Оглавление

1 Одновременное оценивание движения ВС и систематических ошибок. Алгоритмы параллельной фильтрации процессов, связанных через измерения

В настоящее время в системах УВД для определения параметров движения воздушных судов (координаты, скорости, ускорения и т.д.) используются алгоритмы линейного рекуррентного оценивания, близкие по используемой математической технике к фильтру Калмана. В качестве основного метода применяется алгоритм IMM. Главная особенность состоит в том, что задача оценки параметров движения для всех ВС, нахоящихся в зоне наблюдения, решается независимо для каждого ВС. Это полностью соответствует представлению о том, что движение каждого ВС никак не зависит от движения других ВС. Также это удобно с точки зрения архитектуры программы, реализующей систему мультитраекторной обработки — данные, описывающие каждое ВС, можно легко выделить в отдельный объект, который можно создавать, удалять и использовать, например, для сравнения со вновь поступающими не привязанными к конкретному ВС измернеиями. С точки зрения математических алгоритмов, такое разделение также удобно, поскольку позволяет оставаться в рамках расчётов в пространстве достаточно низкой размерности (4–6 для фильтра Калмана, 15–30 для IMM).

Наблюдение за движением ВС производится с помощью радиотехнических средств: как правило это система из нескольких радиолокаторов и система АЗН-В. Реальные измерительные средства, помимо случайных ошибок измерений, имеют систематические ошибки. Случайные ошибки измерения изначально предусмотрены архитектурой алгоритмов рекуррентного оценивания, как фильтра Калмана, так и IMM. Систематические ошибки в случае не сложных вариантов их пространственной зависимости также легко могут быть включены в алгоритмы оценивания, но при их включении обнаруживается одно весьма существенное обстоятельство: систематические ошибки одного и того же измерительного средства присутствуют в уравнении наблюдения для разных воздушных судов. Так, в простом случае связи между неизвестными оцениваемыми состояниями и измерением РЛС возникает следующее линейное уравнение наблюдения:

$$z_{a\,l}(t) = C^{\chi}(t)\chi_a(t) + C^{\varsigma}(t)\varsigma_l(t) + D(t)w_l(t)\,. \tag{1.1}$$

Здесь t — момент времени; a — индекс, обозначающий номер воздушного судна (aircraft); l — индекс радиолокатора (locator); z_{al} — вектор измерения; χ_a — вектор параметров движения ВС; ς_l — вектор параметров, характеризущий состояние РЛС; $w_l(t)$ — текущая реализация случайной ошибки РЛС; $C^{\chi}(t)$, $C^{\varsigma}(t)$, $D(t)$ — матрицы, характеризующие влияние каждого параметра на измерение.

Из вида этого уравнения ясно, что систематическая ошибка локатора l может быть оценена только совместно с параметрами движения ВС a. Но этот радиолокатор наблюдает не только это движение, также верно и обратное — ВС a наблюдается не только радиолокатором l. Фазовые переменные для разных движений оказываются «сцепленными» между собой через параметры систематических ошибок. Таким образом, система всех движений и всех систематических ошибок нуждается в совместном оценивании.

Как будет показано далее, даже в простом случае неуправляемых движений, стандартные процедуры оптимального совместного оценивания — фильтр Калмана, оценка Гаусса–Маркова — приводят к соотношениям, в которых переменые, относящиеся к разным движениям и систематическим ошибкам, существенно связаны друг с другом. Это приводит к следующим неприятным последствиям:

- нет возможности задать в программе отдельные объекты для движений разных ВС;
- затруднено создание и удаление движений;
- в вычислениях необходимо поддерживать большую матрицу ковариации ошибок оценивания, (в которую входят все кросс-ковариации для ошибок оценивания между различными ВС, между каждым ВС и каждым РЛС и т.д.) это выливается в большие затраты по времени вычисления и по памяти.

От требования, чтобы параметры оценивались оптимально, можно отказаться. При этом появляется возможность устранить нежелательные эффекты, указанные выше. Но в таком случае необходимо тщательно проектировать алгоритм оценивания, для того чтобы получаемые оценки были близки к неизвестным истинным параметрам.

Целью исследования, излагаемого ниже, является создание алгоритма лёгкого для параллельной реализации по отдельным воздушным судам и при этом обладающего низким уровнем погрешности оценивания. Исследование логически продолжает исследование, изложенное в отчёте [1].

1.1 Описание задачи наблюдения за многими ВС

Считаем, что каждое воздушное судно подчиняется независимому, но оди по структуре уравнению движения. Так движение ВС номер i имеет описание

$$d\chi_i(t) = f(t, \chi_i(t), u_i(t))dt + dv_i(t)\,,$$

где χ_i — вектор параметров движения ВС; f — функция, задающая скорости движения; $u_i(t)$ — функция управления, специфичная для ВС i; dv_i — приращение случайного возмущения для непрерывного варианта динамики; само дифференциальное уравнение сформулировано, например, в смысле Ито. В силу того, что наблюдение за ВС ведётся «в большом масштабе», вектор χ_i может содержать не очень большое число параметров, а функция f может быть выбрана достаточно простой. Измерения при помощи РЛС производятся в дискретные моменты времени, поэтому дальше удобно иметь дело с дискретизированным вариантом системы. При этом разумно ограничиться динамикой, близкой к линейной

$$\chi_i(t_k) = A_i(t_k, \chi_i(t_{k-1}), u_i(t_k))\chi_i(t_{k-1}) + B_i(t_k)v_i(t_k)\,. \tag{1.2}$$

Здесь v_i — случайное возмущение; B_i — матричная функция, формирующая влияние случайного возмущения на движение; A_i — матрица, формирующая вид движения системы, зависящая от текущего значения управления $u(t_k)$. Моменты времени t_k принадлежат некоторому дискретному множеству $\mathcal{T}$ и, на самом деле, определяются по ходу развития движения, т.е. не являются заданными заранее.

В программе мультирадарной обработки для метода IMM уравнения движения используются именно в виде (1.2). Далее, будем рассматривать более простую линейную динамику без управления

$$\chi_i(t_k) = A_i(\mathcal{T}_k)\chi_i(t_{k-1}) + B_i(t_k)v_i(t_k)\,. \tag{1.3}$$

Здесь $\mathcal{T}_k = \{t_l \in \mathcal{T}\colon\ t_l \leqslant t_k\}$ — множество моментов времени до текущего включительно.

