

СРАВНЕНИЕ НЕЛИНЕЙНЫХ ФИЛЬТРОВ КАЛМАНА ПРИМЕНительно К ЗАДАЧЕ ТРАЕКТОРНОЙ ОБРАБОТКИ

Соболев А.А.
Москва, Россия

Аннотация — В статье рассматриваются нелинейные фильтры — расширенный фильтр Калмана (РФК, Extended Kalman Filter — EKF), сигма-точечный (ансцентный) фильтр Калмана (СТФК, Unscented Kalman Filter — UKF), кубатурный фильтр Калмана (КФК, Cubature Kalman Filter — CKF), их квадратно-корневые версии (КК-РФК, Square Root EKF — SREKF; КК-СТФК, Square Root UKF — SRUKF; КК-КФК, Square-Root CKF — SRCKF), гибридные фильтры — расширенно-сигма-точечный (РСТФК, Extended-Unscented Kalman Filter — EUKF), расширенно-кубатурный (РКФК, Extended-Cubature Kalman Filter — ECKF), а также их квадратно-корневые версии (КК-РСТФК, Square Root EUKF — SREUKF; КК-РКФК, Square Root ECKF — SRECKF) применительно к задаче траекторной обработки. Приводится сравнение среднеквадратичной ошибки (Root Mean Square Error — RMSE) рассмотренных фильтров, а также сравнение скорости их работы относительно РФК.

Ключевые слова — фильтр Калмана; нелинейная фильтрация; траекторная обработка.

ВВЕДЕНИЕ

Траекторная обработка — это совокупность методов и средств, предназначенных для получения оценки траектории объекта путем обработки потока предварительных решений об обнаружении объекта (отметок). Под траекторией движущегося объекта понимается зависимость положения объекта от времени в принятой системе координат [1]. Положение объекта при этом характеризуется вектором состояния, содержащим координаты, вектор скорости и т.д.

В алгоритмах траекторной обработки могут применяться такие методы, как метод наименьших квадратов (МНК), метод максимального правдоподобия (МП), метод максимума апостериорной вероятности (МАВ), метод минимума среднеквадратического отклонения (МСКО) и другие. Однако наибольшее распространение получили методы рекуррентной оптимальной фильтрации на основе различных модификаций фильтра Калмана, который является оптимальным по критерию МСКО [1].

При этом наиболее широко используются квазилинейные и нелинейные фильтры Калмана, что обусловлено нелинейностью уравнений движения объектов. В первом случае используется разложение

нелинейных уравнений в ряд Тейлора, из которого используется только первый член. Такой фильтр получил название расширенный (локально линеаризованный) фильтр Калмана (РФК, Extended Kalman Filter — EKF) [2].

Недостатки РФК во многом обусловлены локальной линеаризацией нелинейных функций, поэтому был предложен ряд фильтров, основанных на взвешенной статистической линейной регрессии. В этих фильтрах, получивших название сигма-точечных (ансцентных) фильтров Калмана (СТФК, Unscented Kalman Filter — UKF), среднее $\hat{\mathbf{x}}$ и ковариация $\mathbf{P}_{\mathbf{x}}$ вектора состояния \mathbf{x} объекта размерности n оценивается при помощи набора сигма-точек \mathcal{X}_i , выбранных специальным образом [1,3]:

$$\frac{1}{N} \sum_{i=1}^N \mathcal{X}_i = \hat{\mathbf{x}}; \quad (1)$$

$$\frac{1}{N} \sum_{i=1}^N (\mathcal{X}_i - \hat{\mathbf{x}})(\mathcal{X}_i - \hat{\mathbf{x}})^T = \mathbf{P}_{\mathbf{x}}, \quad (2)$$

где i — индекс сигма-точки, N — их количество. Далее сигма-точки \mathcal{X}_i пространства \mathbf{x} подвергаются нелинейному преобразованию в сигма-точки \mathcal{Y}_i пространства измерений \mathbf{y} размерности m для оценки среднего $\hat{\mathbf{y}}$ и ковариации $\mathbf{P}_{\mathbf{y}}$ вектора прогноза измерений аналогично (1) и (2), дополнительно применяя взвешивание весами W_i :

$$\mathcal{Y}_i = \mathcal{H}(\mathcal{X}_i); \quad (3)$$

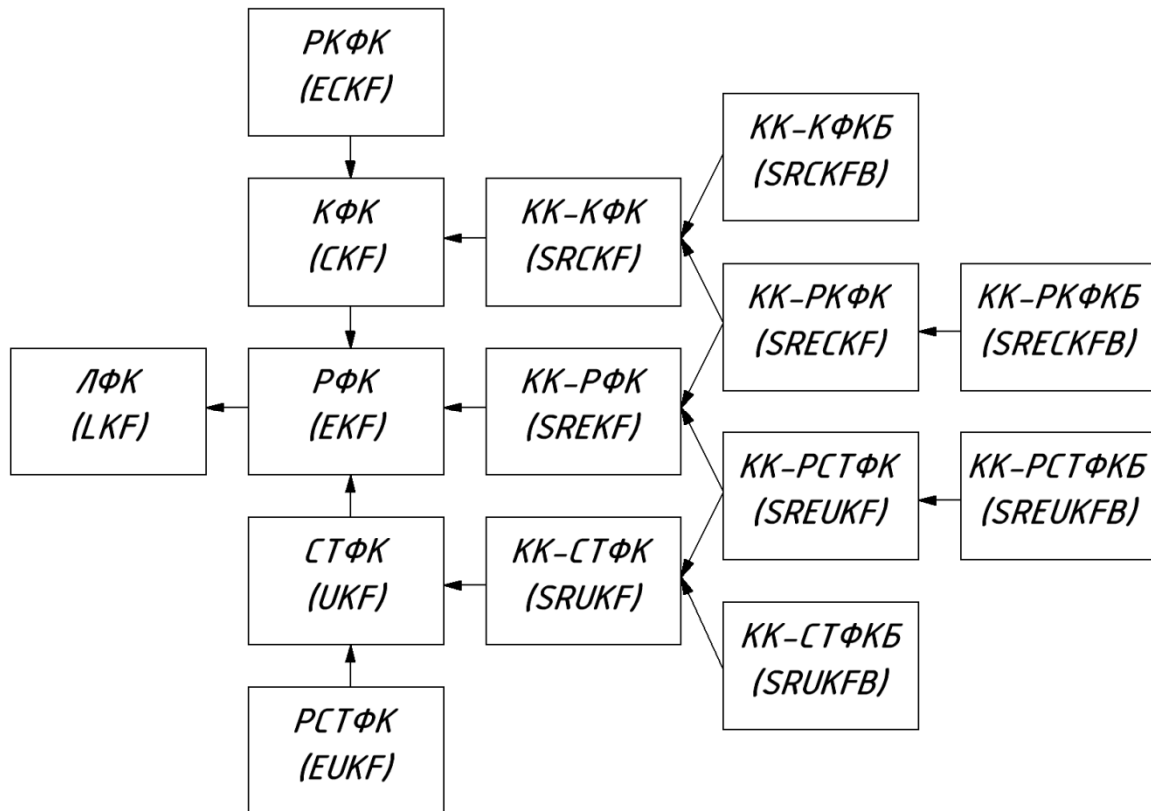
$$\hat{\mathbf{y}} = \sum_{i=1}^N W_i \mathcal{Y}_i; \quad (4)$$

$$\mathbf{P}_{\mathbf{y}} = \sum_{i=1}^N W_i (\mathcal{Y}_i - \hat{\mathbf{y}})(\mathcal{Y}_i - \hat{\mathbf{y}})^T, \quad (5)$$

где \mathcal{H} — функционал перевода вектора состояния \mathbf{x} в измерение \mathbf{y} .

При таком способе оценки результирующая ошибка будет ниже, чем при линеаризации усеченным рядом Тейлора в РФК [1]. На рис.1 далее представлена полная схема наследования фильтров Калмана.

Рис.1. Полная схема иерархии наследования фильтров Калмана



Рассмотрим схему, представленную на рис. 1, с точки зрения программной реализации на языке C++. Программная реализация сделана максимально универсальной и пригодной для широкого спектра задач с учетом всех необходимых действий, которые могут понадобиться при выполнении одного такта обработки (экстраполяции и коррекции по Калману). Здесь главным, родительским классом, является класс линейного фильтра Калмана, содержащий в себе все матрицы линейного фильтра, их размерности, методы доступа к полям класса – так называемые геттеры/сеттеры (getters/setters), обертки функций различных проверок (проверки нахождения вектора состояния в области определения и т.д.), методы проверки и исправления симметричности полных ковариационных матриц, проверка на положительную определенность матрицы и т.д. Размерности матриц являются шаблонными параметрами фильтра (template) для удобства задания. Основными методами класса являются методы экстраполяции (Predict) и коррекции (Correct), которые сделаны виртуальными для последующего универсального переопределения в классах-потомках для удобства использования – имена методов при замене фильтра остаются неизменными, что позволяет легко менять фильтры в исходном коде без существенных затрат.

Следующим звеном иерархии является РФК, отнаследованный от ЛФК. Класс РФК дополняется обертками функционалов экстраполяции \mathcal{F} , перевода состояния в измерение \mathcal{H} и функционалов соответствующих матриц \mathbf{F} и \mathbf{H} .

