

Об экономической схеме сглаживания данных в расчетах в реальном времени

А.В. ГАЙНУТДИНОВА,
аспирант,

В.Г. ГАЙНУТДИНОВ,

д-р техн. наук

(КНИТУ-КАИ, Казань),

Т.Ю. ГАЙНУТДИНОВА,

канд. техн. наук

(КФУ, Казань)

gainut@mail.ru

Предлагается алгоритм сглаживания данных в реальном времени за счет существенного сокращения и упрощения вычислений с использованием рекуррентного метода наименьших квадратов. Работоспособность алгоритмов показана на примере определения воздушной скорости и высоты полета по данным барометрических датчиков.

Схема сглаживания данных, рекуррентная схема метода наименьших квадратов

Задачи фильтрации сигналов в навигационных расчетах с использованием инерциальных навигационных и спутниковых навигационных систем (ИНС и СНС) обычно связывают с различными схемами комплексирования ИНС и СНС [1], основанными на применении интегрального фильтра Калмана для снижения накопления погрешностей ИНС, с одной стороны, и шумов СНС, с другой. Однако задача предварительного сглаживания сигналов с сенсоров может решаться и отдельно [2], в том числе для обеспечения плавности управления приводами и механизмами [3–5]. Необходимо только обеспечить предельное упрощение дополнительных расчетов и минимальную дополнительную загрузку бортового процессора, не задействуя его ресурсы для проведения основных расчетов ориентации и навигации. Один из таких экономичных алгоритмов и практическое его применение приводится в данной статье.

При расчете сигналов для органов управления беспилотным летательным аппаратом (ЛА) по данным различных сенсоров (датчика воздушной скорости, датчика давления, координат и скорости движения по GPS и т.п.) иногда также возникает необходимость провести предварительное сглаживание поступающих «сырых» данных, искаженных аппаратными шумами. Все расчеты при этом должны производиться за предельно короткое время.

В качестве примера покажем «сырые» данные, поступившие с датчика воздушной скорости при разгоне до 70 км/ч в штиль (рис. 1), и данные по скорости, полученные по GPS, синхронизированные по времени (рис. 2). Темп получения данных по радиоканалу на наземную станцию с датчиков, установленных на автомобиле, составлял 20 Гц, от GPS – 2 Гц.

Искаженные шумами данные сенсоров необходимо сглаживать. При вычислительных процедурах сглаживания не должно происходить запаздывания поступления данных для расчетов навигации и управления ЛА.

Как правило, для этих целей используется метод наименьших квадратов (МНК) [6], нашедший широкое применение в задачах адаптивной обработки сигналов (фильтрации) в радиотехнике. Этот метод является самым простым с точки зрения вычислительной сложности (требуемого числа арифметических операций), не сложен для понимания (описывается всего лишь несколькими уравнениями), прост в реализации.

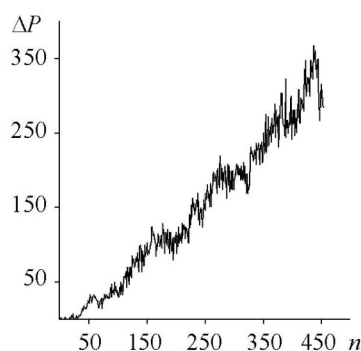


Рис. 1

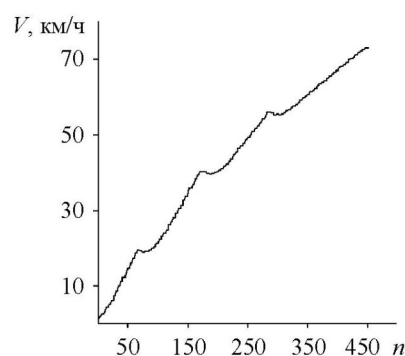


Рис. 2

Все дополнительные вычисления по предварительному сглаживанию сигналов должны укладываться в минимальный временной интервал с учетом ограниченных аппаратных ресурсов бортового процессора. Для этой цели предлагается доработать идею локальной аппроксимации максимально возможным сокращением вычислительных операций.