В качестве основного варианта при моделировании выбираем прямолинейное равномерное движение на плоскости

$$\chi_i(t_k) = \begin{bmatrix} x_i(t_k) \\ v_i(t_k) \end{bmatrix}, \qquad x_i(t_k), v_i(t_k) \in \mathbb{R}^2, \qquad A_i(\mathcal{T}_k) = \begin{bmatrix} I_{2\times 2} & (t_k - t_{k-1})I_{2\times 2} \\ 0_{2\times 2} & I_{2\times 2} \end{bmatrix}, \tag{1.4}$$

где x_i, v_i обозначают векторы координат и скорости на плоскости $\mathbb{R}^2$. Непосредственно в моделировании используется $B_i \equiv 0$, $v_i \equiv 0$.

Формирование наблюдений z_{ij} будем описывать следующим уравнением наблюдения, несколько более сложным, чем уравнение (1.1):

$$z_{ij}(t) = C^{\chi}(t_k)_i\chi_i(t_k) + C^{\varsigma}_j(t_k, \chi_i(t_k))\varsigma_j(t_k) + D_j(t_k, \chi_i(t_k))w_j(t_k)\,. \tag{1.5}$$

Матрицы C^{ς}_j, D_j для всех имеющих смысл случаев зависят от положения ВС, поэтому явно указывается зависимость от χ_i. В качестве параметров ς_j могут выступать постоянная систематическая ошибка по дальности и азимуту, коэффициент линейной зависимости для систематической ошибки по дальности и т.д. Матрица C^{ς}_j описывает влияние этих неизвестных параметров на измерения.

Для параметров ς_j, характеризующих систематические ошибки РЛС, также введём динамику

$$\varsigma_j(t_k) = A^{\varsigma}_j(t_k)\varsigma_j(t_{k-1}) + B^{\varsigma}_j(t_k)v^{\varsigma}_j(t_k)\,. \tag{1.6}$$

Матрица B^{ς}_i характеризует дрейф систематических ошибок со временем. Для моделирования будем принимать:

$$\varsigma_j = \begin{bmatrix} \Delta^r_j \\ \Delta^{\alpha}_j \end{bmatrix}, \qquad A^{\varsigma}_j(t_k) \equiv I_{2\times 2}\,, \qquad B^{\varsigma}_i(t_k) \equiv 0_{2\times 2}\,. \tag{1.7}$$

Здесь $\Delta^r_j, \Delta^{\alpha}_j \in \mathbb{R}$ — значения постоянных систематических ошибок по дальности и азимуту, соответственно. Подробно понятия систематических ошибок по дальности и азимуту введены в отчёте ???.

Рассмотрим общий фазовый вектор

$$\xi(t) = \begin{bmatrix} \chi_1(t) \\ \chi_2(t) \\ \vdots \\ \chi_n(t) \\ \varsigma_1(t) \\ \varsigma_2(t) \\ \vdots \\ \varsigma_m(t) \end{bmatrix}. \tag{1.8}$$

Здесь n и m — количества наблюдаемых ВС и наблюдающих радиолокаторов. Уравнения (1.3), (1.6) можно переписать как

$$
\xi(t_k) = A(\mathcal{T}_k)\xi(t_{k-1}) + B(t_k)v(t_k) =
$$
$$
= \begin{bmatrix} A_1(\mathcal{T}_k) & & & & & 0 \\ & \ddots & & & & \\ & & A_n(\mathcal{T}_k) & & & \\ & & & A_1^{\varsigma}(t_k) & & \\ & & & & \ddots & \\ 0 & & & & & A_m^{\varsigma}(t_k) \end{bmatrix} \begin{bmatrix} \chi_1(t_{k-1}) \\ \vdots \\ \chi_n(t_{k-1}) \\ \varsigma_1(t_{k-1}) \\ \vdots \\ \varsigma_m(t_{k-1}) \end{bmatrix} +
$$
$$
+ \begin{bmatrix} B_1(t_k) & & & & & 0 \\ & \ddots & & & & \\ & & B_n(t_k) & & & \\ & & & B_1^{\varsigma}(t_k) & & \\ & & & & \ddots & \\ 0 & & & & & B_m^{\varsigma}(t_k) \end{bmatrix} \begin{bmatrix} v_1(t_k) \\ \vdots \\ v_n(t_k) \\ v_1^{\varsigma}(t_k) \\ \vdots \\ v_m^{\varsigma}(t_k) \end{bmatrix}, \quad (1.9)
$$

где матрицы A и B представляют собой блочно-диагональные матрицы, объединяющие все A_i, A_i^{ς} и B_i, B_i^{ς}.

Каждый момент времени $t_k \in \mathcal{T}$ свяжем с некоторым измерением $z_{ij}(t_k)$ положения ВС с номером i при помощи радиолокатора j. Одновременное наблюдение одного ВС несколькими радиолокаторами (как и одновременное наблюдение одним радиолокатором нескольких самолётов) будем считать пренебрежимо редким событием и не будем вводить его в модель наблюдения. Запишем уравнение наблюдения в том виде, как оно должно применяться ко всему большому фазовому вектору.

$$
z(t_k) = z_{ij}(t_k) = C(t_k, \xi(t_k))\xi(t_k) + D(t_k, \xi(t_k))w(t_k)\,, \quad (1.10)
$$
$$
C(t_k, \xi(t_k)) = \begin{array}{c} \begin{array}{ccccccccccc} & & & i & & & & n+j & & & \end{array} \\ \begin{pmatrix} 0 & \cdots & 0 & C_i^{\chi}(t_k) & 0 & \cdots & 0 & C_j^{\varsigma}(t_k, \chi_i(t_k)) & 0 & \cdots & 0 \end{pmatrix} \end{array},
$$
$$
D(t_k, \xi(t_k)) = D_j(t_k, \chi_i(t_k))\,, \qquad w(t_k) = w_j(t_k)\,.
$$