Дальнейшая иерархия на рис.1 делится на иерархию кубатурных (верхняя часть рисунка) и сигма-точечных (нижняя часть рисунка) фильтров, различия между которыми обусловлены разным количеством сигма-точек и способом работы с ними (будет рассмотрено далее в последующих разделах).

Из рис. 1 видно, что в иерархии неизбежна ситуация ромбовидного наследования (diamond inheritance), разрешаемая при программной реализации в C++ путем использования виртуального наследования классов фильтров КФК (CKF), СТФК (UKF) и КК-РФК (SREKF). При этом должна быть переопределена логика конструкторов копирования (copy constructor), оператора присвоения (=), конструкторов перемещения (move constructor), методов свапа (swap) (при использовании идеологии copy & swap) иерархии фильтров для возможного последующего использования классов внутри контейнеров стандартной библиотеки std (например, std::vector, std::map и др.).

В данной статье приводится алгоритм расширенного фильтра Калмана (раздел I), сигма-точечного фильтра (раздел II), возникающие проблемы при реализации сигма-точечного фильтра (раздел III); дальнейшие разделы посвящены устранению проблем сигма-точечного фильтра (разделы IV – V), оптимизации производительности (разделы VI – VII) и экспериментальной проверке рассмотренных фильтров (раздел VIII). Выводы представлены в разделе IX.

Целью данной статьи является сравнение указанных фильтров Калмана по критерию минимума среднеквадратичной ошибки оценки вектора состояния (RMSE), а также скорости их работы относительно РФК, ставшего за многие годы эталоном в задачах траекторной обработки.

I. РАСШИРЕННЫЙ ФИЛЬТР КАЛМАНА

Нелинейные модели состояния и измерений представляются в виде:

$$\mathbf{x}_k = \mathcal{F}(\mathbf{x}_{k-1}, \Delta t) + \mathbf{w}_k; \quad (6)$$

$$\mathbf{y}_k = \mathcal{H}(\mathbf{x}_k) + \mathbf{v}_k, \quad (7)$$

где k – индекс временного такта, $\Delta t = t_k - t_{k-1}$ – интервал экстраполяции, \mathcal{F} – функционал экстраполяции (модель движения объекта), \mathcal{H} – функционал перевода вектора состояния \mathbf{x} в измерение \mathbf{y} , \mathbf{w} и \mathbf{v} – независимые аддитивные случайные величины с нулевым средним и ковариационными матрицами \mathbf{Q} и \mathbf{R} соответственно.

Вводятся матрицы первых частных производных [1,2]:

$$\mathbf{F} = \left. \frac{\partial \mathcal{F}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_0}, \quad \mathbf{H} = \left. \frac{\partial \mathcal{H}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_0}. \quad (8)$$

При этом предполагается, что в заданной точке \mathbf{x}_0 существуют все частные производные первого порядка. С учетом введенных выше обозначений уравнения РФК могут быть записаны в виде алгоритма, представленного ниже.

Алгоритм 1: РФК (EKF)

1) Инициализация:

$$\hat{\mathbf{x}}(t_0) = \mathbb{E}[\mathbf{x}(t_0)] = \hat{\mathbf{x}}_0; \quad (9)$$

$$\mathbf{P}(t_0) = \mathbb{E}[(\mathbf{x}(t_0) - \hat{\mathbf{x}}_0)(\mathbf{x}(t_0) - \hat{\mathbf{x}}_0)^T] = \mathbf{P}_0; \quad (10)$$

$$\mathbf{Q}_k = \mathbb{E}[\mathbf{w}_k \mathbf{w}_k^T] = \mathbf{Q}; \quad (11)$$

$$\mathbf{R}_k = \mathbb{E}[\mathbf{v}_k \mathbf{v}_k^T] = \mathbf{R}, \quad (12)$$

где $\mathbb{E}\{\cdot\}$ – математическое ожидание, $k \in (1, \dots, \infty)$, ковариационная матрица шумов состояния \mathbf{Q} задается приведенной к интервалу экстраполяции 1 секунда.

2) Экстраполяция:

$$\hat{\mathbf{x}}_{k|k-1} = \mathcal{F}(\hat{\mathbf{x}}_{k-1}, \Delta t); \quad (13)$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^T + \mathbf{Q} \Delta t; \quad (14)$$

$$\hat{\mathbf{y}}_{k|k-1} = \mathcal{H}(\hat{\mathbf{x}}_{k|k-1}), \quad (15)$$

где $\mathbf{Q} \Delta t$ – приведение ковариационной матрицы шумов состояния \mathbf{Q} к интервалу экстраполяции Δt путем умножения.

3) Коррекция:

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}; \quad (16)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1}; \quad (17)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}); \quad (18)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}, \quad (19)$$

где \mathbf{S} – ковариационная матрица экстраполированной оценки вектора измерений $\hat{\mathbf{y}}_{k|k-1}$, \mathbf{K} – коэффициент усиления фильтра, \mathbf{y}_k – вектор измерений текущего k -го такта, \mathbf{I} – единичная матрица.

II. СИГМА-ТОЧЕЧНЫЙ ФИЛЬТР КАЛМАНА

Если в РФК используется локальная линеаризация нелинейных функций в одной точке, то в СТФК используется специальная аппроксимация случайной величины, получившая название ансцентное преобразование (Unscented Transformation – UT). Существует несколько вариантов ансцентного преобразования, различающихся способом выбора сигма-точек, их количеством, а также весами среднего и ковариации. [1,3,4].

Рассмотрим далее 3 наиболее широко используемых способа.

A. Базовое ансцентное преобразование

Базовый вариант [3], названный в [5] Base Set, использует $2n$ сигма-точек. Задается матрица сигма-точек \mathcal{X} размерности $[n, 2n]$ (сигма-точки – столбцы), а также массив весов среднего (m) и ковариации (c) $W_i^{(m)} = W_i^{(c)} = W_i$ следующим образом:

$$\begin{aligned} \mathcal{X}_i &= \hat{\mathbf{x}} + \gamma \sqrt{\mathbf{P}}_i, & i &= 1, \dots, n; \\ \mathcal{X}_i &= \hat{\mathbf{x}} - \gamma \sqrt{\mathbf{P}}_{i-n}, & i &= n+1, \dots, 2n; \\ W_i &= 1/(2n), & i &= 1, \dots, 2n, \end{aligned} \quad (20)$$

где i – индекс сигма-точки, n – размерность вектора состояния, $\gamma = \sqrt{n}$, $\sqrt{\mathbf{P}}_i$ – i -й столбец квадратного корня матрицы \mathbf{P} . При этом матрицей квадратного корня матрицы \mathbf{P} называется такая матрица \mathbf{A} , для которой выполняется $\mathbf{A} \mathbf{A}^T = \mathbf{P}$. Практически это, как правило, вычисляется при помощи нижнетреугольного разложения Холецкого: $\mathbf{A} = \sqrt{\mathbf{P}} = \text{chol}(\mathbf{P}, 'lower')$, где обозначение chol относится к MATLAB/Octave/Armadillo C++ (обозначение 'lower' нижнетреугольного разложения далее по тексту опускается).

B. Обобщенное ансцентное преобразование

В данном случае используется набор $2n+1$ сигма-точек – в качестве дополнительной точки («нулевой» точки, имеющей индекс $i=0$) задается сам вектор состояния [1,3]:

$$\begin{aligned}
\mathcal{X}_i &= \hat{\mathbf{x}}, & i &= 0; \\
\mathcal{X}_i &= \hat{\mathbf{x}} + \gamma\sqrt{\mathbf{P}}_i, & i &= 1, \dots, n; \\
\mathcal{X}_i &= \hat{\mathbf{x}} - \gamma\sqrt{\mathbf{P}}_{i-n}, & i &= n+1, \dots, 2n; \\
W_i &= w0, & i &= 0; \\
W_i &= (1-w0)/2n, & i &= 1, \dots, 2n,
\end{aligned} \quad (21)$$

где $\gamma = \sqrt{n/(1-w0)}$, $w0$ – параметр настройки, определяющий разброс между центральной и i -й точкой. Такой способ в [5] назван Gauss Set (при отсутствии ограничений на $w0$) или Mean Set (при неотрицательных значениях $w0$).

С. Масштабируемое ансцентное преобразование

Данный способ, введенный в [3], обобщенный в [4] и названный в [5] Scaled Set, также использует набор из $2n+1$ сигма-точек и отличается отдельным формированием весов для вычисления среднего $W_i^{(m)}$ и ковариации $W_i^{(c)}$. Параметры разброса сигма-точек α, β, κ выбираются следующими: $\alpha = 1.0e^{-4}, \dots, 1.0$, β полагается равной либо нулю, либо 2 для гауссовых шумов, κ полагается равной либо нулю, либо $3-n$ [6]. Матрица сигма-точек и веса задаются как:

$$\begin{aligned}
\mathcal{X}_i &= \hat{\mathbf{x}}, & i &= 0; \\
\mathcal{X}_i &= \hat{\mathbf{x}} + \gamma\sqrt{\mathbf{P}}_i, & i &= 1, \dots, n; \\
\mathcal{X}_i &= \hat{\mathbf{x}} - \gamma\sqrt{\mathbf{P}}_{i-n}, & i &= n+1, \dots, 2n; \\
W_i^{(m)} &= \lambda / \gamma^2, & i &= 0; \\
W_i^{(c)} &= W_i^{(m)} + (1-\alpha^2 + \beta), & i &= 0; \\
W_i^{(m)} &= W_i^{(c)} = 1 / (2\gamma^2), & i &= 1, \dots, 2n,
\end{aligned} \quad (22)$$

где $\lambda = \gamma^2 - n$, $\gamma^2 = \alpha^2(n + \kappa)$.