За основу возьмем схему скользящего окна [2, 7], в соответствии с которой используется ограниченное число данных (рис. 3).

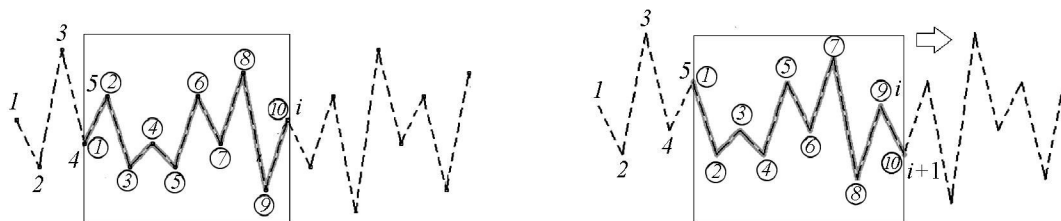


Рис. 3

В соответствии с этой схемой при равномерном по времени поступлении данных в условное окно определенного размера, равного числу одновременно помещающихся в нем данных $n = n_w + 1$, поступает очередное i -е значение и производятся необходимые расчеты по сглаживанию.

Рассмотрим возможность минимизации вычислений в этой схеме на примере аппроксимации данных кубическим полиномом

$$A_0 + A_1 x_i + A_2 x_i^2 + A_3 x_i^3 = y_i. \quad (1)$$

Разрешающее матричное уравнения МНК представляется выражением

$$\begin{Bmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \end{Bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12} & a_{22} & a_{23} & a_{24} \\ a_{13} & a_{23} & a_{33} & a_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{bmatrix}^{-1} \begin{Bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{Bmatrix}. \quad (2)$$

При равномерном поступлении данных во времени с шагом dt

$$x(i) = (i-1)dt, \quad i = 1, n, \quad (3)$$

а компоненты матричного уравнения (2) определяются следующим образом:

$$\begin{aligned} a_{11} &= n, \quad a_{12} = a_{21} = dt \sum_{i=1}^{n-1} i, \quad a_{13} = a_{22} = a_{31} = (dt)^2 \sum_{i=1}^{n-1} i^2, \\ a_{14} &= a_{41} = a_{23} = a_{32} = (dt)^3 \sum_{i=1}^{n-1} i^3, \quad a_{24} = a_{42} = a_{33} = (dt)^4 \sum_{i=1}^{n-1} i^4, \\ a_{34} &= a_{43} = (dt)^5 \sum_{i=1}^{n-1} i^5, \quad a_{44} = (dt)^6 \sum_{i=1}^{n-1} i^6, \\ C_0 &= c_0 = \sum_0^n y_i, \quad C_1 = dt \sum_0^n y_i i = c_1 dt, \quad C_2 = dt^2 \sum_0^n y_i i^2 = c_2 dt^2, \\ C_3 &= dt^3 \sum_0^n y_i i^3 = c_3 dt^3. \end{aligned} \quad (4)$$

Для сокращения времени вычислений a_{ij} будем использовать известные формулы

$$\sum_{i=1}^n i = \frac{n^2 + n}{2}; \quad \sum_{i=1}^n i^2 = \frac{2n^3 + 3n^2 + n}{6}; \quad \sum_{i=1}^n i^3 = \frac{n^4 + 2n^3 + n^2}{4} \text{ и т.д.}$$

В данном случае операция суммирования в формулах (4) для определения c_0, c_1, c_2, c_3 заменяется рекуррентными вычислениями

$$\begin{aligned} c_3 &= c_3 - 3c_2 + 2c_1 - c_0 + n_w + 1 + y(i)n_w^3, \\ c_2 &= c_2 - 2c_1 + c_0 - y_w(0) + y(i)n_w^2, \end{aligned} \quad (5)$$

$$\begin{aligned}c_1 &= c_1 - c_0 + y_w(0) + y(i)n_w, \\c_0 &= c_0 - y_w(0) + y(i),\end{aligned}\quad (5)$$

где $y_w(0)$ – первое значение в «окне» (слева); $y(i)$ – вновь полученное значение (справа).