Как указывалось выше, для моделирования будем применять предположение постоянных систематических ошибок по дальности и азимуту. При этом будем использовать линеаризованную модель воздействия таких ошибок на измерения. Соответствующие матрицы $C^{\chi}(t_k)$, $C^{\varsigma}(t_k, \chi_i(t_k))$, $D(t_k, \chi_i(t_k))$ имеют вид:

$$
C_i^{\chi}(t_k) \equiv \begin{bmatrix} I_{2\times 2} & 0_{2\times 2} \end{bmatrix}, \qquad C_j^{\varsigma}(t_k, \chi_i(t_k)) = \begin{bmatrix} \frac{1}{\left\| x_i(t_k) - x_j^{\mathsf{R}} \right\|}(x_i(t_k) - x_j^{\mathsf{R}}) & \Omega_{2\times 2}^{\pi/2}(x_i(t_k) - x_j^{\mathsf{R}}) \end{bmatrix},
$$
$$
D(t_k, \chi_i(t_k)) = C_j^{\varsigma}(t_k, \chi_i(t_k))\,, \qquad \Omega_{2\times 2}^{\pi/2} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \quad (1.11)
$$

Здесь x_j^{R} — координаты точки стояния радиолокатора j; $\Omega_{2\times 2}^{\pi/2}$ — матрица поворота на угол $\frac{\pi}{2}$ против часовой стрелки на плоскости $\mathbb{R}^2$ с учётом северо-восточной системы координат. Случайные ошибки

$$
w_j^{\varsigma}(t_k) = \begin{bmatrix} w_j^r(t_k) \\ w_j^{\alpha}(t_k) \end{bmatrix}
$$

разделяются на случайные ошибки, действующие по дальности и азимуту.

Для всех случайных ошибок считаем справедливыми свойства:

$$\mathbb{E}\{v_i(t_k)\} = \mathbb{E}\{v_i^{\varsigma}(t_k)\} = \mathbb{E}\{w_j(t_k)\} = 0\,, \tag{1.12}$$
$$\mathbb{C}\mathbf{ov}\{v_{i_1}(t_k), v_{i_2}(t_l)\} = \delta_{kl}\delta_{i_1 i_2} V_{i_1}\,, \qquad \mathbb{C}\mathbf{ov}\{w_{j_1}(t_k), w_{j_2}(t_l)\} = \delta_{kl}\delta_{j_1 j_2} W_{j_1}\,,$$
$$\mathbb{C}\mathbf{ov}\{v_i(t_k), w_j(t_l)\} = 0\,, \qquad \forall i, i_1, i_2 \in 1, \ldots, n\,,\ \forall j, j_1, j_2 \in 1, \ldots, m\,,\ \forall t_k, t_l \in \mathcal{T}\,,$$

где δ_{pq} — символ Кронекера; V_i — постоянная матрица дисперсии случайных возмущений для уравнений движения; W_j — постоянная матрица дисперсии случайных ошибок наблюдения для радиолокатора j. Матрицы ковариаций для больших столбцов v и w будем обозначать

$$V = \begin{bmatrix} V_1 & & 0 \\ & \ddots & \\ 0 & & V_n \end{bmatrix}, \qquad W = \begin{bmatrix} W_1 & & 0 \\ & \ddots & \\ 0 & & W_m \end{bmatrix}.$$

Для моделирования будем применять W_j вида:

$$W_j = \begin{bmatrix} \sigma_{rj}^2 & 0 \\ 0 & \sigma_{\alpha j}^2 \end{bmatrix}, \tag{1.13}$$

где σ_{rj}, $\sigma_{\alpha j}$ — заданные среднеквадратичные отклонения для случайных ошибок наблюдения по дальности и азимуту, относящихся к радиолокатору j. Матрицы V_i будем брать одинаковыми диагональными

$$V_i = \begin{bmatrix} \sigma_{x^1}^2 & & & 0 \\ & \sigma_{x^2}^2 & & \\ & & \sigma_{v^1}^2 & \\ 0 & & & \sigma_{v^2}^2 \end{bmatrix}. \tag{1.14}$$

1.2 Задача фильтрации

Целью фильтрации является получение оценки $\hat{\xi}(t_k)$ фазового вектора ξ на момент t_k поступления последнего измерения. Предполагается, что оценка вычисляется как некоторая функция Ξ от информации обо всех измерениях до этого момента времени:

$$\hat{\xi}(t_k) = \Xi\big(\{z(t)\}_{t \in \mathcal{T}_k}\big)\,,$$

а также от априорной информации. Также предполагается, что задан некоторый критерий, по которому будет определяться качество оценивания. Популярным выбором является:

$$J(t_k) = \mathbb{E}\Big\{\|h^{\mathsf{T}}(\hat{\xi}(t_k) - \xi(t_k))\|^2\Big\}\,, \tag{1.15}$$

где h^{T} — некоторая заданная линейная функция, выделяющая, например, некоторую часть координат из всего вектора, $\xi(t_k)$ — истинное значение фазового вектора ξ в момент времени t_k. Поскольку речь идёт об оценивании в присутствии случайных ошибок наблюдения, оценка $\hat{\xi}(t_k)$ является случайной величиной, и в критерии присутствует символ математического ожидания $\mathbb{E}\{\cdot\}$.

Наиболее простыми и разумными с точки зрения оптимальности являются линейные рекуррентные оценки с линейным прогнозированием:

$$\bar{\xi}(t_k) = A(\mathcal{T}_k)\hat{\xi}(t_{k-1})\,, \tag{1.16}$$
$$\hat{\xi}(t_k) = L(t_k, R(t_k))\bar{\xi}(t_k) + K(t_k, R(t_k))z(t_k)\,, \tag{1.17}$$
$$R(t_k) = \mathcal{F}(t_k, R(t_{k-1})\,. \tag{1.18}$$

Здесь $L(t_k, R(t_k))$ и $K(t_k, R(t_k))$ — матричные коэффициенты, выбираемые для каждого момента самостоятельно, и зависящие от параметров линейных уравнений (1.9), (1.10), а также от вектора дополнительных параметров $R(t_k)$, пересчитываемого отдельно по некоторму, уже в общем случае нелинейному, правилу (1.18). Уравнение прогноза (1.16) обеспечивает оптимальную по имеющейся информации $\hat{\xi}(t_{k-1})$ оценку вектора $\xi(t_k)$ среди всех возможных оценок вообще. Т.е. при оптимальном выборе $\hat{\xi}(t_{k-1})$ оценка $\bar{\xi}(t_k)$ является оптимальной среди всех оценок вектора $\xi(t_k)$ по измерениям, предшествующим моменту t_k. Уравнение (1.17) называют уравнением коррекции. Его целью является получение новой оценки, учитывющий последнее измерение.

Далее в тексте, если рассматриваемые величины $\hat{\xi}(t_k)$, $\bar{\xi}(t_k)$, $L(t_k, R(t_k))$, и т. д. относятся к одному и тому же моменту времени t_k, скобки с аргументами в некоторых случаях будут опускаться, если это не будет создавать двусмысленности.