Более детальной методики выбора параметров разброса сигма-точек, применимой в каждом конкретном случае, не выработано, поэтому фильтр рекомендуется настраивать путем опытного подбора на экспериментальных данных.

Алгоритм стандартного сигма-точечного фильтра по [6] для масштабируемого и обобщенного (веса полагаются $W_i = W_i^{(c)} = W_i^{(m)}$ при этом) ансцентного преобразования представлен далее.

Алгоритм 2: СТФК (UKF)

1) Инициализация:

Выбор параметров разброса сигма-точек и вычисление весов среднего $W_i^{(m)}$ и ковариации $W_i^{(c)}$ по (21) или (22).

Задание $\hat{\mathbf{x}}_0, \mathbf{P}_0, \mathbf{Q}, \mathbf{R}$ по (9)–(12).

2) Экстраполяция:

Вычисление матрицы сигма-точек \mathcal{X} размерности $[n, 2n+1]$ (сигма-точки – столбцы):

$$\mathcal{X}_{k-1} = [\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{x}}_{k-1} + \gamma\sqrt{\mathbf{P}_{k-1}}, \hat{\mathbf{x}}_{k-1} - \gamma\sqrt{\mathbf{P}_{k-1}}]. \quad (23)$$

$$\mathcal{X}_{k|k-1}^* = \mathcal{F}(\mathcal{X}_{k-1}, \Delta t); \quad (24)$$

$$\hat{\mathbf{x}}_{k|k-1}^* = \sum_{i=0}^{2N} W_i^{(m)} \mathcal{X}_{i,k|k-1}^*; \quad (25)$$

$$\mathbf{P}_{k|k-1} = \sum_{i=0}^{2N} W_i^{(c)} (\mathcal{X}_{i,k|k-1}^* - \hat{\mathbf{x}}_{k|k-1}^*)(\mathcal{X}_{i,k|k-1}^* - \hat{\mathbf{x}}_{k|k-1}^*)^T + \mathbf{Q}\Delta t. \quad (26)$$

Далее производится переычисление матрицы сигма-точек \mathcal{X} и $\hat{\mathbf{x}}_{k|k-1}$ для учета влияния шумов состояния [7]:

$$\mathcal{X}_{k|k-1} = [\hat{\mathbf{x}}_{k|k-1}^*, \hat{\mathbf{x}}_{k|k-1}^* + \gamma\sqrt{\mathbf{P}_{k|k-1}}, \hat{\mathbf{x}}_{k|k-1}^* - \gamma\sqrt{\mathbf{P}_{k|k-1}}]. \quad (27)$$

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2N} W_i^{(m)} \mathcal{X}_{i,k|k-1}; \quad (28)$$

$$\mathcal{Y}_{k|k-1} = \mathcal{H}(\mathcal{X}_{k|k-1}); \quad (29)$$

$$\hat{\mathbf{y}}_{k|k-1} = \sum_{i=0}^{2N} W_i^{(m)} \mathcal{Y}_{i,k|k-1}. \quad (30)$$

3) Коррекция:

$$\mathbf{S}_k = \sum_{i=0}^{2N} W_i^{(c)} (\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1})(\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1})^T + \mathbf{R}; \quad (31)$$

$$\mathbf{P}_{\mathbf{xy}_k} = \sum_{i=0}^{2N} W_i^{(c)} (\mathcal{X}_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1})(\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1})^T; \quad (32)$$

$$\mathbf{K}_k = \mathbf{P}_{\mathbf{xy}_k} \mathbf{S}_k^{-1}; \quad (33)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}); \quad (34)$$

$$\mathbf{P}_k = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T. \quad (35)$$

III. ПРОБЛЕМЫ РЕАЛИЗАЦИИ СТФК

При программной реализации СТФК возникает ряд вопросов, требующих отдельного рассмотрения.

А. Вычисление средне-взвешенных величин

При вычислении средневзвешенных значений $\hat{\mathbf{x}}_{k|k-1}$ и $\hat{\mathbf{y}}_{k|k-1}$ в (25), (28) и (30) при практической реализации фильтра в задаче траекторной обработки необходимо учитывать факт наличия в векторах состояния \mathbf{x} и/или измерений \mathbf{y} параметров, имеющих кольцевую область определения (например, курс или азимут). Для таких параметров средневзвешенные значения могут быть вычислены по способу, предложенному в [8].

В. Вычисление разности величин

При вычислении разностей в (23), (26), (27), (31), (32), (34) также необходимо учитывать кольцевую область определения, например, курса или азимута. Данная операция может быть выполнена по способу, также предложенному в [8].

С. Выбор разброса сигма-точек и устойчивость фильтра

Легко видеть, что при обобщенном ансцентном преобразовании можно получить отрицательный вес

«нулевой» сигма-точки, задавая, соответственно, параметр $w_0 < 0$. Аналогично, для масштабируемого ансцентного преобразования существуют наборы параметров α, β, κ , дающие отрицательные веса $W_i^{(m)}$ и/или $W_i^{(c)}$. Это приводит к вычислительной нестабильности фильтра – ковариационная матрица \mathbf{P} перестает быть положительно-определенной и, следовательно, не имеет матричного квадратного корня (на множестве действительных чисел), который необходимо вычислять при создании набора сигма-точек в (20)–(22).

Возможным решением данной проблемы является переход к квадратно-корневой реализации фильтра. При такой реализации вместо ковариационных матриц в фильтре используются их квадратные корни или \mathbf{UDU}^T разложение, а значит, необходимость в вычислении квадратного корня из матрицы отпадает. Данный подход применялся в фильтре Поттера (Potter's Square Root Filter) и UD-фильтре Бирмана-Тронтонна [9], которые являются модификациями РФК. В [6] был впервые предложен алгоритм квадратно-корневого сигма-точечного фильтра Калмана, который, однако, не решил проблему отрицательного веса «нулевой» сигма-точки, поэтому реализация [6] была названа в [10] псевдо-квадратно-корневой. Наконец, в работах [11,12] проблема квадратно-корневой реализации сигма-точечного фильтра была успешно решена при помощи J-ортогонального QR-разложения, учитывающего знаки весов.

Таким образом, проблема отрицательного веса «нулевой» сигма-точки может быть решена либо путем использования базового ансцентного преобразования с набором из $2n$ сигма-точек (и тогда в принципе нет «нулевой» точки и проблемы ее отрицательного веса: данный фильтр также называется кубатурным фильтром Калмана – см. далее), либо путем использования квадратно-корневой версии сигма-точечного фильтра с J-ортогональным QR-разложением. Далее будут рассмотрены оба пути.

IV. КУБАТУРНЫЙ ФИЛЬТР КАЛМАНА

Кубатурный фильтр (КФК), как самостоятельный подвид сигма-точечного фильтра, был впервые представлен в [10], однако, как можно убедиться, является частным случаем при выборе разброса сигма-точек обобщенного ансцентного преобразования $w_0 = 0$ или масштабируемого ансцентного преобразования при $\alpha = 1, \beta = 0, \kappa = 0$.

Алгоритм 3: КФК (CKF)

1) Инициализация:

Вычисление весов W_i по (20).

Задание $\hat{\mathbf{x}}_0, \mathbf{P}_0, \mathbf{Q}, \mathbf{R}$ по (9)–(12).

2) Экстраполяция:

Вычисление матрицы сигма-точек \mathcal{X} размерности $[n, 2n]$ (сигма-точки – столбцы):

$$\mathcal{X}_{k-1} = \begin{bmatrix} \hat{\mathbf{x}}_{k-1} + \gamma \sqrt{\mathbf{P}_{k-1}}, & \hat{\mathbf{x}}_{k-1} - \gamma \sqrt{\mathbf{P}_{k-1}} \end{bmatrix}. \quad (36)$$

$$\mathcal{X}_{k|k-1}^* = \mathcal{F}(\mathcal{X}_{k-1}, \Delta t); \quad (37)$$

$$\hat{\mathbf{x}}_{k|k-1}^* = \sum_{i=1}^{2N} W_i \mathcal{X}_{i,k|k-1}^*; \quad (38)$$

$$\mathbf{P}_{k|k-1} = \sum_{i=1}^{2N} W_i (\mathcal{X}_{i,k|k-1}^* - \hat{\mathbf{x}}_{k|k-1}^*)(\mathcal{X}_{i,k|k-1}^* - \hat{\mathbf{x}}_{k|k-1}^*)^T + \mathbf{Q} \Delta t; \quad (39)$$

$$\mathcal{X}_{k|k-1} = \begin{bmatrix} \hat{\mathbf{x}}_{k|k-1}^* + \gamma \sqrt{\mathbf{P}_{k|k-1}}, & \hat{\mathbf{x}}_{k|k-1}^* - \gamma \sqrt{\mathbf{P}_{k|k-1}} \end{bmatrix}; \quad (40)$$

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=1}^{2N} W_i \mathcal{X}_{i,k|k-1}; \quad (41)$$

$$\mathcal{Y}_{k|k-1} = \mathcal{H}(\mathcal{X}_{k|k-1}); \quad (42)$$

$$\hat{\mathbf{y}}_{k|k-1} = \sum_{i=1}^{2N} W_i \mathcal{Y}_{i,k|k-1}. \quad (43)$$

3) Коррекция:

$$\mathbf{S}_k = \sum_{i=1}^{2N} W_i (\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1})(\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1})^T + \mathbf{R}; \quad (44)$$

$$\mathbf{P}_{\mathbf{xy}_k} = \sum_{i=1}^{2N} W_i (\mathcal{X}_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1})(\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1})^T. \quad (45)$$

Далее по (33)–(35).

