \eta(t) = \zeta(t)\zeta^T(t).$$

Шаг 4. Проверить условия:

если $i = M$, то положить $t_{k+1} = t_k + h$, $k = k + 1$ и перейти к шагу 2;

если $i < M$, то положить $i = i + 1$ и перейти к шагу 3.

Прежде чем перейти ко второму алгоритму фильтрации, обозначим через $\tilde{F}(t, \tilde{X}, Y, h)$ функцию, ставящую в соответствие реализации вектора состояния $\tilde{X} = \tilde{X}_k$ вспомогательной стохастической системы и вектору измерений $Y = Y_k$, соответствующему вектору состояния $X = X_k$ объекта наблюдения, в момент времени $t = t_k$ реализацию вектора состояния \tilde{X}_{k+1} в момент времени $t_{k+1} = t_k + h$. Вспомогательная стохастическая система отличается от (1) вектором сноса [7]:

$$\tilde{f}(t, x, y) = f(t, x) - g(t, x) \left[\frac{\partial h(t, x)}{\partial x} \right]^T y,$$

$$g(t, x) = \sigma(t, x)\sigma^T(t, x), \quad h(t, x) = q(t)c(t, x).$$

Как и для ранее введенных функций, функция $\tilde{F}(t, \tilde{X}, Y, h)$ определяется выбранным методом численного решения стохастических дифференциальных уравнений.

АЛГОРИТМ ФИЛЬТРАЦИИ 2

Шаг 1. Задать M – число вспомогательных моделируемых траекторий (частиц); h – шаг численного интегрирования. Получить реализацию начальных векторов состояний X_0 и \tilde{X}_0^i согласно заданному начальному распределению, $i = 1, 2, \dots, M$.

Положить $k = 0$, $Y_0 = 0$, $\tilde{\omega}_0^i = 1$ ($\tilde{\omega}_k^i$ – весовой коэффициент), $i = 1, 2, \dots, M$.

Шаг 2. Положить

$$M_k^* = \sum_{i=1}^M \tilde{\omega}_k^{i*} \quad (M_0^* = M),$$

где $\tilde{\omega}_k^{i*} = \tilde{\omega}_k^i e^{h^T(t_k, \tilde{X}_k^i)Y_k}$, и найти статистические характеристики вектора состояния по реализациям $\tilde{X}_k = \{\tilde{X}_k^i\}_{i=1,2,\dots,M}$ с учетом весовых коэффициентов $\tilde{\mathbb{W}}_k^* = \{\tilde{\omega}_k^{i*}\}_{i=1,2,\dots,M}$: оценки вектора состояния и ковариационной матрицы ошибки оценивания по результатам измерений:

$$\hat{X}_k = \frac{1}{M_k^*} \sum_{i=1}^M \tilde{\omega}_k^{i*} \tilde{X}_k^i,$$

$$\hat{R}_k = \frac{1}{M_k^*} \sum_{i=1}^M \tilde{\omega}_k^{i*} (\tilde{X}_k^i - \hat{X}_k)(\tilde{X}_k^i - \hat{X}_k)^T.$$

Проверить условие $T - t_k = 0$. Если оно выполнено, то завершить процесс. Получить реализацию оцениваемого вектора состояния в следующем узле сетки $\{t_k\}$:

$$X_{k+1} = F(t_k, X_k, h), \quad Y_{k+1} = C(t_k, X_k, Y_k, h).$$

Шаг 3. Получить реализацию вектора состояния вспомогательной стохастической системы и весовой коэффициент в следующем узле сетки $\{t_k\}$:

$$\tilde{X}_{k+1}^i = \tilde{F}(t_k, \tilde{X}_k^i, Y_k, h), \quad \tilde{\omega}_{k+1}^i = \tilde{\omega}_k^i e^{v(t_k, \tilde{X}_k^i, Y_k)h},$$