Популярным дополнительным условием является условие несмещённости оценки

$$\mathbb{E}\Big\{\hat{\xi}(t_k)\Big\} = \mathbb{E}\{\xi(t_k)\} \ , \tag{1.19}$$

которое в случае детерминированного фазового вектора ξ, например, в случае равенства нулю матрицы $B(t_k)$ в уравнении (1.9), принимает вид

$$\mathbb{E}\Big\{\hat{\xi}(t_k)\Big\} = \xi(t_k)\,. \tag{1.20}$$

Если оценка $\hat{\xi}(t_{k-1})$ удовлетворяет условию (1.19), легко видеть, что и оценка $\bar{\xi}(t_k)$ ему удовлетворяет в силу уравнения (1.9). Для уравнения коррекции (1.17) условие несмещённости (1.19) приводит к следующему условию

$$\begin{aligned}\mathbb{E}\Big\{\hat{\xi}(t_k)\Big\} &= L\,\mathbb{E}\{\bar{\xi}(t_k)\} + K\,\mathbb{E}\{z(t_k)\} = \\ &= L\,\mathbb{E}\{\xi(t_k)\} + K\,\mathbb{E}\{C(t_k,\xi(t_k))\xi(t_k)\} + K\,\mathbb{E}\{D(t_k,\xi(t_k))w(t_k)\} = \\ &= L\,\mathbb{E}\{\xi(t_k)\} + K\,\mathbb{E}\{C(t_k,\xi(t_k))\xi(t_k)\} + K\,\mathbb{E}\{D(t_k,\xi(t_k))\}\,\mathbb{E}\{w(t_k)\} = \\ &= L\,\mathbb{E}\{\xi(t_k)\} + K\,\mathbb{E}\{C(t_k,\xi(t_k))\xi(t_k)\} \ , \\ &\Longrightarrow L\,\mathbb{E}\{\xi(t_k)\} = I - K\,\mathbb{E}\{C(t_k,\xi(t_k))\xi(t_k)\} \ ,\end{aligned}$$

которое для случая матрицы C, не зависящей от ξ, или для случая, когда вектор ξ является детерминированным, переходит в матричное условие

$$L(t_k, R(t_k)) = I - K(t_k, R(t_k))\,C(t_k, \xi(t_k))\,. \tag{1.21}$$

В случае рассматриваемой нами модельной системы матрица C очень слабо зависит от ξ. Так, заменив в выражении (1.11) для матрицы $C_j^{\varsigma}(t_k, \xi(t_k))$ вектор $x_i(t_k)$ на $z_{ij}(t_k)$ или $\bar{x}_i(t_k)$ (часть прогнозной оценки $\bar{\xi}(t_k)$), мы получим близкое выражение, пригодное для использования в линейных алгоритмах. Далее, все алгоритмы будут рассматриваться с условием (1.21) с приближенной заменой $x_i(t_k)$ на $\bar{x}_i(t_k)$ в матрице C — это соответствует варианту Enhanced Kalman Filter (EKF) для нелинейной системы. При его подстановке в уравнение коррекции получается

$$\hat{\xi}(t_k) = \bar{\xi}(t_k) + K(t_k, R(t_k))\,\left(z(t_k) - C(t_k, \bar{\xi}(t_k))\,\bar{\xi}(t_k)\right)$$

или в упрощённой записи

$$\hat{\xi}(t_k) = \bar{\xi}(t_k) + K\left(z(t_k) - C\,\bar{\xi}(t_k)\right) \ . \tag{1.22}$$

Слагаемое $C(t_k,\bar{\xi}(t_k))\,\bar{\xi}(t_k)$ можно проинтерпретировать как прогнозное измерение на момент t_k. Таким образом в выражении оценки (1.22) фигурирует разность между действительным и прогнозым измерениями.

Далее, в разделах посвящённых алгоритмам параллельной фильтрации, поскольку все рассматриваемые соотношения касаются шага между моментами t_{k-1} и t_k, аргументы будут опускаться. Т. е. будут приняты обозначения

$$A = A(\mathcal{T}_k)\,, \qquad B = B(t_k)\,, \qquad C = C(t_k,\bar{\xi}(t_k))\,, \qquad D = D(t_k,\bar{\xi}(t_k))\,.$$

Введём обозначение для матрицы ковариаций ошибки оценивания для прогнозной оценки

$$\begin{aligned}
&\bar{P}(t_k) = \mathbb{C}\mathbf{ov}\big\{\bar{\xi}(t_k) - \xi(t_k)\big\} = \\
&= \begin{bmatrix} \mathbb{C}\mathbf{ov}\{\bar{\chi}_1(t_k)-\chi_1(t_k),\bar{\chi}_1(t_k)-\chi_1(t_k)\} & \cdots & \mathbb{C}\mathbf{ov}\{\bar{\chi}_1(t_k)-\chi_1(t_k),\bar{\varsigma}_m(t_k)-\varsigma_m(t_k)\} \\ \vdots & \ddots & \vdots \\ \mathbb{C}\mathbf{ov}\{\bar{\varsigma}_m(t_k)-\varsigma_m(t_k),\bar{\chi}_1(t_k)-\chi_1(t_k)\} & \cdots & \mathbb{C}\mathbf{ov}\{\bar{\varsigma}_m(t_k)-\varsigma_m(t_k),\bar{\varsigma}_m(t_k)-\varsigma_m(t_k)\} \end{bmatrix} = \\
&\qquad = \begin{bmatrix} \bar{P}_{\chi_1\chi_1}(t_k) & \cdots & \bar{P}_{\chi_1\varsigma_m}(t_k) \\ \vdots & \ddots & \vdots \\ \bar{P}_{\varsigma_m\chi_1}(t_k) & \cdots & \bar{P}_{\varsigma_m\varsigma_m}(t_k) \end{bmatrix}
\end{aligned} \tag{1.23}$$