V. КВАДРАТНО-КОРНЕВЫЕ ФИЛЬТРЫ КАЛМАНА

Прежде чем перейти к рассмотрению квадратно-корневых сигма-точечных фильтров, рассмотрим вначале квадратно-корневой расширенный фильтр. Фильтр Поттера и UD-фильтр, упомянутые ранее, рассмотрены не будут, поскольку в [9] показано, что первый практически не отличается по точности от РФК, а UD-фильтр иногда может давать худший результат.

A. Квадратно-корневой расширенный фильтр Калмана

Для большей аналогии с обычным, неквадратно-корневым фильтром, квадратно-корневые матрицы далее обозначаются как $\sqrt{\mathbf{A}} = \mathbf{\bar{A}} = \text{chol}(\mathbf{A})$. Уравнения фильтра приводятся по [13].

Алгоритм 4: КК-РФК (SREKF)

1) Инициализация:

$$\hat{\mathbf{x}}(t_0) = \mathbb{E}[\mathbf{x}(t_0)] = \hat{\mathbf{x}}_0; \quad (46)$$

$$\bar{\mathbf{P}}(t_0) = \text{chol} \left(\mathbb{E} \left[(\mathbf{x}(t_0) - \hat{\mathbf{x}}_0)(\mathbf{x}(t_0) - \hat{\mathbf{x}}_0)^T \right] \right) = \bar{\mathbf{P}}_0; \quad (47)$$

$$\bar{\mathbf{Q}}_k = \text{chol} \left(\mathbb{E} \left[\mathbf{w}_k \mathbf{w}_k^T \right] \right) = \bar{\mathbf{Q}}; \quad (48)$$

$$\bar{\mathbf{R}}_k = \text{chol} \left(\mathbb{E} \left[\mathbf{v}_k \mathbf{v}_k^T \right] \right) = \bar{\mathbf{R}}. \quad (49)$$

2) Экстраполяция:

Вычисление $\hat{\mathbf{x}}_{k|k-1}$ по (13).

$$\bar{\mathbf{P}}_{k|k-1} = \left[\text{QR} \left(\begin{bmatrix} \mathbf{F}_k \bar{\mathbf{P}}_{k-1}, & \bar{\mathbf{Q}} \sqrt{\Delta t} \end{bmatrix}^T \right) \right]^T, \quad (50)$$

где обозначение QR – QR-разложение матрицы, возвращающее только матрицу \mathbf{R} .

Вычисление $\hat{\mathbf{y}}_{k|k-1}$ по (15).

3) Коррекция:

$$\bar{\mathbf{S}}_k = \left[QR \left(\left[\mathbf{H}_k \bar{\mathbf{P}}_{k|k-1}, \bar{\mathbf{R}} \right]^T \right) \right]^T; \quad (51)$$

$$\mathbf{P}_{xy_k} = \bar{\mathbf{P}}_{k|k-1} \bar{\mathbf{P}}_{k|k-1}^T \mathbf{H}_k^T; \quad (52)$$

$$\mathbf{K}_k = \mathbf{P}_{xy_k} (\bar{\mathbf{S}}_k^T)^{-1} \bar{\mathbf{S}}_k^{-1}; \quad (53)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}); \quad (54)$$

$$\mathbf{A} = \mathbf{K}_k \bar{\mathbf{S}}_k. \quad (55)$$

Цикл по j -ым столбцам \mathbf{A} :

$$\bar{\mathbf{P}}_k = \text{cholupdate}(\bar{\mathbf{P}}_k, \mathbf{A}_j, -1); \quad (56)$$

Конец цикла.

Здесь *cholupdate* в (56) – соответствующая функция MATLAB/Octave или **chdd* библиотеки LINPACK [14].

Стоит также отметить, что (53) может быть вычислена в MATLAB/Octave как $\mathbf{K}_k = (\mathbf{P}_{xy_k} / \bar{\mathbf{S}}_k^T) / \bar{\mathbf{S}}_k$ или при помощи оператора *solve* в Armadillo C++ как $\mathbf{K}_k = \left[\text{solve}(\bar{\mathbf{S}}_k^T, \text{solve}(\bar{\mathbf{S}}_k, \mathbf{P}_{xy_k}^T)) \right]^T$ во избежание использования операции обращения матрицы [6].

В. Квадратно-корневой сигма-точечный фильтр Калмана

Представленный в [6] алгоритм SRUKF был улучшен введением J-ортогонального QR разложения в [11] и представлен в пригодном для практической реализации виде в [12]: впервые был изложен прием перестановки «нулевой» сигма-точки, потенциально имеющей отрицательный вес, в (23) и (27) из крайнего левого столбца в крайний правый и, соответственно, блочной пристыковки ковариационных матриц шумов процесса и измерений не справа, а слева в (50) и (51). Аналогичной перестановке подлежат веса среднего и ковариации. Данная модификация [12] устранила препятствия для практической реализации алгоритма SRUKF с отрицательным весом «нулевой» сигма-точки.

Сигма-точки и веса обобщенного ансцентного преобразования (21) преобразуются к виду:

$$\mathcal{X}_i = \hat{\mathbf{x}} + \gamma \sqrt{\mathbf{P}}_{i+1}, \quad i = 0, \dots, n-1; \quad (57)$$

$$\mathcal{X}_i = \hat{\mathbf{x}} - \gamma \sqrt{\mathbf{P}}_{i+1-n}, \quad i = n-1, \dots, 2n-1;$$

$$\mathcal{X}_i = \hat{\mathbf{x}}, \quad i = 2n;$$

$$W_i = (1 - w_0) / 2n, \quad i = 0, \dots, 2n-1;$$

$$W_i = w_0, \quad i = 2n,$$

а масштабируемого преобразования (22) к виду:

$$\mathcal{X}_i = \hat{\mathbf{x}} + \gamma \sqrt{\mathbf{P}}_i, \quad i = 0, \dots, n-1; \quad (58)$$

$$\mathcal{X}_i = \hat{\mathbf{x}} - \gamma \sqrt{\mathbf{P}}_{i-n}, \quad i = n-1, \dots, 2n-1;$$

$$\mathcal{X}_i = \hat{\mathbf{x}}, \quad i = 2n;$$

$$W_i^{(m)} = W_i^{(c)} = 1 / (2\gamma^2), \quad i = 0, \dots, 2n-1;$$

$$W_i^{(m)} = \lambda / \gamma^2, \quad i = 2n;$$

$$W_i^{(c)} = W_i^{(m)} + (1 - \alpha^2 + \beta), \quad i = 2n.$$

Ссылка на алгоритм J-ортогонального QR-разложения приводится авторами [11]. Квадратная матрица \mathbf{J} данного разложения определяется как диагональная матрица, элементы главной диагонали которой равны +1 или -1 [15]. Учитывая указанный выше прием с перестановкой сигма-точек, получим:

$$\mathbf{J}_M = \text{diag}\{\mathbf{I}_{M-1}, -1\}, \quad (59)$$

где \mathbf{I}_{M-1} – единичная матрица размера $[M-1, M-1]$.

Отмечая красоту и изящество алгоритма SRUKF, представленного в [11,12] по форме [16], стоит заметить, что введение и использование матрицы весов (см. (30) в [11]):

$$|\mathbf{W}|^{1/2} = \left(\mathbf{I}_{2n+1} - \mathbf{1}^T \otimes \left[W_0^{(m)}, \dots, W_{2n}^{(m)} \right]^T \right) \text{diag} \left\{ \sqrt{|W_0^{(c)}|}, \dots, \sqrt{|W_{2n}^{(c)}|} \right\}$$

усложняет необходимые проверки, о которых было сказано в III.A и III.B, поэтому далее изложены алгоритмы с использованием матриц сигма-точек $\Delta\mathcal{X}$ и $\Delta\mathcal{Y}$ пространств состояния и измерений соответственно.

Алгоритм 5: КК-СТФК (SRUKF)

1) Инициализация:

Выбор параметров разброса сигма-точек и вычисление весов среднего $W_i^{(m)}$ и ковариации $W_i^{(c)}$ по (57) или (58).

Задание $\hat{\mathbf{x}}_0, \bar{\mathbf{P}}_0, \bar{\mathbf{Q}}, \bar{\mathbf{R}}$ по (46)–(49) соответственно.