Заметим, что матрица $[a_{ij}]^{-1}$ вычисляется один раз, и количество арифметических операций (5) невелико, но и их можно свести к минимуму. Рассмотрим это на примере линейной аппроксимации. Положим временной интервал по длине «окна» равным единице (рис. 4). Тогда можно принять $dt = 1/n_w$ и представить разрешающее уравнение МНК в следующей форме:

$$\begin{Bmatrix} A_0 \\ A_1 \end{Bmatrix} = \frac{1}{D} \begin{bmatrix} 4n^3 + 6n^2 + 2n & -6(n^3 + n^2) \\ -6(n^3 + n^2) & 12(n^3 + n^2) \end{bmatrix} \begin{Bmatrix} c_0 - y_w(0) + y(i) \\ c_1 - c_0 + y_w(0) + n_w y(i) \end{Bmatrix}, \quad (6)$$

где $D = n^4 + 4n^3 + 5n^2 + 2n$.

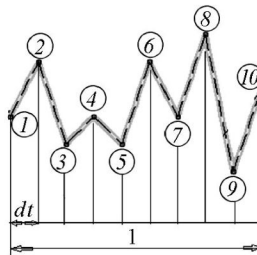


Рис. 4

В итоге сглаженное значение $y_{\text{filtr}}(i)$ можно будет вычислить по формуле

$$y_{\text{filtr}}(i) = \left(\frac{1}{D} \right) \left[2(n - n^3)(c_0 - y_w(0) + y(i)) + 6(n^2 + n)(c_1 - c_0 + y_w(0) + n_w y(i)) \right]. \quad (7)$$

Дополнительно необходимо будет еще произвести сдвиг данных влево $y_w(k-1) = y_w(k)$, $k = 2, n_w + 1$, $y_w(n_w + 1) = y(i)$ для завершения всей процедуры необходимых вычислений. Заметим, что вычисление величин $\left(\frac{1}{D} \right)$, $2(n - n^3)$, $6(n^2 + n)$ в формуле (7) происходит один раз и до начала поступления данных.

Вычислительные операции (7) не занимают много времени и аппаратных ресурсов (используется сложение и умножение) для эффективного сглаживания поступающих данных с датчиков в алгоритмах расчета в реальном времени.

Покажем результаты сглаживания данных датчика воздушной скорости (рис. 5) и сопоставление их с данными GPS по скорости (рис. 6). Такое сопоставление позволяет произвести калибровку датчика воздушной скорости и составить алгоритм вычисления воздушной скорости по сглаженным данным ΔP_{smooth} . При этом есть возможность избежать вычисления корня квадратного. Квадратичная $\Delta P = A_0 + A_1 V + A_2 V^2$ и линейная аппроксимация $\Delta P = A_0 + A_1 V$ дает возможность организовать следующую итерационную процедуру по вычислению скорости полета по данным датчика динамического давления

$$V_{(0)} = \frac{\Delta P - A_0}{A_1}; V_{(k+1)} = \frac{\Delta P - A_0}{A_1 + A_2 V_{(k)}}, \quad k = 1, 4. \quad (8)$$

Итерационная процедура (8) дает возможность вычислить скорость по значению ΔP_{bar} за две-четыре итерации. Использование только квадратичной аппроксимации приводило бы к увеличению числа необходимых итераций до семи-восьми

$$V_{(0)} = 0, V_{(k+1)} = \frac{\Delta P_{\text{bar}} - A_0}{A_1 + A_2 V_{(k)}}, \quad k = 1, 8, \quad (9)$$

что уже сравнимо по затратам времени с вычислением корня квадратного.