и для матрицы ковариации ошибки основной оценки по измерениям до момента t_k включительно

$$\begin{aligned}
&\hat{P}(t_k) = \mathbb{C}\mathbf{ov}\Big\{\hat{\xi}(t_k) - \xi(t_k)\Big\} = \\
&= \begin{bmatrix} \mathbb{C}\mathbf{ov}\{\hat{\chi}_1(t_k)-\chi_1(t_k),\hat{\chi}_1(t_k)-\chi_1(t_k)\} & \cdots & \mathbb{C}\mathbf{ov}\{\hat{\chi}_1(t_k)-\chi_1(t_k),\hat{\varsigma}_m(t_k)-\varsigma_m(t_k)\} \\ \vdots & \ddots & \vdots \\ \mathbb{C}\mathbf{ov}\{\hat{\varsigma}_m(t_k)-\varsigma_m(t_k),\hat{\chi}_1(t_k)-\chi_1(t_k)\} & \cdots & \mathbb{C}\mathbf{ov}\{\hat{\varsigma}_m(t_k)-\varsigma_m(t_k),\hat{\varsigma}_m(t_k)-\varsigma_m(t_k)\} \end{bmatrix} = \\
&\qquad = \begin{bmatrix} \hat{P}_{\chi_1\chi_1}(t_k) & \cdots & \hat{P}_{\chi_1\varsigma_m}(t_k) \\ \vdots & \ddots & \vdots \\ \hat{P}_{\varsigma_m\chi_1}(t_k) & \cdots & \hat{P}_{\varsigma_m\varsigma_m}(t_k) \end{bmatrix}\,.
\end{aligned} \tag{1.24}$$

Приведём общие уравнения для эволюции этих матриц. В силу уравнения (1.9) и (1.16) справедливо

$$\bar{\xi}(t_k) - \xi(t_k) = A\hat{\xi}(t_{k-1}) - A\xi(t_{k-1}) - Bv(t_k) = A\left(\hat{\xi}(t_{k-1}) - \xi(t_{k-1})\right) - Bv(t_k)\,.$$

Следовательно, в силу независимости случайной ошибки динамики $v(t_k)$ и ошибок оценивания $\hat{\xi}(t_{k-1}) - \xi(t_{k-1})$, зависящих от случайных величин $v(t)$, $w(t)$ при $t \in \mathcal{T}_{k-1}$, верно соотношение

$$\bar{P}(t_k) = \mathbb{E}\Big\{\big(\bar{\xi}(t_k) - \xi(t_k)\big)\,\big(\bar{\xi}(t_k) - \xi(t_k)\big)^{\mathsf{T}}\Big\} = A\hat{P}(t_{k-1})A^{\mathsf{T}} + BVB^{\mathsf{T}}\,. \tag{1.25}$$

Пусть выполнено условие несмещённости и уравнение коррекции (1.17) переходит в (1.22), тогда для произвольного матричного коэффициента K (без разницы каким образом полученного) справедливо

$$\begin{aligned}
&\hat{\xi}(t_k) - \xi(t_k) = \bar{\xi}(t_k) - \xi(t_k) + K\left(z(t_k) - C\hat{\xi}(t_k)\right) = \\
&= \bar{\xi}(t_k) - \xi(t_k) + K\left(C\xi(t_k) + w(t_k) - C\hat{\xi}(t_k)\right) = (I - KC)\left(\bar{\xi}(t_k) - \xi(t_k)\right) + Kw(t_k)\,.
\end{aligned}$$

Так же как и при выводе соотношения для $\bar{P}(t_k)$, можно утверждать о независимости случайной ошибки наблюдения $w(t_k)$ и ошибок оценивания $\bar{\xi}(t_k) - \xi(t_k)$, так как последние зависят от случайных величин $v(t)$, $w(t)$ при $t \in \mathcal{T}_{k-1}$ и от $v(t_k)$. Следовательно, верно соотношение

$$\hat{P}(t_k) = \mathbb{E}\Big\{\Big(\hat{\xi}(t_k) - \xi(t_k)\Big)\Big(\hat{\xi}(t_k) - \xi(t_k)\Big)^{\mathsf{T}}\Big\} = \\ = (I - KC)\bar{P}(t_k)(I - KC)^{\mathsf{T}} + KWK^{\mathsf{T}} \quad (1.26)$$

известное как *формула Иозефа.*

Для критерия (1.15) известна формула

$$J(t_k) = \mathrm{tr}\Big\{h^{\mathsf{T}}\hat{P}(t_k)h\Big\}\ . \quad (1.27)$$

1.3 Уравнения оптимальной фильтрации

Уравнения оптимальной фильтрации можно получить, минимизируя след матрицы $\hat{P}(t_k)$ в соотношении (1.26) варьированием различных матричных коэффициентов K. При этом выводится коэффициент $K^*(t_k)$ минимизирующий критерий на каждом шаге работы алгоритма.

$$K^*(t_k) = \bar{P}(t_k)C^{\mathsf{T}}\left(C\bar{P}(t_k)C^{\mathsf{T}} + DWD^{\mathsf{T}}\right)^{-1}\ . \quad (1.28)$$

Интересно, что оптимальное значение K^* подходит и для любого h в формуле (1.27), т. е. соответствует равномерной по h оценке.

Полностью, с подстановкой соотношения (1.28), уравнения рекуррентной фильтрации называются уравнениями фильтра Калмана (или реккуррентной оценки Гаусса-Маркова для случая $B = 0$). Приведём их полностью:

$$\begin{aligned} \bar{\xi}(t_k) &= A\hat{\xi}(t_{k-1})\,, \\ \bar{P}(t_k) &= A\hat{P}(t_{k-1})A^{\mathsf{T}} + BVB^{\mathsf{T}}\,, \\ \Lambda &= C\bar{P}(t_k)C^{\mathsf{T}} + DWD^{\mathsf{T}}\,, \\ K^* &= \bar{P}(t_k)C^{\mathsf{T}}\Lambda^{-1}\,, \\ \hat{\xi}(t_k) &= \bar{\xi}(t_k) + K^*\left(z(t_k) - C\,\bar{\xi}(t_k)\right)\,, \\ \hat{P}(t_k) &= (I - K^*C)\bar{P}(t_k)(I - K^*C)^{\mathsf{T}} + K^*WK^{*\mathsf{T}} = \\ &= (I - K^*C)\bar{P}(t_k) = \bar{P}(t_k) - K^*\Lambda K^{*\mathsf{T}}\,. \end{aligned} \quad (1.29)$$

Матрица Λ является матрицей ковариации отклонения прогнозного измерения $C\bar{\xi}(t_k)$ от действительного измерения $z(t_k)$.

Отметим, что в качестве дополнительных параметров $R(t_k)$, по которым пересчитывается коэффициент K, в данном случае выступают прогнозная $\bar{P}(t_k)$ и действительная $\hat{P}(t_k)$ матрицы ковариаций ошибок оценивания.