2) Экстраполяция:

Вычисление матрицы сигма-точек \mathcal{X} размерности $[n, 2n+1]$ (сигма-точки – столбцы, «нулевая» сигма-точка справа):

$$\mathcal{X}_{k-1} = [\hat{\mathbf{x}}_{k-1} + \gamma \bar{\mathbf{P}}_{k-1}, \quad \hat{\mathbf{x}}_{k-1} - \gamma \bar{\mathbf{P}}_{k-1}, \quad \hat{\mathbf{x}}_{k-1}]. \quad (60)$$

Вычисляются $\mathcal{X}_{k|k-1}^*$ и $\hat{\mathbf{x}}_{k|k-1}^*$ по (24), (25) соответственно, учитывая перестановку «нулевой» сигма-точки.

$$\Delta\mathcal{X}_{k|k-1}^* = \sqrt{|W_i^{(c)}|} \left(\mathcal{X}_{i,k|k-1}^* - \hat{\mathbf{x}}_{k|k-1}^* \right), \quad i = 0, \dots, 2n; \quad (61)$$

$$\bar{\mathbf{P}}_{k|k-1} = \left[JQR \left(\left[\bar{\mathbf{Q}} \sqrt{\Delta t}, \quad \Delta\mathcal{X}_{k|k-1}^* \right]^T, \mathbf{J}_{n+(2n+1)} \right) \right]^T, \quad (62)$$

где $JQR(\mathbf{A}, \mathbf{J})$ – J-ортогональное QR-разложение матрицы \mathbf{A} , возвращающее только матрицу \mathbf{R} , \mathbf{J} – квадратная матрица знаков соответствующей размерности (см. (59)).

$$\mathcal{X}_{k|k-1} = [\hat{\mathbf{x}}_{k|k-1}^* + \gamma \bar{\mathbf{P}}_{k|k-1}, \quad \hat{\mathbf{x}}_{k|k-1}^* - \gamma \bar{\mathbf{P}}_{k|k-1}, \quad \hat{\mathbf{x}}_{k|k-1}^*]. \quad (63)$$

Далее по (28)–(30), учитывая перестановку «нулевой» сигма-точки.

3) Коррекция:

$$\Delta\mathcal{X}_{k|k-1} = \sqrt{|W_i^{(c)}|} \left(\mathcal{X}_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1} \right), \quad i = 0, \dots, 2n; \quad (64)$$

$$\Delta\mathcal{Y}_{k|k-1} = \sqrt{|W_i^{(c)}|} \left(\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1} \right), \quad i = 0, \dots, 2n; \quad (65)$$

$$\bar{\mathbf{S}}_k = \left[JQR \left(\left[\bar{\mathbf{R}}, \quad \Delta\mathcal{Y}_{k|k-1} \right]^T, \mathbf{J}_{m+(2n+1)} \right) \right]^T; \quad (66)$$

$$\mathbf{P}_{xy_k} = \Delta \mathcal{X}_{k|k-1} \mathbf{J}_{2n+1} \Delta \mathcal{Y}_{k|k-1}^T. \quad (67)$$

Далее по (59)–(60).

$$\bar{\mathbf{P}}_k = \left[JQR \left(\begin{bmatrix} \mathbf{K}_k \bar{\mathbf{R}}, & \Delta \mathcal{X}_{k|k-1} - \mathbf{K}_k \Delta \mathcal{Y}_{k|k-1} \end{bmatrix}^T, \mathbf{J}_{m+(2n+1)} \right) \right]^T. \quad (68)$$

С. Квадратно-корневой сигма-точечный фильтр Калмана с блочной коррекцией

В представленном выше алгоритме 5 на этапе коррекции требуется 2 раза вычислять QR-разложение матрицы в (66) и (68). С целью сокращения вычислений в [16,11,12] предложен вариант фильтра с блочной коррекцией, требующей только одну операцию QR-разложения. Рассмотрим такой вариант далее.

Алгоритм 6: КК-СТФКБ (SRUKFB)

1) Инициализация (по алгоритму 5).

2) Экстраполяция (по алгоритму 5).

3) Коррекция:

Вычисление $\Delta \mathcal{X}$ и $\Delta \mathcal{Y}$ по (64) и (65) соответственно.

Формирование блочной матрицы \mathbf{B} размера $[m+n, m+(2n+1)]$ следующим образом:

$$\mathbf{B} = \begin{bmatrix} \bar{\mathbf{R}}, & \Delta \mathcal{Y}_{k|k-1} \\ 0, & \Delta \mathcal{X}_{k|k-1} \end{bmatrix}. \quad (69)$$

Вычисление блочной матрицы $\tilde{\mathbf{P}}$ и считывание из нее результата:

$$\tilde{\mathbf{P}} = \left[JQR \left(\mathbf{B}^T, \mathbf{J}_{m+(2n+1)} \right) \right]^T; \quad (70)$$

$$\begin{bmatrix} \bar{\mathbf{S}}_k & 0 \\ \mathbf{P}_{xy_k}^* & \bar{\mathbf{P}}_k \end{bmatrix} = \tilde{\mathbf{P}}. \quad (71)$$

$$\mathbf{K}_k = \mathbf{P}_{xy_k}^* \bar{\mathbf{S}}_k^{-1}; \quad (72)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}). \quad (73)$$

Д. Квадратно-корневой кубатурный фильтр Калмана

Аналогично алгоритму 5, алгоритм 3 модифицируется для использования квадратно-корневых матриц, причем отсутствует необходимость использовать J-ортогональное QR-разложение матриц, обходясь обычным.

Алгоритм 7: КК-КФК (SRCKF)

1) Инициализация:

Вычисление весов W_i по (20).

Задание $\hat{\mathbf{x}}_0$, $\bar{\mathbf{P}}_0$, $\bar{\mathbf{Q}}$, $\bar{\mathbf{R}}$ по (46)–(49) соответственно.

2) Экстраполяция:

Вычисление матрицы сигма-точек \mathcal{X} размерности $[n, 2n]$ (сигма-точки – столбцы):

$$\mathcal{X}_{k-1} = [\hat{\mathbf{x}}_{k-1} + \gamma \bar{\mathbf{P}}_{k-1}, \quad \hat{\mathbf{x}}_{k-1} - \gamma \bar{\mathbf{P}}_{k-1}]. \quad (74)$$

Вычисляются $\mathcal{X}_{k|k-1}^*$ и $\hat{\mathbf{x}}_{k|k-1}^*$ по (37), (38) соответственно.

$$\Delta \mathcal{X}_{k|k-1}^* = \sqrt{W_i} (\mathcal{X}_{i,k|k-1}^* - \hat{\mathbf{x}}_{k|k-1}^*), \quad i = 1, \dots, 2n; \quad (75)$$

$$\bar{\mathbf{P}}_{k|k-1} = \left[QR \left(\begin{bmatrix} \Delta \mathcal{X}_{k|k-1}^*, & \bar{\mathbf{Q}} \sqrt{\Delta t} \end{bmatrix}^T \right) \right]^T; \quad (76)$$

$$\mathcal{X}_{k|k-1} = [\hat{\mathbf{x}}_{k|k-1}^* + \gamma \bar{\mathbf{P}}_{k|k-1}, \quad \hat{\mathbf{x}}_{k|k-1}^* - \gamma \bar{\mathbf{P}}_{k|k-1}]. \quad (77)$$

Далее по (41)–(43).

3) Коррекция:

$$\Delta \mathcal{X}_{k|k-1} = \sqrt{W_i} (\mathcal{X}_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}), \quad i = 1, \dots, 2n; \quad (78)$$

$$\Delta \mathcal{Y}_{k|k-1} = \sqrt{W_i} (\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1}), \quad i = 1, \dots, 2n; \quad (79)$$

$$\bar{\mathbf{S}}_k = \left[QR \left(\begin{bmatrix} \Delta \mathcal{Y}_{k|k-1}, & \bar{\mathbf{R}} \end{bmatrix}^T \right) \right]^T; \quad (80)$$

$$\mathbf{P}_{xy_k} = \Delta \mathcal{X}_{k|k-1} \Delta \mathcal{Y}_{k|k-1}^T. \quad (81)$$

Далее по (53)–(54).

$$\bar{\mathbf{P}}_k = \left[QR \left(\begin{bmatrix} \Delta \mathcal{X}_{k|k-1} - \mathbf{K}_k \Delta \mathcal{Y}_{k|k-1}, & \mathbf{K}_k \bar{\mathbf{R}} \end{bmatrix}^T \right) \right]^T. \quad (82)$$

Е. Квадратно-корневой кубатурный фильтр Калмана с блочной коррекцией

Аналогично алгоритму 6 вводится блочная коррекция в алгоритм 7.

Алгоритм 8: КК-КФКБ (SRCKFB)

1) Инициализация (по алгоритму 7).

2) Экстраполяция (по алгоритму 7).

3) Коррекция:

Вычисление $\Delta \mathcal{X}$ и $\Delta \mathcal{Y}$ по (64) и (65) соответственно.

Формирование блочной матрицы \mathbf{B} размера $[m+n, m+2n]$ по (69).

Вычисление блочной матрицы $\tilde{\mathbf{P}}$ и считывание из нее результата:

$$\tilde{\mathbf{P}} = \left[QR \left(\mathbf{B}^T \right) \right]^T. \quad (83)$$

Далее по (71)–(73).

VI. ГИБРИДНЫЕ ФИЛЬТРЫ КАЛМАНА

Нетрудно видеть возрастание объема вычислений в сигма-точечном фильтре относительно расширенного. Так, в одной только операции экстраполяции требуется 2 раза вычислить набор сигма-точек в (23) и (27), что сильно увеличивает время выполнения. Стоит отметить, что именно экстраполяция является одной из самых вычислительно затратных операций одного такта траекторной обработки, состоящего в общем случае из инициализации траекторий, их экстраполяции, селекции отметок по траекториям и коррекции параметров траекторий по привязанным на этапе селекции отметкам.

Исходя из этого, в целях снижения вычислительной нагрузки, предлагаются к рассмотрению гибридные фильтры, призванные сократить объем вычислений за счет использования этапа экстраполяции из РФК.