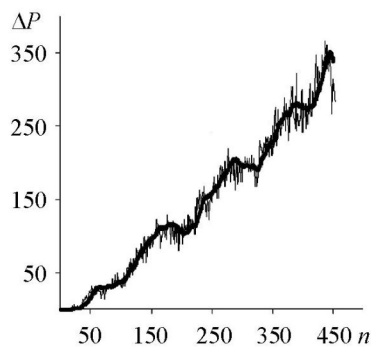


Рис. 5

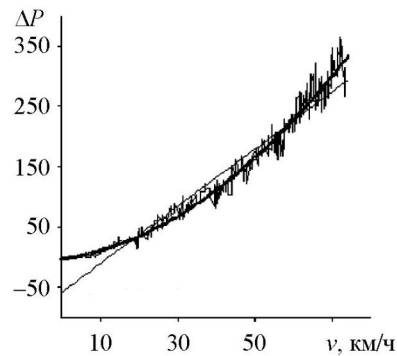


Рис. 6

Для того чтобы вычислить истинную воздушную скорость V при других температурах T_H и атмосферном давлении P_H , необходимо зафиксировать температуру T_0 и давление P_0 при тестовых калибровочных испытаниях. Для расчета истинной воздушной скорости достаточно пересчитать получаемые значения динамического давления $\Delta P = \Delta P_H \frac{P_0 T_H}{P_H T_0}$ и затем вычислить скорость по формуле (8).

Расчеты показали, что наибольший эффект сглаживания дает линейная аппроксимация. Приведем результаты линейной и квадратичной аппроксимации данных GPS по скорости (рис. 7, кривая 1). Линейная аппроксимация уже при $n_w = 9$ (рис. 7, кривая 2) дает заметное сглаживание, а квадратичная (рис. 7, кривая 3) – еще нет.

Барометрический датчик, входящий в стандартный состав ИНС, используется для определения давления и служит источником данных для расчета высоты полета. Изменение давления по высоте $P(H)$ определяется формулой

$$P(h) = P_0 e^{-\frac{Mg}{RT} H},$$

где P_0 – давление воздуха на уровне моря; M – молярная масса воздуха; T – абсолютная температура; R – универсальная газовая постоянная; g – ускорение свободного падения. Данные относительного давления $\Delta = P/P_0$ в зависимости от высоты взяты из таблицы стандартной атмосферы [8]. Аппроксимация этих данных до высоты 5000 м кубической параболой $H = H(\Delta)$ по МНК показывает хорошее совпадение справочных и рассчитанных значений (рис. 8).

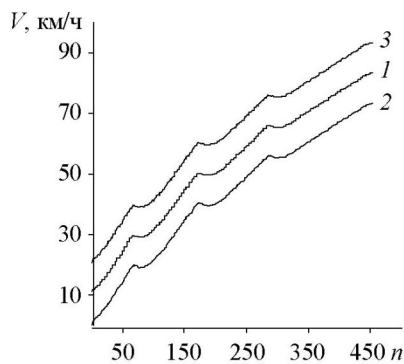


Рис. 7

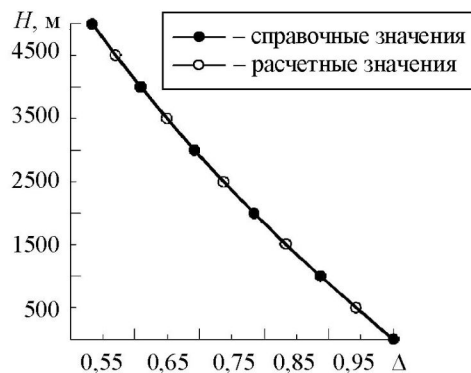


Рис. 8

Отличие справочных и расчетных значений высоты H не превышает 0,2 %.

Для того чтобы рассчитать изменение высоты H относительно произвольной высоты точки старта H_{origin} по относительному давлению $\Delta = P/P_{\text{origin}}$, на основании зависимости

$$H - H_{\text{origin}} = A(1)\Delta + A(2)\Delta^2 + A(3)\Delta^3 - [A(1) + A(2) + A(3)]$$

определим коэффициенты кубической параболы как функцию высоты точки старта над уровнем моря в следующей форме:

$$A(i) = k(i,0)/100 + (k(i,1)H_{\text{origin}} + k(i,2)H_{\text{origin}}^2 + k(i,3)H_{\text{origin}}^3)/10. \quad (10)$$

Коэффициенты $k(i,0)$ были рассчитаны для высот точки старта H_{origin} в диапазоне 0...1500 м и представлены целыми числами (таблица).