Фильтр для фазового вектора.

Этап предсказания:

$$\bar{x}_t = A_x\hat{x}_{t-1}$$

$$\bar{P}_{x,t} = A_x\hat{P}_{x,t-1}A_x^T + B_xB_x^T$$

Этап коррекции:

$$\hat{x}_t = \bar{x}_t + K_x \Lambda (z_t - C_x \bar{x}_t - C_s \bar{s}_t)$$
$$\hat{P}_{x,t} = \bar{P}_{x,t} - K_x \Lambda K_x^T$$
$$K_x = \bar{P}_{x,t} C_x^T + \bar{P}_{xs,t} C_s^T$$

Фильтр для систематической ошибки.

Этап предсказания:

$$\bar{s}_t = A_s \hat{s}_{t-1}$$
$$\bar{P}_{s,t} = A_s \hat{P}_{s,t-1} A_s^T + B_s B_s^T$$

Этап коррекции:

$$\hat{s}_t = \bar{s}_t + K_s \Lambda (z_t - C_x \bar{x}_t - C_s \bar{s}_t)$$
$$\hat{P}_{s,t} = \bar{P}_{s,t} - K_s \Lambda K_s^T$$
$$K_s = \bar{P}_{s,t} C_s^T + \bar{P}_{xs,t}^T C_x^T$$

Обновление блока кросс-ковариации:

$$\bar{P}_{xs,t} = A_x \hat{P}_{xs,t-1} A_s^T$$

$$\hat{P}_{xs,t} = \bar{P}_{xs,t} - K_x \Lambda K_s^T$$

В данном случае для обоих фильтров используется одна матрица Λ:

$$\Lambda = C_x \bar{P}_{x,t} C_x^T + C_s \bar{P}_{s,t} C_s^T + C_x \bar{P}_{xs,t} C_s^T + C_s \bar{P}_{xs,t}^T C_x^T + DD^T$$

Уравнение наблюдения:

$$z_t = C_x x_t + C_s s_t + Dw$$

1.4 Упрощеные алгоритмы оценивания по Henk Blom

В статье [2] рассматривается точно такая же задача одновременного оценивания движения многих ВС и определения систематических ошибок. Приводятся варианты упрощения алгоритма фильтрации Калмана, показавшие хорошую работу на практике.

1.4.1 Фильтр Калмана для фазового вектора, Макро фильтр для систематической ошибки

Фильтр для фазового вектора.

Этап предсказания:

$$\bar{x}_t = A_x \hat{x}_{t-1}$$
$$\bar{P}_{x,t} = A_x \hat{P}_{x,t-1} A_x^T + B_x B_x^T$$

Этап коррекции:

$$\hat{x}_t = \bar{x}_t + K_x \Lambda_x (z_t - C_x \bar{x}_t - C_s \bar{s}_t)$$
$$\hat{P}_{x,t} = \bar{P}_{x,t} - K_x \Lambda_x K_x^T$$

Аппроксимация:

$$K_x = \bar{P}_{x,t} C_x^T$$

$$\Lambda_x = C_x \bar{P}_{x,t} C_x^T + DD^T$$

Фильтр для систематической ошибки.

Этап предсказания:

$$\bar{s}_t = A_s \hat{s}_{t-1}$$

$$\bar{P}_{s,t} = A_s \hat{P}_{s,t-1} A_s^T + B_s B_s^T$$

Этап коррекции:

$$\hat{s}_t = \bar{s}_t + K_s \Lambda_s (z_t - C_x \bar{x}_t - C_s \bar{s}_t)$$

$$\hat{P}_{s,t} = \bar{P}_{s,t} - K_s \Lambda_s K_s^T$$

В вычислении матриц K_s и Λ_s используются аппроксимация члена $C_x \bar{P}_{xs,t}$:

$$K_s = \bar{P}_{s,t} C_s^T + H^T$$

$$\Lambda = C_x \bar{P}_{x,t} C_x^T + C_s \bar{P}_{s,t} C_s^T + H C_s^T + C_s H^T + DD^T$$

Где H:

$$F_x = \sum_{i=1}^{M} (D_i D_i^T)^{-1}$$

$$F_s = \sum_{i=1}^{M} (D_i D_i^T)^{-1} C_{s,i}$$

$$H = -(F_x^T F_x)^{-1} F_x^T F_s \bar{P}_{s,t}$$

Где M - количество радиолокаторов.

Уравнение наблюдения:

$$z_t = C_x x_t + C_s s_t + Dw$$

1.4.2 Разделённые фильтры для фазового вектора и для систематической ошибки

Фильтр для фазового вектора.

Этап предсказания:

$$\bar{x}_t = A_x \hat{x}_{t-1}$$

$$\bar{P}_{x,t} = A_x \hat{P}_{x,t-1} A_x^T + B_x B_x^T$$

Этап коррекции:

$$\hat{x}_t = \bar{x}_t + K_x \Lambda_x (z_t - C_x \bar{x}_t - C_s \bar{s}_t)$$

$$\hat{P}_{x,t} = \bar{P}_{x,t} - K_x \Lambda_x K_x^T$$

Аппроксимация:

$$K_x = \bar{P}_{x,t} C_x^T$$

$$\Lambda_x = C_x \bar{P}_{x,t} C_x^T + DD^T$$

Фильтр для систематической ошибки.

Этап предсказания:

$$\bar{s}_t = A_s \hat{s}_{t-1}$$

$$\bar{P}_{s,t} = A_s \hat{P}_{s,t-1} A_s^T + B_s B_s^T$$

Этап коррекции:

$$\hat{s}_t = \bar{s}_t + K_s \Lambda_s (z_t - C_x \bar{x}_t - C_s \bar{s}_t)$$

$$\hat{P}_{s,t} = \bar{P}_{s,t} - K_s \Lambda_s K_s^T$$

Аппроксимация:

$$K_s = \bar{P}_{s,t} C_s^T$$

$$\Lambda_s = C_s \bar{P}_{s,t} C_s^T + D D^T$$

Уравнение наблюдения:

$$z_t = C_x x_t + C_s s_t + Dw$$

2 Алгоритм многогипотезного восстановления траектории

Данный раздел отчёта посвящен исследованию задачи получения оценок параметров движения воздушного судна по поступающим радиолокационным замерам. В рассматриваемом подходе наряду с текущими показателями (Федотов: слово параметры нехорошо исп, изза предыдущ. предлож.) движения формируются наиболее вероятные варианты предыстории движения в виде модельных траекторий с заданными начальной точкой и управлениями. Рассматривается движение в горизонтальной плоскости. Модель движения – система дифференциальных уравнений 4-го порядка с ограничением на продольное и боковое ускорения.