A. Расширенно-сигма-точечный фильтр Калмана

В данном фильтре используется экстраполяция от расширенного фильтра Калмана, а коррекция от сигма-точечного фильтра, с учетом необходимости предварительного создания сигма-точек.

Алгоритм 9: РСТФК (EUKF)

1) Инициализация (по алгоритму 2).

2) Экстраполяция (по алгоритму 1).

3) Коррекция:

Вычисление матрицы сигма-точек \mathcal{X} размерности $[n, 2n+1]$ (сигма-точки – столбцы):

$$\mathcal{X}_{k|k-1} = [\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{x}}_{k-1} + \gamma\sqrt{\mathbf{P}_{k|k-1}}, \hat{\mathbf{x}}_{k-1} - \gamma\sqrt{\mathbf{P}_{k|k-1}}]. \quad (84)$$

Вычисление матрицы сигма-точек \mathcal{Y} размерности $[n, 2n+1]$ (сигма-точки – столбцы):

$$\mathcal{Y}_{k|k-1} = \mathcal{H}(\mathcal{X}_{k|k-1}). \quad (85)$$

Далее по (31)–(35).

B. Расширенно-кубатурный фильтр Калмана

Аналогично РСТФК, здесь используется экстраполяция расширенного фильтра Калмана, а коррекция – кубатурного фильтра, также с учетом необходимости предварительного создания сигма-точек.

Алгоритм 10: РКФК (ECKF)

1) Инициализация (по алгоритму 3).

2) Экстраполяция (по алгоритму 1).

3) Коррекция:

Вычисление матрицы сигма-точек \mathcal{X} размерности $[n, 2n]$ (сигма-точки – столбцы):

$$\mathcal{X}_{k|k-1} = [\hat{\mathbf{x}}_{k-1} + \gamma\sqrt{\mathbf{P}_{k|k-1}}, \hat{\mathbf{x}}_{k-1} - \gamma\sqrt{\mathbf{P}_{k|k-1}}]. \quad (86)$$

Вычисление матрицы сигма-точек \mathcal{Y} размерности $[n, 2n]$ (сигма-точки – столбцы):

$$\mathcal{Y}_{k|k-1} = \mathcal{H}(\mathcal{X}_{k|k-1}). \quad (87)$$

Далее по (44), (45), (33)–(35).

VII. КВАДРАТНО-КОРНЕВЫЕ ГИБРИДНЫЕ ФИЛЬТРЫ КАЛМАНА

Вводятся квадратно-корневые гибридные фильтры и их версии с блочной коррекцией путем комбинирования соответствующих блоков алгоритмов, представленных выше:

1) квадратно-корневой расширенно-сигма-точечный фильтр Калмана (КК-РСТФК, SREUKF);

2) квадратно-корневой расширенно-сигма-точечный фильтр Калмана с блочной коррекцией (КК-РСТФКБ, SREUKFB);

3) квадратно-корневой расширенно-кубатурный фильтр Калмана (КК-РКФК, SRECKF);

4) квадратно-корневой расширенно-кубатурный фильтр Калмана с блочной коррекцией (КК-РКФКБ, SRECKFB).

Полные алгоритмы данных фильтров не приводятся для краткости изложения.

VIII. ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ

A. Модель системы

Для экспериментальной проверки представленных фильтров введем пространство измерений $\mathbf{y}\{r, \beta, Vr\}$, где r – дальность, β – азимут относительно севера, Vr – радиальная скорость и пространство состояния $\mathbf{x}\{x, y, V, K, \dot{K}\}$, где x, y – плоские декартовы координаты (ось x направлена на север, ось y – на восток, начало координат – РЛС), V – полная скорость объекта, K – курс объекта относительно севера, \dot{K} – скорость изменения курса. Неоднозначности (например, по дальности и/или скорости) в векторе измерений отсутствуют. Таким образом, задается модель движения объекта с постоянной скоростью и постоянной скоростью изменения курса, также известная как Coordinated Turn (CT). Данная модель является универсальной и позволяет отслеживать не только маневры объекта по курсу, но и прямолинейное движение путем управления соответствующим элементом ковариационной матрицы \mathbf{Q} .

Стоит также отметить, что данная модель, содержащая «естественный» параметр движения объекта – полную скорость, а не ее проекции на декартовы оси (V_x и V_y), является лучшим выбором, что показано в [17,18,19].

Зададим функционал \mathcal{F} , см. (6):

$$\left\{ \begin{array}{l} x_i = x_{i-1} + dx, \\ y_i = y_{i-1} + dy, \\ V_i = V_{i-1}, \\ K_i = K', \\ \dot{K}_i = \dot{K}_{i-1}, \end{array} \right\} \begin{array}{l} dx = -(V_{i-1} / \dot{K}_{i-1})(\cos(K') - \cos(K_{i-1})); \\ dy = (V_{i-1} / \dot{K}_{i-1})(\sin(K') - \sin(K_{i-1})); \\ K' = K_{i-1} + \dot{K}_{i-1}\Delta t; \end{array} \quad (88)$$

и функционал \mathcal{H} , см. (7):

$$\left\{ \begin{array}{l} r_i = \sqrt{x_i^2 + y_i^2}; \beta = \text{atan2}(x_i, y_i); Vr_i = V_i \cos(K_i - \beta_i) \end{array} \right\}. \quad (89)$$

Матрицы \mathbf{F} и \mathbf{H} вычисляются по (8). Зададим ковариационные матрицы шумов состояния и измерений:

$$\mathbf{Q} = \text{diag}\{(0.001\text{км})^2, (0.001\text{км})^2, (0.1\text{м/с})^2, (0.3^\circ)^2, (0.04^\circ/\text{с})^2\}$$

и $\mathbf{R} = \text{diag}\{(0.05\text{км})^2, (0.008^\circ)^2, (0.016\text{м/с})^2\}$. Вероятность отметок зададим единичной, интервал их поступления $\Delta t = 1\text{с}$.

Истинное положение объекта зададим вектором $\mathbf{x}\{100\text{км}, 200\text{км}, 100\text{м/с}, 45^\circ, 10^{-50}/\text{с}\}$, а начальное приближение вектора состояния объекта: $\hat{\mathbf{x}}_0\{100.032\text{км}, 199.953\text{км}, 165.001\text{м/с}, 26.577^\circ, 10^{-50}/\text{с}\}$.

СКО параметров отметки соответствуют матрице \mathbf{R} , заданной выше.

Начальную матрицу \mathbf{P} зададим как

$$\mathbf{P} = \text{diag}\{(0.0075\text{км})^2, (0.0075\text{км})^2, (876\text{м/с})^2, (426^\circ)^2, (0.0016^\circ/\text{с})^2\}.$$

В. Исследование влияния параметров разброса сигма-точек

Прежде чем приступить непосредственно к сравнению фильтров, выясним влияние параметров разброса сигма-точек на качество сопровождения. Для этого возьмем КК-СТФК (SRUKF) фильтр и проведем варьирование параметров разброса сигма-точек в некотором диапазоне с определенным шагом, вычисляя для каждого случая среднеквадратическую ошибку

$$RMSE_{MC,t_k} = \sqrt{\frac{1}{MC} \sum_{mc=1}^{MC} (x_{t_k}^{TRUE} - \hat{x}_{mc,t_k})^2} \quad (90)$$

по $MC=10000$ случайных реализаций, сравнивая с аналогичной характеристикой РФК (ЕКФ). Результат сравнения представлен на рис. 1, где РФК – синий цвет, КК-СТФК при масштабируемом ансцентном преобразовании – от светло-зеленого до темно-зеленого при $\alpha = 1.0, 0.7, 0.6, 0.55, 0.5, 0.45, 0.4, 0.35$, $\beta = 2$, $\kappa = -2$.

Проведенное исследование показывает, что дальнейшее уменьшение параметра разброса $\alpha < 0.3$ масштабируемого преобразования приводит к ухудшению характеристик (также отмечено в [5]), несмотря на то, что в [6] рекомендуют выбирать малые значения α .

Масштабируемое и обобщенное ансцентное преобразование дают примерно одинаковые результаты и можно показать, что соответствующим выбором параметров разброса α, β, κ можно добиться таких же результатов, как и при выборе $\alpha=0$.