Значения коэффициентов	$k(_,0)$	$k(_,1)$	$k(_,2)$	$k(_,3)$
$A(1)$	-2579272	-16976	99452	-56854
$A(2)$	1413042	23225	-114280	65091
$A(3)$	-361757	-9283	42938	-24331

При использовании таблицы высота точки старта H_{origin} должна быть выражена в километрах, т.е. для $H_{\text{origin}} = 200\text{ м}$ в формуле (10) необходимо подставить $H_{\text{origin}} = 0,2\text{ км}$.

В соответствии с этим в точке старта фиксируется показание бародатчика P_{origin} , по GPS определяется высота над уровнем моря H_{origin} , температура T_{origin} . По формуле (10) вычисляются соответствующие коэффициенты $A(i)$. В полете определяется отношение текущего показания бародатчика $\Delta = P/P_{\text{origin}}$ и вычисляется высота относительно точки старта $H - H_{\text{origin}}$ в метрах.

Приведем «сырые» данные бародатчика при равномерном подъеме на высоту 42 м (рис. 9) и сглаженные при размере «окна» $n_w = 9$ (рис. 10). «Сырые» и сглаженные данные при размере окна $n_w = 49$ (рис. 11, а) и $n_w = 24$ (рис. 11, б) на меньшем перепаде высот (0–7 м) демонстрируют достаточно эффективную линейную аппроксимацию.

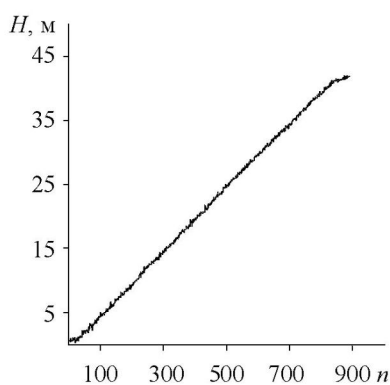


Рис. 9

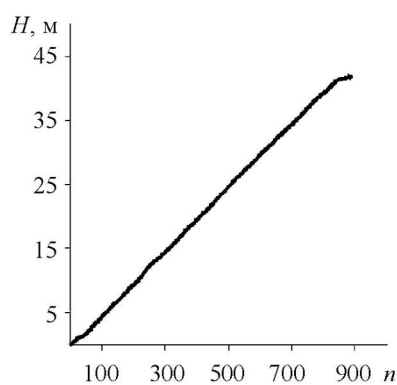
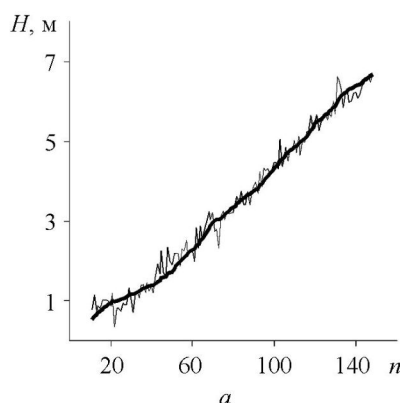
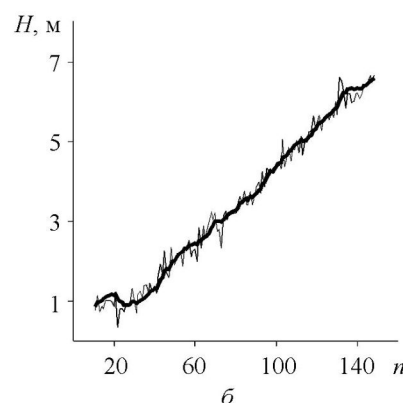


Рис. 10



а



б

Рис. 11

«Сырые» и сглаженные данные вертикальных колебательных перемещений показаны на рис. 12. Размер «окна» $n_w = 74$ дает хорошее сглаживание, но слишком большое запаздывание для использования в расчетах в реальном времени. Размер «окна» $n_w = 24$ (рис. 13) дает адекватное сглаживание данных, запаздывание практически отсутствует.