где

$$v(t, x, y) = -y^T \frac{\partial h(t, x)}{\partial x} f(t, x) -$$

$$-\frac{1}{2} \text{tr} \left[g(t, x) \nabla \nabla^T (y^T h(t, x)) \right] +$$

$$+\frac{1}{2} y^T \frac{\partial h(t, x)}{\partial x} g(t, x) \left[\frac{\partial h(t, x)}{\partial x} \right]^T y -$$

$$-\frac{1}{2} h^T(t, x) c(t, x) - y^T \frac{\partial h(t, x)}{\partial t}.$$

Шаг 4. Проверить условия:

если $i = M$, то положить $t_{k+1} = t_k + h$, $k = k + 1$ и перейти к шагу 2;

если $i < M$, то положить $i = i + 1$ и перейти к шагу 3.

3. ОПИСАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Разработанное программное обеспечение представляет собой консольное приложение. На первом этапе его работы моделируется траектория объекта наблюдения и результаты измерений на заданном временном промежутке, полученные данные сохраняются в текстовом файле. Далее на втором этапе эти данные обрабатываются фильтром частиц, а именно вычисляется оценка вектора состояния объекта наблюдения и дисперсия этой оценки (для каждой координаты в отдельности), которые сохраняются в текстовом файле.

Основное предназначение разработанного программного обеспечения – тестирование описанных выше алгоритмов для задач большой размерности при малом шаге численного интегрирования и большом числе моделируемых траекторий, т. е. числе частиц, что требует привлечения значительных вычислительных ресурсов.

Использование языка программирования C/C++ позволяет реализовать алгоритмы фильтров частиц на различных платформах: Windows, Unix/Linux и др. Однако тестирование проводилось только на платформе Windows.

Базовый вариант программного обеспечения разработан в среде Microsoft Visual Studio, этот вариант не использует какие-либо технологии параллельного программирования и нужен только для последующего сравнения.

Для повышения скорости производимых расчетов используется пакет Intel Parallel Studio (для разработки 64-битных приложений), а именно оптимизирующий компилятор Intel, который формирует более эффективный код программы по сравнению со стандартным компилятором из среды разработки приложений Microsoft Visual Studio. Также используется библиотека Intel Math Kernel Library, которая содержит оптимизированные многопоточные математические алгоритмы. В частности, были использованы

генераторы псевдослучайных чисел, которые эффективнее в сравнении со стандартной функцией `rand()`.

Кроме того, чтобы повысить скорость работы программы, используется технология параллельного программирования OpenMP для распараллеливания циклов, а для полной загрузки ядер процессора применяются функции из стандартной библиотеки `tread`, дополнительно разбивающие фильтр частиц на потоки. Для лучшей оптимизации программы выходные данные записываются в текстовый файл только после всех проводимых вычислений.

4. ТЕСТОВАЯ ЗАДАЧА И РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Задача отслеживания координат и скоростей самолета, осуществляющего маневр в горизонтальной плоскости [13, 14], имеет следующие параметры $X = [\varepsilon, \eta, \zeta, \dot{\varepsilon}, \dot{\eta}, \dot{\zeta}, \omega]^T$, где ε, η, ζ – координаты самолета в декартовой системе координат, $\dot{\varepsilon}, \dot{\eta}, \dot{\zeta}$ – значения скоростей в направлении соответствующих координатных векторов, а ω – угловая скорость самолета; размерность вектора состояния равна семи ($n = 7$). Их изменение описывается уравнением (зависимость от t для краткости опущена), $t \in [0, 60 \text{ с}]$:

$$d \begin{pmatrix} \varepsilon \\ \dot{\varepsilon} \\ \eta \\ \dot{\eta} \\ \zeta \\ \dot{\zeta} \\ \omega \end{pmatrix} = \begin{pmatrix} \dot{\varepsilon} \\ -\omega \dot{\eta} \\ \dot{\eta} \\ \omega \dot{\varepsilon} \\ \dot{\zeta} \\ 0 \\ 0 \end{pmatrix} dt + \begin{pmatrix} 0 & 0 & 0 & 0 \\ \sigma_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \sigma_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_1 & 0 \\ 0 & 0 & 0 & \sigma_2 \end{pmatrix} dW$$

(объект наблюдения),

в котором $\sigma_1 = \sqrt{0.2}$, $\sigma_2 = 0.007$; размерность $W(t)$ равна четырем ($s = 4$). Числовые параметры σ_1 и σ_2 описывают влияние непредсказуемых факторов на движение самолета: турбулентность, порывы ветра и т. п.