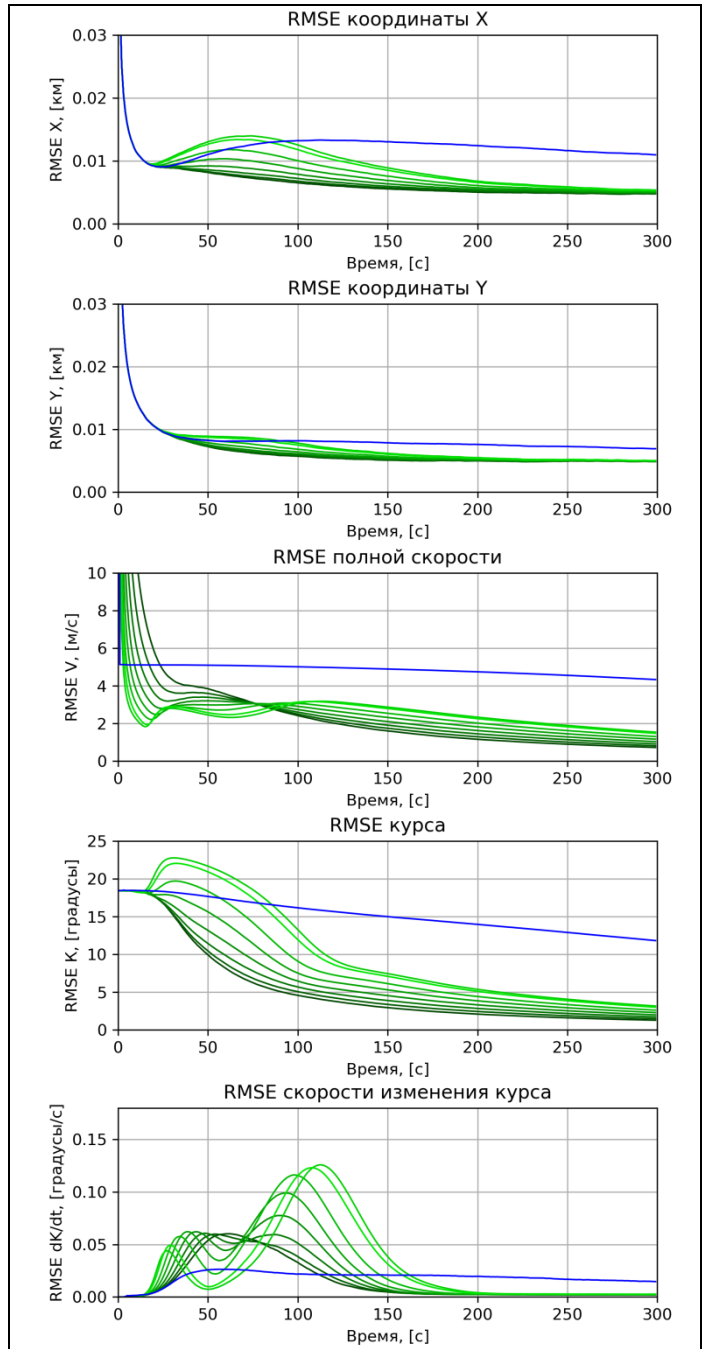
Исходя из проведенного исследования и рис.2 для дальнейших экспериментов выберем создание сигма-точек по масштабируемому ансцентному преобразованию с параметрами $\alpha = 0.687369$, $\beta = 2$, $\kappa = -2$ для сигма-точечных фильтров в полных матрицах (граничные значения, дающие положительный вес $W_0^{(c)}$, критичный для устойчивости фильтра) и $\alpha = 0.45$, $\beta = 2$, $\kappa = -2$ для квадратно-корневых сигма-точечных фильтров.

С. Сравнение RMSE фильтров

Проведем сравнение всех рассмотренных выше фильтров по 10000 случайных реализаций, аналогичным способом определяя по (90) RMSE параметров вектора состояния. Отметим, что графики для фильтров в полных и квадратно-корневых матрицах (например CKF, SRCKF, SRCKFB) совпадают, поэтому приводятся графики только для квадратно-корневых версий фильтров с блочной коррекцией. Результаты представлены на рис.3. Эталонном сравнения, как и ранее, служит ЕКФ. График для UKF ($\alpha = 0.687369$) не приводится, поскольку практически не отличается от SRCKFB ($\alpha = 1.0$).

Анализ показывает, что все сигма-точечные фильтры в конечном итоге выходят на меньший уровень RMSE всех параметров вектора состояния. Гибридные фильтры (SREUKFB и SRECKFB) в целом демонстрируют лучший результат относительно ЕКФ, при этом SRECKFB лишь незначительно уступает для скорости изменения курса на

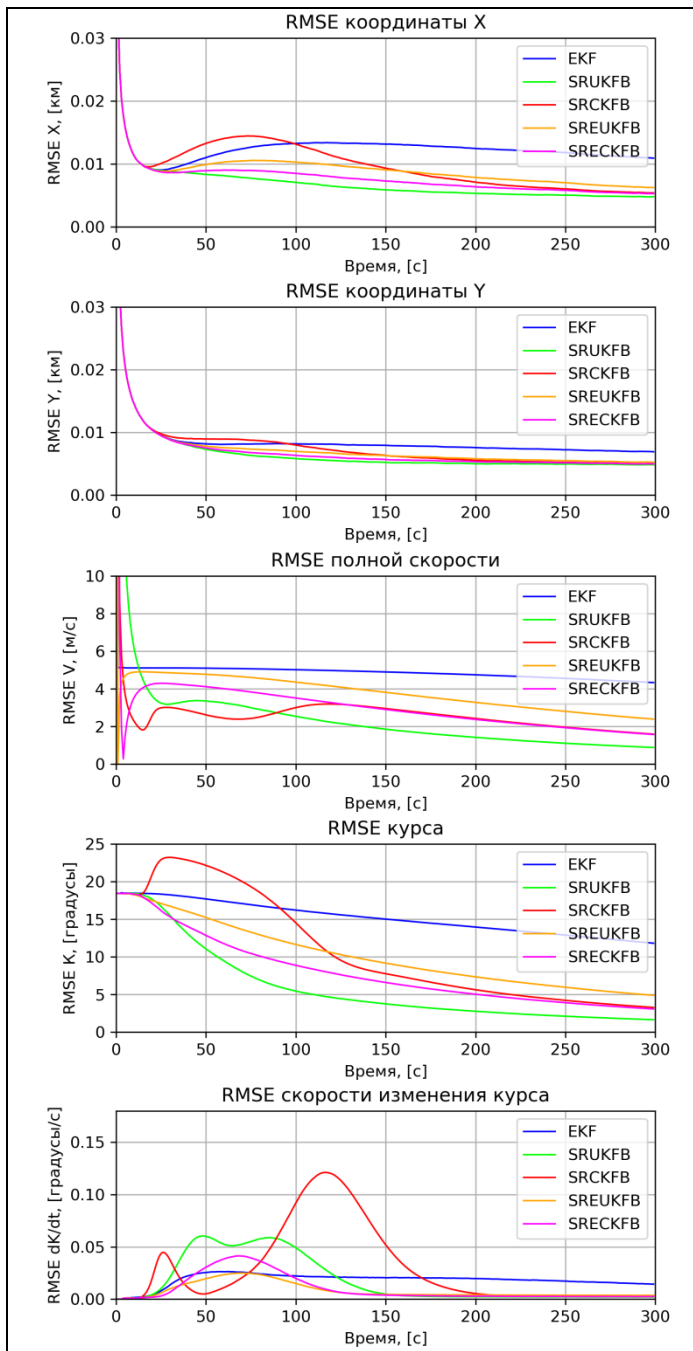
Рис.2. RMSE составляющих вектора состояния при масштабируемом ансцентном преобразовании, $\alpha = 1.0, \dots, 0.35$; $\beta = 2$; $\kappa = -2$



участке $\sim 40 \dots \sim 100$ секунд и превосходит ЕКФ по RMSE всех остальных параметров вектора состояния.

Стоит отметить, что подобная тенденция сохраняется как при уменьшении вероятности отметки с 1.0 до 0.5, так и при увеличении интервала поступления отметок Δt с 1 до 5 секунд, при этом RMSE гибридных фильтров всегда оказывается лучше или сопоставимо с ЕКФ, а сигма-точечный и кубатурный фильтры могут немного хуже оценивать скорость и/или производную курса.

Рис.3. RMSE параметров вектора состояния расширенного и различных квадратно-корневых фильтров Калмана



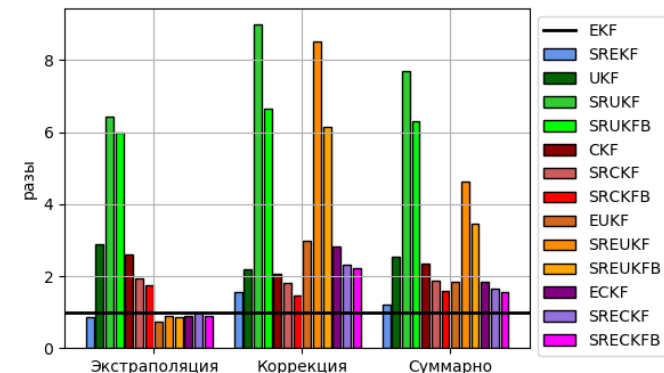
D. Сравнение скорости работы фильтров

В задаче траекторной обработки отдельное внимание уделяется скорости работы алгоритма. Вместе с тем, распространенные в последние годы многогипотезные алгоритмы траекторной обработки используют большое количество траекторных гипотез (вариантов) траектории по параметрам вектора состояния (например, курсу, скорости и т.д.), что увеличивает время экстраполяции.