Формируемая совокупность возможных траекторий движения воздушного судна с кусочно-постоянными управлениями используются для получения текущей средневзвешенной оценки текущего положения ВС. В общем случае такая задача является многоэкстремальной, даже в случае, когда для её решения используются замеры одного и того же воздушного судна. Поэтому является естественным использование в качестве оценки движения ВС нескольких вариантов траектории.

Предлагаемый подход во многом пересекается с подходом, используемым в настоящее время при мультирадарной обработке данных в НИТА, где задействуется фильтр Калмана и метод IMM. В частности, как и в IMM, в каждый момент времени рассматриваются различные варианты движения. Расчёт на улучшение показателей (по сравнению с методом IMM) по оценке текущих параметров движения сделан в первую очередь на дополнительно использование оценок предыстории движения.

2.1 Постановка задачи

Рассматривается движение воздушного судна в горизонтальной плоскости. Требуется для задачи мультирадарной обработки оценить возможности использования подхода, опирающегося на способ представления траекторного движения воздушного судна в виде набора наиболее вероятных движений. В качестве текущей оценки положения ВС рассмотреть варианты выбора по "наилучшей траектории"и выбора по средней траектории с использованием принципа максимальной достоверности.

Формирование "пучка"траекторий должна производиться с условием максимальной представительности вариантов движения. Расчёты с ветвлением "пучка"и последующее прореживание с целью ограничения общего количества вероятных треков веполнять для каждого вновь поступившего замера.

Модель траекторного движения ВС должна соответствовать [ссылка на ГОСТ (есть) и сборник Красова (пока не нашел)]

Провести моделирование на реальных и модельных данных. Для обработки реальных данных выполнить преобразование в местную горизонталь для некоторой средней точки.

2.2 Схема первоначального построения и пересчёта пучка траекторий

Тут про расчёты в текущем окне

это наверно завтра

2.3 Математическая модель траекторного движения ВС

Алгоритм использует следующее модельное описание динамики самолёта:

$$\begin{aligned} \dot{x} &= v\cos\varphi, \\ \dot{z} &= v\sin\varphi, \\ \dot{\varphi} &= u/v, \\ \dot{v} &= w. \end{aligned} \tag{2.1}$$

Здесь x, z — координаты положения на плоскости; путевой угол φ — угол на плоскости между вектором скорости от оси x; v — величина скорости ($v > 0$); u — боковое ускорение; w — продольное ускорение.

Предполагаем, что управления u, w стеснены геометрическими ограничениями $|u| \le u_{\max}$, $|w| \le w_{\max}$.

При практическом использовании интегрирования с построением трека в геометрических координатах используется сетка по времени, включающая все секундные отметки (это позволяет корректно сравнивать треки с несовпадающими моментами переключения).

2.3.1 Аналитические расчёты при интегрировании уравнений движения

... непонял, какой вариант берём? (Федотов)

2.4 Структура данных программы, термины

Участок постоянного управления — промежуток времени, характеризующийся постоянством ускорений ВС. На нём фиксируются: u_i (поле записи `.Upr`) — значение поперечного управления; w_i (поле `.Wpr`) — значение продольного управления; t_{ni} (`.UTn`) — время начала участка постоянного управления; t_{ki} (`.UTk`) — время конца участка постоянного управления.

Трек управления — двунаправленный список участков постоянного управления. Каждый участок постоянного управления в треке управления должен содержать ссылки на предыдущий участок (поле записи `.Prd`) и последующий участок (поле `.Sld`). Величины t_{ni} и t_{ki} последовательных элементов списка должны быть согласованы, т. е. $t_{ki-1} = t_{ni}$. Также участки постоянного управления могут содержать поля `.nTr` и `.kTr` — ссылки на участок геометрического трека (определяется ниже), который был порождён рассматриваемым участком постоянного управления.

Замер РЛС содержит поля: `.Xzr` — координата замера «на север» в плоскости Земли; `.Zzr` — координата замера «на восток» в плоскости Земли; `.Hzr` — высота замера над плоскостью Земли; `.Tzr` — время замера; `.Nrls` — номер РЛС замера.

Трек замеров — двунаправленный список замеров РЛС. Каждый замер РЛС в треке замеров должен содержать ссылки на предыдущий замер (поле записи `.Prd`) и последующий замер (поле `.Sld`).

Точка геометрического трека описывает положение ВС и содержит поля: `.X` — координата ВС «на север» в плоскости Земли; `.Z` — координата ВС «на восток» в плоскости Земли; `.H` — высота ВС над плоскостью Земли; `.Phi` — направление скорости ВС (угол по часовой стрелке относительно направления на север); `.V` — величина скорости ВС. Время точки геометрического трека в необходимых случаях вычисляется используя её положение в геометрическом треке.

Геометрический трек — двунаправленный список точек геометрического трека. Каждая точка геометрического трека должена содержать ссылки на предыдущий замер (поле записи `.Prd`) и последующий замер (поле `.Sld`). Вычисление геометрического трека по треку управления возможно при помощи процедуры `PostrTrekaSTekUchUprPrymo()`. Аргументом этой процедуры является первый элемент списка трека управления, этот элемент должен иметь поле `.nTr`, указывающее на точку геометрического трека. Эта точка используются как начальная при интегрировании уравнений движения ВС.

2.5 Вычисление веса треков и меры расхождения треков

Вес траектории используется для сравнения текущих треков с целью отбора наилучшего при условии их "близости"(меры расхождения). Вес трека характеризует апостериорную вероятность движения по заданной совокупности замеров. Численно мера расхождения треков рассчитывается как среднее расстояние между треками в геометрических координатах, рассчитываемое на равномерной сетке времени в текущем расчётном окне.

2.6 Алгоритм оптимизации

Алгоритм оптимизации основывается на методе Хука – Дживса [??].

В качестве входной информации алгоритм получает ссылку на трек управления (последовательность участков постоянного управления).

Алгоритм варьирует: значения продольного управления w_i, значения поперечного управления u_i, время переключения между участками постоянного управления t_{ni}. При этом учитываются ограничения: на абсолютные значения управления, на минимальную продолжительность постоянного управления. Целью варьирования является построение последовательности управлений, которые бы определяли геометрический трек с минимальным весом.