Начальный вектор состояния (неслучайный):

$$X_0 = [1000 \text{ м}, 0 \text{ м/с}, 2650 \text{ м}, 150 \text{ м/с}, 200 \text{ м}, 0 \text{ м/с}, 6^\circ/\text{с}]^T.$$

Непосредственному наблюдению доступны величины $Z = [r, \theta, \phi]^T$:

$$\begin{pmatrix} r \\ \theta \\ \phi \end{pmatrix} = \begin{pmatrix} \sqrt{\varepsilon^2 + \eta^2 + \zeta^2} \\ \arctg(\eta / \varepsilon) + \gamma(\varepsilon, \eta) \\ \arctg(\zeta / \sqrt{\varepsilon^2 + \eta^2}) \end{pmatrix} + \begin{pmatrix} \sigma_r & 0 & 0 \\ 0 & \sigma_\theta & 0 \\ 0 & 0 & \sigma_\phi \end{pmatrix} N$$

(измерительная система),

где $\sigma_r = 50$ м, $\sigma_\theta = 0.1^\circ$, $\sigma_\phi = 0.1^\circ$; размерность вектора измерений, а также размерность $V(t)$ и $N(t)$ равна трем ($m = d = 3$). Здесь r – дистанция до самолета от начала координат, в котором расположен радар, θ – азимут, ϕ – угол места (угловая высота наблюдаемого объекта). Функция $\gamma(\varepsilon, \eta)$ определяется следующим образом:

$$\gamma(\varepsilon, \eta) = \begin{cases} 0, & \varepsilon \geq 0, \eta \geq 0, \\ \pi, & \varepsilon < 0, \\ 2\pi, & \varepsilon \geq 0, \eta < 0. \end{cases}$$

Критерий качества при оценивании – критерий минимума среднеквадратической ошибки (ошибка – расстояние между истинной траекторией и ее оценкой в каждый момент времени на заданном промежутке). По сравнению с работами [13, 14] в модель системы наблюдения внесены некоторые коррективы. Во-первых, изменена матрица $\sigma(t, x)$ в уравнении объекта наблюдения. В [13, 14] это квадратная диагональная матрица размера 7×7 :

$$\sigma(t, x) = \text{diag}(0, \sigma_1, 0, \sigma_1, 0, \sigma_1, \sigma_2),$$

однако наличие нулевых столбцов в ней избыточно, оно приведет к лишним вычислениям, связанным с получением псевдослучайной последовательности при моделировании траекторий объекта наблюдения. Поэтому в этой работе размер матрицы $\sigma(t, x)$ составляет 7×4 и она не содержит нулевых столбцов. Во-вторых, в уравнение измерительной системы добавлена функция $\gamma(\varepsilon, \eta)$, обеспечивающая непрерывность величины θ и диапазон ее изменения $[0, 2\pi]$. Без этой функции угол $\arctg(\eta / \varepsilon)$ меняется скачком при переходе точки (ε, η) из первого квадранта во второй квадрант и из третьего квадранта в четвертый квадрант на плоскости.

