

# Оглавление

<b>1</b>	<b>Одновременное оценивание движения ВС и систематических ошибок. Алгоритмы параллельной фильтрации процессов, связанных через измерения</b>	<b>2</b>
1.1	Описание задачи наблюдения за ВС . . . . .	3
1.2	Уравнения оптимальной фильтрации . . . . .	6
1.2.1	Полная система . . . . .	6
1.3	Упрощенные алгоритмы оценивания по Henk Blom . . . . .	7
1.3.1	Фильтр Калмана для фазового вектора, Макро фильтр для систематической ошибки . . . . .	7
1.3.2	Разделённые фильтры для фазового вектора и для систематической ошибки . . . . .	8
<b>2</b>	<b>Алгоритм многогипотезного восстановления траектории</b>	<b>9</b>
2.1	??? Название ??? . . . . .	9
2.2	Математическая модель ??? . . . . .	9
2.3	Структура данных программы, термины . . . . .	9
2.4	Вычисление веса траектории ??? . . . . .	10
2.5	Алгоритм оптимизации . . . . .	10
2.6	Программа конвертации данных . . . . .	10
	<b>Литература</b>	<b>13</b>

# 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, находящихся 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 измерениями. 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:

$$z_{al}(t) = C^x(t)\chi_a(t) + C^s(t)\varsigma_l(t) + D(t)w_l(t). \quad (1.1)$$

Здесь  $t$  — момент времени;  $a$  — индекс, обозначающий номер воздушного судна (aircraft);  $l$  — индекс радиолокатора (locator);  $z_{al}$  — вектор измерения;  $\chi_a$  — вектор параметров движения ВС;  $\varsigma_l$  — вектор параметров, характеризующий состояние РЛС;  $w_l(t)$  — текущая реализация случайной ошибки РЛС;  $C^x(t)$ ,  $C^s(t)$ ,  $D(t)$  — матрицы, характеризующие вклад 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 переменные, 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 вычислительные затраты.

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, при этом обладающим низким уровнем погрешности оценивания. Исследование логически продолжает исследование, изложенное в отчёте [1].

## 1.1 Описание задачи наблюдения за ВС

Считаем, что каждое воздушное судно подчиняется независимому, но одному и тому же 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  $\chi_i$  — 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. В силу того, что наблюдение за ВС ведётся «в большом масштабе», вектор  $\chi_i$  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

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

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 ИММ уравнения движения используются именно в виде (1.2). Далее, будем рассматривать более простую линейную динамику без управления

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

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

В качестве основного варианта при моделировании будем выбирать прямолинейное равномерное движение на плоскости

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

где  $x_i, v_i$  обозначают векторы координат и скорости на плоскости  $\mathbb{R}^2$ .

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

$$z_{ij}(t) = C^x(t_k)\chi_i(t_k) + C_j^s(t_k, \chi_i(t_k))\varsigma_j(t_k) + D_j(t_k, \chi_i(t_k))w_j(t_k). \quad (1.5)$$

Матрицы  $C_j^s, D_j$  для всех имеющих смысл случаев зависят от положения ВС, поэтому явно указывается зависимость от  $\chi_i$ . В качестве параметров  $\varsigma_j$  могут выступать постоянная систематическая ошибка по дальности и азимуту, коэффициент линейной зависимости для систематической ошибки по дальности и т.д. Матрица  $C_j^s$  описывает влияние этих неизвестных параметров на измерения.

Для параметров  $\varsigma_j$ , характеризующих систематические ошибки РЛС, также введём динамику

$$\varsigma_j(t_k) = A_j^s(t_k)\varsigma_j(t_{k-1}) + B_j^s(t_k)v_j^s(t_k). \quad (1.6)$$

Матрица  $B_i^s$  характеризует дрейф систематических ошибок со временем. Для моделирования будем принимать:

$$\varsigma_j = \begin{bmatrix} \Delta_j^r \\ \Delta_j^\alpha \end{bmatrix}, \quad A_j^s(t_k) \equiv I_{2 \times 2}, \quad B_i^s(t_k) \equiv 0_{2 \times 2}. \quad (1.7)$$

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

Рассмотрим общий фазовый вектор

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

Здесь  $n$  и  $m$  — количества наблюдаемых ВС и наблюдающих радиолокаторов. Урав-

нения (1.3), (1.6) можно переписать как

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

где матрицы  $A$  и  $B$  представляют собой блочно-диагональные матрицы, объединяющие все  $A_i$ ,  $A_i^\varsigma$  и  $B_i$ ,  $B_i^\varsigma$ .