Для вычисления веса трека алгоритм формирует временный трек управления. В процессе формирования временного трека происходит проверка нарушения ограничений на управление, если ограничения нарушаются, то происходит возврат в основной алгоритм Хука – Дживса, при этом в качестве значения минимизируемой функции возвращается штраф пропорциональный величине нарушения ограничения (при этом при подаче на вход алгоритма оптимизации трека управления с нарушением ограничений возможно «скатывание» алгоритма в область, где ограничения не нарушаются). Для построения геометрического трека по сформированному треку управления используется обращение к процедуре `PostrTrekaSTekUchUprPrymo()` (которая производит интегрирование), затем вес трека вычисляется обращением к функции `RaschetVesaTreka()` и происходит возврат в основной алгоритм алгоритм Хука – Дживса с возвратом веса трека в качестве значения минимизируемой функции.

Варьирование прекращается когда текущие шаги варьирования оказываются меньше заданных финальных шагов варьирования.

Алгоритм возвращает ссылку на новый трек управления, получившийся в результате варьирования. В свою очередь трек управления содержит ссылки на соответствующий геометрический трек.

Константы-параметры алгоритма оптимизации

Алгоритм использует следующие постоянные параметры (константы):
`Du1 = 0.5` – начальный шаг варьирования поперечного управления;
`Dw1 = 0.25` – начальный шаг варьирования продольного управления;
`Dt1 = 32.0` – начальный шаг варьирования разбивки времени;
`DuFin = 0.01` – финальный шаг варьирования поперечного управления;
`DwFin = 0.01` – финальный шаг варьирования продольного управления;
`DtFin = 0.02` – финальный шаг варьирования разбивки времени;
`MAXu = 4` ($u_{\max}$) – максимальное значение поперечного управления;
`MAXw = 2` ($w_{\max}$) – максимальное значение продольного управления;
`dtmin = 10.0` – минимальный промежуток времени постоянного управления;
`dh = 0.5` – множитель уменьшения шага при неудаче в «поиске вокруг базовой точки» метода Хука – Дживса.

2.7 Программа конвертации данных

Программа `tracks_plots_00` предназначена для фильтрации и конвертации данных (РЛС, АЗН-В, монорадарная обработка, мультирадарная обработка) из текстовых файлов `tracks.txt`, `plots.txt`, `plots_ads.txt`, получаемых при помощи программы `vidparser.exe` из файлов `.vid`. Цель конвертации — получить небольшие по объёму текстовые файлы данных для использования в программе многогипотезного восстановления траектории.

В связи с тем, что исходные файлы имеют очень большой объём (гигабайты), полная загрузка информации из файлов в оперативную память (при использовании 32-битной ОС и 32-битного компилятора) не представляется возможной. Используются неоднократное чтение файлов и приближённая оценка сверху количества замеров в РЛС-треках.

Алгоритм

- Первое чтение файла `tracks.txt`, сбор общей статистики по источникам, глобальным идентификаторам трека и т.п.
- Выделение памяти для АЗН-треков.
- Чтение файла `plots_ads.txt`. Заполнение массивов замеров АЗН.
- Упорядочение замеров внутри АЗН-треков по времени, запись треков в файлы `ads.new4` и `ads.plt`.
- Освобождение памяти для АЗН-треков.
- Выделение памяти (на основе приближённой оценки) для монорадарных треков (моно-треков), треков мультирадарной обработки (мульти-треков), РЛС-треков.

- Второе чтение файла `tracks.txt`. Заполнение массивов моно-треков и мульти-треков, заполнение номеров замеров в РЛС-треках.
- Упорядочение замеров внутри моно-треков и мульти-треков по времени, запись треков в файлы `*_mr.new4` и `*_mr.plt`.
- Чтение файла `plots.txt`. Заполнение массивов замеров РЛС-треков.
- Упорядочение замеров внутри РЛС-треков по времени, запись треков в файлы `*_r.new4` и `*_r.plt`.

Особенности текущей версии, параметры программы

Воздушные суда идентифицируются по параметру `global` (глобальный номер трека), при этом в обработку идёт не более `AZN_TRACKS_MAX` воздушных судов, АЗН-треки которых имеют не менее `AZN_TRACKS_MIN_MEASUR` замеров (при этом из подходящих выбираются АЗН-треки с наибольшим числом замеров).

Из файлов `tracks.txt`, `plots.txt` учитываются РЛС-замеры и моно-замеры, параметр `sensor` которых равен параметру `RADAR_SENSOR_ONLY`.

Если после всех фильтраций трек содержит меньше `TRACKS_MIN_MEASUR` замеров, то он не записывается.

Выходные файлы записываются в поддиректорию `new`, которая не должна существовать до запуска программы. В директории `new` создаются поддиректории, совпадающие с параметром `global` записываемых ВС. Записываются файлы `ads.new4` и `ads.plt` (АЗН-треки), `240_mr.new4` и `240_mr.plt` (мульти-треки), `*_mr.new4` и `*_mr.plt` (моно-треки), `*_r.new4` и `*_r.plt` (РЛС-треки). Здесь `*` — номер РЛС.

Файлы `*.plt` имеют формат треков программы Ozi Explorer и используются для визуализации треков при помощи программы GPSMapEdit.

Формат файлов `.new4`

Текстовые файлы, в которых построчно записаны замеры. Столбцы: время замера, широта замера, долгота замера, высота замера. Столбцы разделяются символом табуляции.

Для РЛС-треков в качестве координат замеров используются поля `lat` и `lon` файла `plots.txt`, т.е. широта и долгота, сформированные из дальности и азимута сырых РЛС-замеров, которые предварительно корректируются с учётом текущей оценки систематических ошибок в программе мультирадарной обработки.

Литература

[1] Бедин Д. А., Денисов А. П., Иванов А. Г., Федотов А. А., Черетаев И. В., Ганебный С. А., Васильев А. В. “Одновременное определение координат движущегося ВС и коррекция систематических ошибок РЛС при помощи фильтра Калмана,” Тех. отчет, ИММ УрО РАН, 2015.

[2] Blom, H. A. P. and Van Doorn, B. A., “Systematic Error Estimation in Multisensor Fusion Systems,” *Proceedings of SPIE — The International Society for Optical Engineering*, Vol. 1954, Oct. 1993, pp. 450–461.