Тестирование разработанного программного обеспечения на задаче отслеживания ко-

ординат и скоростей самолета, осуществляющего маневр в горизонтальной плоскости, проводилось на процессорах Intel семейств Core i3, Core i5 и Core i7. Это 2-ядерный Intel Core i3 2120 (3.3 ГГц, 4 потока), 4-ядерный Intel Core i5 3550 (3.70 ГГц, 4 потока) и 6-ядерный Intel Core i7 4930К (3.9 ГГц, 12 потоков). Разработанное программное обеспечение наиболее эффективно при числе потоков, кратном числу поддерживаемых потоков процессора. Результаты тестирования приведены в табл. 1 и 2. Табл. 1 содержит время расчетов в секундах по алгоритму 1 для числа моделируемых траекторий (частиц) $M = 30000$ и $M = 3000000$ при шаге численного интегрирования $h = 0.01$ (применялся стохастический метод Эйлера). В табл. 2 приведено время расчетов в секундах по алгоритму 2 для того же числа моделируемых траекторий при том же шаге численного интегрирования. Из полученных данных можно сделать вывод, что реализованные алгоритмы 1 и 2 с применением пакета Intel Parallel Studio и оптимизирующего компилятора Intel C/C++ позволяют обрабатывать данные и вычислять оптимальную оценку в реальном масштабе времени для числа частиц $M = 30000$, которое обеспечивает сопоставимую погрешность метода статистического моделирования (эта погрешность обратно пропорциональна квадратному корню из числа M) по сравнению с погрешностью численного метода. При этом есть запас по времени, особенно для процессора Intel Core i7 4930К, позволяющий уменьшить шаг численного интегрирования либо использовать при расчетах метод численного решения стохастических дифференциальных уравнений с более высоким порядком сходимости. Для числа частиц $M = 3000000$ расчеты в однопоточной конфигурации не проводились, в соответствующих строках таблиц содержатся прочерки.

Существенное увеличение числа частиц до $M = 3000000$ значительно увеличивает время расчетов (более чем в сто раз), при этом увеличение точности не будет таким существенным. Погрешность метода статистического моделирования уменьшается только в десять раз, кроме того, при таком числе ча-

Таблица 1

Время расчетов по алгоритму 1, $M=30000 / M=3000000$, $h=0.01$

Число потоков	Intel Core i3 2120	Intel Core i5 3550	Intel Core i7 4930K
1	153.702 / -----	129.995 / -----	117.171 / -----
2	20.410 / 4563.371	10.966 / 3177.975	9.875 / 2929.561
4	18.505 / 3133.739	7.488 / 1526.838	5.366 / 1378.075
6	17.595 / 3071.551	5.616 / 1589.315	3.386 / 1052.534
8	15.731 / 2929.913	5.554 / 1464.858	3.463 / 953.988
12	11.174 / 2714.224	5.694 / 1352.429	2.667 / 929.777
16	10.502 / 2652.522	5.632 / 1310.761	2.793 / 872.089
24	10.487 / 2363.743	5.740 / 1035.249	2.621 / 784.244

Таблица 2

Время расчетов по алгоритму 2, $M=30000 / M=3000000$, $h=0.01$

Число потоков	Intel Core i3 2120	Intel Core i5 3550	Intel Core i7 4930K
1	273.854 / -----	234.094 / -----	210.897 / -----
2	38.406 / 7400.799	25.506 / 5366.456	22.885 / 5172.407
4	32.023 / 5079.623	14.867 / 2665.030	12.808 / 2414.635
6	31.149 / 4645.909	13.197 / 2285.982	9.719 / 1556.009
8	30.368 / 4550.625	13.385 / 2245.187	8.361 / 1368.965
12	28.325 / 4251.441	13.088 / 2205.391	6.896 / 1149.706
16	27.107 / 4146.297	13.385 / 2156.954	6.864 / 1093.546
24	27.154 / 3510.442	13.915 / 1560.580	6.458 / 939.699

стиц имеет смысл на порядок или на два порядка уменьшить шаг численного интегрирования для уравнивания статистической погрешности и погрешности численного метода, что повлечет пропорциональное увеличение времени расчетов. На два порядка следует уменьшить шаг численного интегрирования при использовании стохастического метода Эйлера (порядок сходимости 0.5), для уменьшения шага численного интегрирования на порядок может оказаться достаточным применение метода Хьюна или обобщенного метода типа Розенброка (максимальный порядок сходимости 1.0).