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.

$$\begin{aligned} z(t_k) &= z_{ij}(t_k) = C(t_k, \xi(t_k))\xi(t_k) + D(t_k, \xi(t_k))w(t_k), \quad (1.10) \\ C(t_k, \xi(t_k)) &= \begin{pmatrix} 0 & \dots & 0 & C^x(t_k) & 0 & \dots & 0 & C_j^\varsigma(t_k, \chi_i(t_k)) & 0 & \dots & 0 \end{pmatrix}, \\ D(t_k, \xi(t_k)) &= D_j(t_k, \chi_i(t_k)), \quad w(t_k) = w_j(t_k). \end{aligned}$$

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  $C^x(t_k)$ ,  $C^\varsigma(t_k, \chi_i(t_k))$ ,  $D(t_k, \chi_i(t_k))$  имеют вид:

$$\begin{aligned} C^x(t_k) &\equiv I_{2 \times 2}, \quad C_j^\varsigma(t_k, \chi_i(t_k)) = \left[ \frac{1}{\|x_i(t_k) - x_j^R\|} (x_i(t_k) - x_j^R) \quad \Omega_{2 \times 2} (x_i(t_k) - x_j^R) \right], \\ D(t_k, \chi_i(t_k)) &= C_j^\varsigma(t_k, \chi_i(t_k)), \quad \Omega_{2 \times 2}^{\frac{\pi}{2}} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \end{aligned}$$

Здесь  $x_j^R$  — координаты точки стояния радиолокатора  $j$ ;  $\Omega_{2 \times 2}^{\frac{\pi}{2}}$  — 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

$$\mathbb{E}\{v_i(t_k)\} = \mathbb{E}\{v_i^s(t_k)\} = \mathbb{E}\{w_j(t_k)\} = \mathbb{E}\{w_j^s(t_k)\} = 0, \quad (1.11)$$

$$\mathbf{Cov}\{v_{i_1}(t_k), v_{i_2}(t_l)\} = \delta_{kl}\delta_{i_1 i_2} V, \quad \mathbf{Cov}\{w_{j_1}(t_k), w_{j_2}(t_l)\} = \delta_{kl}\delta_{j_1 j_2} W_{j_1}, \quad (1.12)$$

где  $\delta_{pq}$  — символ Кронекера;  $V$  — постоянная матрица дисперсии случайных возмущений уравнений движения;  $W_j$  — постоянная матрица дисперсии случайных ошибок наблюдения для радиолокатора  $j$ .

Для моделирования будем применять  $W_j$  вида:

$$W_j = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\alpha^2 \end{bmatrix}, \quad (1.13)$$

где  $\sigma_r$ ,  $\sigma_\alpha$  — заданные среднеквадратичные отклонения для случайных ошибок наблюдения по дальности и азимуту, соответственно.

## 1.2 Уравнения оптимальной фильтрации

### 1.2.1 Полная система

Фильтр для фазового вектора.

Этап предсказания:

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

Этап коррекции:

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

Фильтр для систематической ошибки.

Этап предсказания:

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

Этап коррекции:

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

Обновление блока кросс-ковариации:

$$\begin{aligned} \bar{P}_{xs,t} &= A_x \hat{P}_{xs,t-1} A_s^T \\ \hat{P}_{xs,t} &= \bar{P}_{xs,t} - K_x \Lambda K_s^T \end{aligned}$$

В данном случае для обоих фильтров используется одна матрица  $\Lambda$ :

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

Уравнение наблюдения:

$$z_t = C_x x_t + C_s s_t + D w$$

### 1.3 Упрощенные 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.

#### 1.3.1 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:

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

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

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  $\Lambda_s$  используются аппроксимация члена  $C_x \bar{P}_{xs,t}$ :

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

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

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:

$$z_t = C_x x_t + C_s s_t + D w$$

### 1.3.2 Разделённые фильтры для фазового вектора и для систематической ошибки

Фильтр для фазового вектора.

Этап предсказания:

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

Этап коррекции:

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

Аппроксимация:

$$\begin{aligned}K_x &= \bar{P}_{x,t} C_x^T \\ \Lambda_x &= C_x \bar{P}_{x,t} C_x^T + D D^T\end{aligned}$$

Фильтр для систематической ошибки.

Этап предсказания:

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

Этап коррекции:

$$\begin{aligned}\hat{s}_t &= \bar{s}_t + K_s \Lambda_s (z_t - C_x \bar{x}_t - C_s \bar{s}_t) \\ \hat{P}_{s,t} &= \bar{P}_{s,t} - K_s \Lambda_s K_s^T\end{aligned}$$

Аппроксимация:

$$\begin{aligned}K_s &= \bar{P}_{s,t} C_s^T \\ \Lambda_s &= C_s \bar{P}_{s,t} C_s^T + D D^T\end{aligned}$$

Уравнение наблюдения:

$$z_t = C_x x_t + C_s s_t + D w$$



## 2 Алгоритм многогипотезного восстановления траектории

### 2.1 ??? Название ???

### 2.2 Математическая модель ???

??? слова про плоскость / плоскую Землю ???

Математическая модель движения воздушного судна 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;  $\varphi$  — угол наклона вектора скорости, отсчитываемый по часовой стрелке от вертикальной оси  $x$ ;  $v$  — величина скорости;  $u$  — боковое ускорение;  $w$  — продольное ускорение.

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

...

### 2.3 Структура данных программы, термины

Участок постоянного управления — промежуток времени, характеризующийся постоянством ускорений ВС. 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). Поля  $t_{ni}$  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), порождённый рассматриваемым треком 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; .Nr1s — 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 (азимут по часовой стрелке относительно направления север); *.V* — величина скорости ВС.

?? Где-то тут нужны слова про равномерную сетку времени...

Геометрический трек — двунаправленный список точек геометрического трека. Каждая точка геометрического трека должна 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. Значения полей этой точки используются как начальная точка при интегрировании уравнений движения ВС.

## 2.4 Вычисление веса траектории ???

## 2.5 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 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.

## 2.6 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 GPSTMapEdit.

### 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.