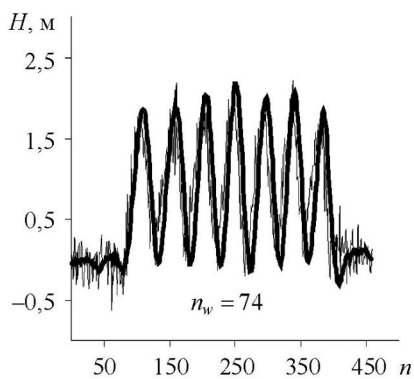


Рис. 12

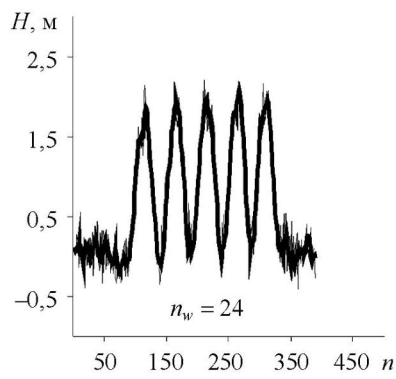


Рис. 13

Практическая реализация предложенных схем вычислений на модели конвертоплана добавила значительную плавность в характер поведения ЛА на режиме висения с автоматическим выдерживанием заданной позиции и высоты.

Таким образом, предложенная модернизация идеи скользящего МНК равномерно поступающих по времени данных снижает количество выполняемых операций при их предварительном сглаживании и позволяет использовать его в алгоритмах расчета в реальном времени. Практические приемы расчета воздушной скорости и высоты полета позволяют обеспечить приемлемую точность и меньшие требуемые аппаратные ресурсы бортового процессора благодаря простым вычислительным операциям сложения и умножения при их реализации.

СПИСОК ЛИТЕРАТУРЫ

1. Красильщиков М.Н., Себряков Г.Г. Современные информационные технологии в задачах навигации и наведения беспилотных маневренных летательных аппаратов. М.: Физматлит, 2009. 556 с.
2. Белинска В., Клуга А., Клуга Я. Применение среднеквадратического метода с меняющимися параметрами для увеличения точности GPS // Электроника и электротехника. 2010. № 8 (104). С. 109–112.
3. Ефремова Е.С. Имитатор системы воздушных сигналов самолета // Изв. вузов. Авиационная техника. 2016. № 2. С. 81–86.
4. Брусов В.С., Пиявский С.А. Многокритериальный анализ концепций высотных беспилотных летательных аппаратов // Изв. вузов. Авиационная техника. 2016. № 4. С. 9–12.
5. Кошевой Н.Д. и др. Авиационный комплекс для изменения угловых отклонений рулевых поверхностей летательного аппарата // Изв. вузов. Авиационная техника. 2013. № 4. С. 71–73.
6. Widrow B. Thinking about Thinking: The Discovery of the LMS Algorithm // IEEE Signal Processing Magazine. 2005. Vol. 22. № 1. P. 100–106.
7. Taylor J. An Introduction to Error Analysis. Sausalito: University Science Books, 1997. 349 p.
8. Горбатенко С.А. и др. Механика полета. М.: Машиностроение, 1969. 420 с.

Поступила в редакцию 20.11.15

An Efficient Algorithm of Data Smoothing in Real-Time Calculations

A.V. GAINUTDINOVA, V.G. GAINUTDINOV, AND T.YU. GAINUTDINOVA

We propose an algorithm for real-time data smoothing due to considerable reduction and simplification of computations by employing the recursive least mean squares technique. The efficiency of algorithms is shown by an example of determining the flight airspeed and altitude based on the data of pressure sensors.

Data smoothing algorithm, recursive least mean squares technique