Начиная с числа потоков, равного двенадцати, время расчетов как по алгоритму 1, так и по алгоритму 2 стабилизируется. Это время обратно пропорционально числу ядер процессора. Разумеется, на скорость вычислений оказывает влияние не только число ядер используемого процессора, но и его тактовая частота, размер кэша, скорость

обмена данных с оперативной памятью и т.п. Детальное исследование влияния перечисленных факторов в этой работе не проводилось. Кроме того, не проводилось сравнение с обобщенным фильтром Калмана – Бьюси или его модификациями [8, 15–17], например, сигма-точечным фильтром Калмана – Бьюси [21], поскольку очевидно, что время расчетов при их применении будет существенно (кратно) меньше. В этом их преимущество, однако такие алгоритмы фильтрации – субоптимальные, оптимальные только в классе линеаризованных стохастических систем. В отличие от фильтров частиц, которые применяются непосредственно к нелинейным системам.

ЗАКЛЮЧЕНИЕ

В ходе проведенных исследований разработано и апробировано программное обеспечение для оптимального оценивания траекторий непрерывных стохастических систем

по результатам измерений с помощью непрерывных фильтров частиц. Основные инструменты разработки – язык программирования C/C++, технология параллельного программирования OpenMP, среда разработки приложений Microsoft Visual Studio и пакет Intel Parallel Studio. Разработанное программное обеспечение реализует два варианта фильтров частиц для непрерывных систем наблюдения и оценивания. В качестве примера рассмотрена задача отслеживания координат и скоростей самолета, осуществляющего маневр в горизонтальной плоскости. На этом примере продемонстрирована возможность решения задачи оптимальной фильтрации в реальном масштабе времени.

Работа выполнена при финансовой поддержке РФФИ (проект № 17-08-00530-а).

СПИСОК ЛИТЕРАТУРЫ

1. Сравнительный анализ среднеквадратической погрешности определения координат объекта в бесплатформенной инерциальной навигационной системе при использовании различных алгоритмов нелинейной фильтрации / А. С. Конаков [и др.] // Доклады ТУСУР. – 2012. – № 1 (25), ч. 1. – С. 5–9.
2. Сопоставление методов фильтрации в задачах статистической регуляризации при оценивании параметров радиолокационных систем / А. А. Сытник [и др.] // Вестник Воронеж. гос. ун-та. Сер. Системный анализ и информационные технологии. – 2013. – № 1. – С. 10–16.
3. Сопоставительный анализ ансамблевого и расширенного фильтров Калмана при вторичной обработке информации в спутниковых радионавигационных системах / А. Н. Мороз [и др.] // Доклады БГУИР. – 2014. – № 4 (82). – С. 66–72.
4. Солонар, А. С. Особенности применения методов дискретной фильтрации для задач сопровождения баллистических объектов, совершающих спиральный маневр / А. С. Солонар, П. А. Хмарский // Доклады БГУИР. – 2015. – № 1 (87). – С. 71–77.
5. Кудрявцева, И. А. Анализ эффективности расширенного фильтра Калмана, сигма-точечного фильтра Калмана и сигма-точечного фильтра частиц / И. А. Кудрявцева // Научный вестник МГТУ ГА. – 2016. – № 224 (2). – С. 43–51.
6. Куликова, М. В. Численно устойчивые реализации фильтра Калмана для оценивания линейных парных марковских моделей с гауссовым шумом / М. В. Куликова, Ю. В. Цыганова // Вычислительные технологии. – 2017. – Т. 22, № 3. – С. 45–60.
7. Рыбаков, К. А. Статистические методы анализа и фильтрации в непрерывных стохастических системах / К. А. Рыбаков. – Москва : Изд-во МАИ, 2017. – 176 с.
8. Методы описания, анализа и синтеза нелинейных систем управления / В. В. Семенов [и др.]. – Москва : Изд-во МАИ, 1993. – 312 с.
9. Bain, A. Fundamentals of Stochastic Filtering / A. Bain, D. Crisan. – Springer, 2009. – 404 p.
10. Rybakov, K. A. Robust Duncan – Mortensen – Zakai equation for non-stationary stochastic systems / K. A. Rybakov // Proceedings of the 2017 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON) (Novosibirsk Akademgorodok, Russia, September 18–22, 2017). – IEEE, 2017. – P. 151–154.
11. Руденко, Е. А. Оптимальный нелинейный рекуррентный фильтр с конечной памятью / Е. А. Руденко // Известия РАН. Теория и системы управления. – 2018. – № 1. – С. 45–63.
12. Рыбаков, К. А. Применение фильтра частиц в задаче оценивания траектории спускаемого аппарата при аэродинамическом торможении / К. А. Рыбаков // Актуальные проблемы прикладной математики, информатики и механики : сб. тр. Междунар. науч.-техн. конф. (Воронеж, 18–20 декабря 2017 г.). – Воронеж: Изд-во «Научно-исследовательские публикации», 2017. – С. 863–872.
13. Куликова, М. В. Численные методы нелинейной фильтрации для обработки сигналов и измерений / М. В. Куликова, Г. Ю. Куликов // Вычислительные технологии. – 2016. – Т. 21, № 4. – С. 64–98.
14. Arasaratnam, I. Cubature Kalman filtering for continuous-discrete systems: theo-