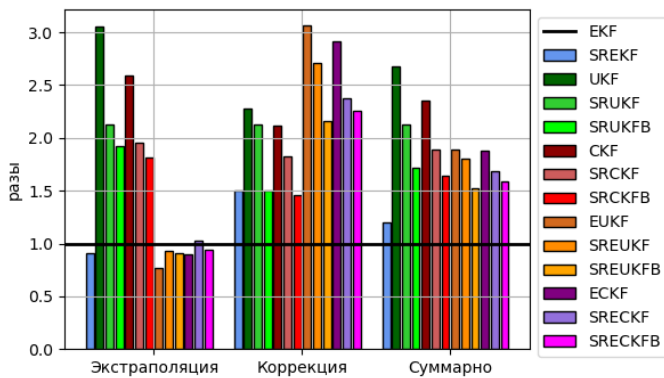
Поэтому при практическом применении сигма-точечных фильтров в алгоритмах траекторной обработки логично перейти к гибридным версиям,

Рис.4. Сравнение времени работы рассмотренных фильтров относительно РФК (EKF)

а) при использовании отрицательного веса «нулевой» сигма-точки $W_0^{(c)}$ в SRUKF(B), SREUKF(B) и, соответственно, J-ортогонального QR-разложения



б) при использовании отрицательного веса «нулевой» сигма-точки $W_0^{(c)}$ в SRUKF(B), SREUKF(B) и, соответственно, обычного QR-разложения (метод Грам-Шмидта)



сочетающим в себе вычислительно эффективный этап экстраполяции от РФК и сигма-точный этап коррекции, улучшающий RMSE оценки параметров вектора состояния, особенно на начальном этапе сопровождения. Это подтверждается как рис.3, так и рис.4, на котором представлено сравнение времени работы фильтров отдельно по этапам экстраполяции, коррекции, а также суммарно, усредненное по 10000 случайных реализаций. Значения на рис.4 приводятся относительно РФК (показанного черной горизонтальной чертой), ставшего за многие годы его использования эталоном в подобных задачах.

IX. Выводы

На основании проведенного исследования можно сделать следующие выводы:

- алгоритм J-ортогонального QR-разложения, на который ссылаются авторы [11,12], хоть и решает проблему отрицательного веса «нулевой» сигма-точки, но является вычислительно неэффективным – хорошо видно по рис.4а замедление SRUKF, SRUKFB,

Список литературы

- [1] Коновалов, А. А. Основы траекторной обработки радиолокационной информации: в 2 ч. / А. А. Коновалов. — СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2013.
- [2] Smith G.L., Schmidt S.F., McGee L.A. Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle // National Aeronautics and Space Administration. — 1962.
- [3] Julier S.J., Uhlmann J.K. Unscented filtering and nonlinear estimation // Proc. of the IEEE. — 2004. — Vol. 92, No. 3. — C.401-422.
- [4] Julier S.J. The scaled unscented transformation // Proc. of American Control Conf. — 2002. — Vol.6, No. 5. — C.4555-4559.
- [5] Sebastian B., Technische Universität Dresden. — 2016. — Режим доступа: <https://github.com/sbitzer/UKF-exposed/blob/master/UKF.pdf> (дата обращения: 06.05.2022).
- [6] Van der Merwe R., Wan E.A. The square-root unscented Kalman filter for state and parameter estimation // Proc. of the Int. Conf. on Acoustics, Speech and Signal Proc. — 2001. — Vol. 6. — C.3461-3464.
- [7] Yuanxin Wu, Dewen Hu, Meiping Wu, Xiaoping Hu. Unscented Kalman Filtering for Additive Noise Case: Augmented versus Nonaugmented // IEEE Signal Processing Letters. — 2005. — Vol. 12, No. 5. — C.357-360.
- [8] David F.C. Cubature/ Unscented/ Sigma Point Kalman Filtering with Angular Measurement Models // 18th International Conf. on Information Fusion Washington, DC. — 2015. — C.1550-1557.
- [9] Lemon K., Welch B.W. Comparison of Nonlinear Filtering Techniques for Lunar Surface Roving Navigation // Glenn Research Center, Cleveland, Ohio. NASA/TM-2008-215152. — 2008.
- [10] Arasaratnam I., Haykin S., Cubature Kalman Filters // IEEE Transactions on Automatic Control. — 2009. — Vol.54, No.6. — C.1254-1269.
- [11] Kulikov G.Yu, Kulikova M.V. Square-root accurate continuous-discrete extended-unscented Kalman filtering methods with embedded orthogonal and J-orthogonal QR decompositions for estimation of nonlinear continuous-time stochastic models in radar tracking // Signal Processing. — 2020. — Vol. 1.
- [12] Kulikov G.Yu, Kulikova M.V. The J-orthogonal Square-Root Euler-Maruyama-Based Unscented Kalman Filter for Nonlinear Stochastic Systems // 21st IFAC World Congress. — Berlin, Germany, 2020. — Vol.53, No.2. — C.2361-2366.
- [13] Nuthong C. Estimation of Tire-Road Friction Forces using Kalman Filtering for Advanced Vehicle Control // Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik. — 2009. — Режим доступа: <https://d-nb.info/1067974571/34> (дата обращения: 06.05.2022).
- [14] Репозиторий Netlib. — Режим доступа: <http://www.netlib.org/linpack/index.html> (дата обращения: 06.05.2022)
- [15] Higham, N.J. J-orthogonal matrices: Properties and Generalization // SIAM Review. — 2003. — Vol.45, No. 3. — C.504-519.
- [16] Simo Särkkä, Bayesian Filtering and Smoothing // Institute of Mathematical Statistics, Cambridge University Press. — 2013. — 232c.
- [17] Gustafsson F., Isaksson A.J. Best Choice of Coordinate System for Tracking Coordinated Turns // Proc. Of the 35th Conf. on Decision and Control. — Kobe, Japan, 1996. — C.3145-3150.
- [18] Roth M., Hendeby G., Gustafsson F. EKF/UKF Maneuvering Target Tracking using Coordinated Turn Models with Polar/Cartesian Velocity // 17th International Conf. of Information Fusion. — 2014.
- [19] Mohammad Al Shabi, Khaled Hatamleh, Samer Al Shaer, Iyad Salameh, S. Andrew Gadsden. Comprehensive Comparison of Sigma-Point Kalman Filters Applied on a Complex Maneuvering Road // Signal Processing, Sensor/Information Fusion, and Target Recognition XXV, edited by Ivan Kadar. — Proc. of SPIE. — 2016. — Vol. 9842, No. 1. — C.1-11.
- [20] Stewart G.W. Matrix Algorithms. Vol 1: Basic decompositions / G.W. Stewart. — SIAM, 1998. — 459c.

SREUKF, SREUKFB фильтров относительно UKF и EUKF соответственно, хотя ожидался обратный эффект, наблюдаемый на рис.4б, где в данных фильтрах было использовано обычное QR-разложение методом Грам-Шмидта [20] и неотрицательные веса «нулевой» сигма-точки;

- использование в SRUKF граничных значений α, β, κ , дающих неотрицательный вес $W_0^{(c)}$, критичный для устойчивости фильтра (см. VIII.B), дает близкие к SRCKF результаты;
- кубатурные фильтры работают в целом быстрее за счет использования меньшего количества сигма-точек: $2n$ штук против $2n+1$;
- применение квадратно-корневых версий фильтров полностью оправданно – хорошо видно уменьшение суммарного времени работы на рис.4 при их использовании;
- квадратно-корневые версии фильтров с блочной коррекцией работают быстрее обычных квадратно-корневых версий за счет уменьшения количества операций QR-разложения, что также следует из рис.4.

Таким образом, в качестве перспективных фильтров в задаче траекторной обработки, где особое внимание требуется уделять не только точности оценки параметров, но и скорости работы алгоритма, можно рекомендовать семейство квадратно-корневых гибридных расширенно-сигма-точечных фильтров Калмана. Данные фильтры обеспечивают разумный компромисс между точностью оценки параметров вектора состояния, где выигрыш остается за КК-СТФК, и вычислительными затратами, где традиционно лидирует используемый на протяжении более полувека хорошо изученный и распространенный в подобного рода задачах РФК.

Среди квадратно-корневых гибридных расширенно-сигма-точечных фильтров выделяются квадратно-корневые расширенно-кубатурные гибридные фильтры, в которых отсутствует проблема отрицательного веса «нулевой» сигма-точки, вследствие чего они являются вычислительно стабильными. Именно модификация такого фильтра с блочной коррекцией (КК-РКФКБ, SRECKFB), как обладающая вычислительной устойчивостью, компромиссной точностью и скоростью работы, рекомендуется к дальнейшим исследованиям и использованию в перспективных разработках алгоритмов траекторной обработки.

Альтернативой КК-РКФКБ, предполагающей возможность выбора разброса сигма-точек и, вследствие чего, более тонкую настройку алгоритма, является квадратно-корневой расширенно-сигма-точечный гибридный фильтр Калмана с блочной коррекцией (КК-РСТФКБ, SREUKFB), однако, как показало проведенное исследование, для него необходим отличный от использованного в данной статье вычислительно эффективный алгоритм J-ортогонального QR-разложения.

COMPARISON OF NONLINEAR KALMAN FILTERS APPLIED TO THE PROBLEM OF TARGET TRACKING

A.A. Sobolev
Moscow, Russia

Abstract – The paper deals with nonlinear Kalman filters – extended (EKF), unscented (UKF), cubature (CKF), their square-root versions (SREKF, SRUKF, SRCKF) and hybrid versions (EUKF, ECKF, SREUKF, SRECKF) in the context of target tracking. Some problems of practical implementation of the unscented filters are discussed, such as difference calculation between angles at the $0..2\pi$ boundary, finding weighed average of angles and most important problem of negative weight of the «zero» sigma-point. Also filters with block correction are shown, comparison of RMSE is considered, and calculation time is measured relative to the EKF. Based on the study, it was concluded that the most perspective filters are square-root hybrid extended-unscented filter with block correction (SREUKFB) and square-root hybrid extended-cubature filter with block correction (SRECKFB). The main barrier for success time-efficient implementation of the SRUKF, SREUKF and SREUKFB is the J-orthogonal QR-decomposition which realization found by author is computationally slow. So, the main recommendation after all is to use SRECKFB filter.

Keywords – Kalman filtering, nonlinear filtration, tracking.