ry and simulations / I. Arasaratnam, S. Haykin, T. R. Hurd // IEEE Transactions on Signal Processing. – 2010. – V. 58, no. 10. – P. 4977–4993.

15. *Bar-Shalom, Y.* Estimation with Applications to Tracking and Navigation / Y. Bar-Shalom, X. R. Li, T. Kirubarajan. – John Wiley & Sons, 2001. – 580 p.

16. *Синицын, И. Н.* Фильтры Калмана и Пугачева / И. Н. Синицын. – Москва : Логос, 2007. – 776 с.

17. *Степанов, О. А.* Применение теории нелинейной фильтрации в задачах обработки навигационной информации / О. А. Степанов. – Санкт-Петербург : ЦНИИ «Электро-прибор», 2003. – 370 с.

18. *Аверина, Т. А.* Верификация численных методов решения систем со случайной

структурой / Т. А. Аверина. – Новосибирск : РИЦ НГУ, 2015. – 178 с.

19. *Аверина, Т. А.* Построение алгоритмов статистического моделирования систем со случайной структурой / Т. А. Аверина. – Новосибирск : РИЦ НГУ, 2015. – 155 с.

20. *Кузнецов, Д. Ф.* Стохастические дифференциальные уравнения: теория и практика численного решения. С программами в среде Matlab / Д. Ф. Кузнецов // Дифференциальные уравнения и процессы управления. – 2017. – № 2.

21. *Särkkä, S.* On unscented Kalman filtering for state estimation of continuous-time nonlinear systems / S. Särkkä // IEEE Transactions on Automatic Control. – 2007. – V. 52, No. 9. – P. 1631–1641.

Рыбаков Константин Александрович – канд. физ.-мат. наук, доцент кафедры «Математическая кибернетика», Московский авиационный институт (национальный исследовательский университет).

E-mail: rkoffice@mail.ru

Rybakov Konstantin – candidate of physico-mathematical sciences, associate professor, Mathematical cybernetics department, Moscow aviation institute (national research university).

E-mail: rkoffice@mail.ru

Ющенко Артем Анатольевич – магистрант кафедры «Математическая кибернетика», Московский авиационный институт (национальный исследовательский университет).

E-mail: artem55_55@mail.ru.

Yushchenko Artem – graduate student, Mathematical cybernetics department, Moscow aviation institute (national research university).

E-mail: rkoffice@mail.ru