УДК 629.7.054.07

Метод восстановления оценки параметров движения беспилотного летательного аппарата при идентификации факта помехового воздействия на бортовой приемник спутниковых навигационных сигналов

Исаев М. А., Линец Г. И.

Актуальность: навигационные задачи на подавляющем числе малых беспилотных летательных аппаратов (БЛА), решаются за счет комплексирования сигналов спутниковой радионавигационной системы (СРНС) и инерциальной навигационной системы. Эффективным методом противодействия БЛА является создание зон локального подавления сигналов навигационных спутников, что искажает оценку параметров движения БЛА, выполняемую бортовым вычислителем в режиме реального времени, и как следствие, отклоняет фактическую траекторию полета от заданной. Известные работы, описывающие алгоритмы функционирования БЛА в условиях помеховой обстановки, не рассматривают промежуток времени между фактическим началом помехового воздействия на приемник СРНС и идентификацией этого факта бортовым вычислителем, что не позволяет обеспечить требуемую для эксплуатации БЛА точность удержания заданной траектории полета при преодолении зон помехового воздействия. Целью работы является повышение точности оценки параметров движения при преодолении зон помехового воздействия за счёт исключения недостоверных данных бортового модуля СРНС из контура системы навигации за промежуток времени между фактическим началом помехового воздействия и идентификацией данного факта бортовым вычислителем. Используемые методы: решение поставленной задачи осуществлено за счет буферизации вектора невязки, который является разностью между оцененным состоянием параметров движения БЛА с помошью математической модели и параметрами, полученными от бортового модуля СРНС. При выполнении полетного задания в условиях достоверных сигналов навигационных спутников, предложенный метод выполняет буферизацию вектора невязки в режиме скользящего окна. После идентификации факта помехового воздействия, данные модуля СРНС не используются, а буферизированное значение вектора невязки вычитается из оценки вектора пространства состояний, тем самым исключая недостоверные данные СРНС и восстанавливая точность оценки параметров движения БЛА. В дальнейшем, до выхода из зоны помехового воздействия, используется инерциальный метод счисления пути для обновления параметров движения БЛА. Новизна: элементом новизны в данной работе является процедура буферизации вектора невязки для исключения недостоверных данных бортового модуля СРНС после идентификации факта помехового воздействия за промежуток времени, необходимый бортовому вычислителю для идентификации факта помехового воздействия. Результат: использование предложенного метода в системе навигации БЛА «Альбатрос СКФУ» позволило обеспечить точность позиционирования БЛА при преодолении зон помехового воздействия не хуже 3,337 м для северного и 0,869 м для восточного направлений после идентификации факта помехового воздействия, что уменьшило погрешность на 55,757 м и 5,273 м соответственно. Практическая значимость: решение задачи с учетом указанных особенностей позволило уменьшить ошибку позиционирования малых БЛА и обеспечить её инвариантность для временного промежутка, необходимого бортовому вычислителю для определения факта помехового воздействия. Представленный метод реализован на языке программирования Си в соответствии с стандартом СП и подходит для применения во встраиваемых системах, которые отличаются низкими вычислительными ресурсами, а также объемами постоянной и оперативной памяти.

Библиографическая ссылка на статью:

Исаев М. А., Линец Г. И. Метод восстановления оценки параметров движения беспилотного летательного аппарата при идентификации факта помехового воздействия на бортовой приемник спутниковых навигационных сигналов // Системы управления, связи и безопасности. 2021. № 4. С. 72-98. DOI: 10.24412/2410-9916-2021-4-72-98

Reference for citation:

Isaev M. A., Linets G. I. Method for recovering estimation of the motion parameters of an unmanned aerial vehicle when identifying the fact of interference on the on-board receiver of satellite navigation signals. *Systems of Control, Communication and Security*, 2021, no. 4, pp. 72-98 (in Russian). DOI: 10.24412/2410-9916-2021-4-72-98

Ключевые слова: беспилотный летательный аппарат, инерциальная навигационная система, помеховое воздействие, приемник сигналов спутниковой радионавигационной системы, фильтр Калмана.

Введение

С развитием малых беспилотных летательных аппаратов (БЛА) наблюдается тенденция к удешевлению их эксплуатации. Это приводит к расширению сфер применения, начиная от сельского хозяйства и заканчивая разведкой местности. В независимости от поставленной задачи, летательный аппарат должен обеспечить точность удержания траектории достаточную, для успешного выполнения полетного задания. Отклонение траектории от заданной можно представить как сумму погрешностей:

$$\Delta \vec{P}_{ouud\delta} = \Delta \vec{P}_{ouud\delta}^{ynpab} + \Delta \vec{P}_{ouud\delta}^{hab},$$

где: $\triangle \vec{P}_{ouud}^{ynpab}$ – ошибка удержания заданной траектории системой управления; $\triangle \vec{P}_{ouud}^{Hab}$ – ошибка оценки параметров навигации бортовым вычислителем.

В современных БЛА $\triangle \vec{P}_{ouud\delta}^{ynpab}$ зависит от частоты дискретизации бортового вычислителя, типа применяемого регулятора, ветровой нагрузки и в среднем варьируется от нескольких сантиметров до двух метров. Оценка параметров навигации выполняется инерциальным методом счисления пути [1-2], но из-за низкой точности используемых датчиков угловой скорости и линейных ускорений периодически корректируется по сигналам спутниковой радионавигационной системы (СРНС) [3]. Из необходимости коррекции следует, что $\triangle \vec{P}_{ouud\delta}^{hab}$ напрямую зависит качества сигналов навигационных спутников. В условиях беспомеховой обстановки погрешность оценки вектора местоположения не превышает двух метров, но при искажении СРНС может увеличиться до сотен метров [4-9]. Поэтому при идентификации бортовым вычислителем факта помехового воздействия, происходит отключение коррекции для обеспечения приемлемой точности позиционирования.

Современный уровень развития средств радиоэлектронного подавления, а также расширение областей применения малых БЛА, требует рассмотрения сценариев, в которых летательный аппарат преодолевает локальные зоны помехового воздействия. Для этого декомпозируем участки полета БЛА:

- 1. t₀ полет в условиях отсутствия помехового воздействия на СРНС;
- 2. t₁ попадание в зону помехового воздействия;
- 3. *t*₂ идентификация факта помехового воздействия бортовым вычислителем и отключение коррекции по сигналам СРНС;
- 4. $\triangle(t_3 t_2)$ полет без использования коррекции;
- 5. *t*₃ выход из зоны помехового воздействия и включение коррекции.

При своевременном определении факта помехового воздействия, погрешность позиционирования за промежуток $\triangle(t_3 - t_2)$ можно представить:

$$\Delta \vec{P}_{ouuo}^{HaB} = \int_{t_2}^{t_3} \int_{t_2}^{t_3} \Delta \vec{a}_{ouuo} dt ,$$

где: \vec{a}_{ouud} – усредненная погрешность выходного сигнала датчика линейных ускорений.

Для случая, когда $t_2 > t_1$, ошибка позиционирования увеличивается за счет использования недостоверных данных СРНС для коррекции:

$$\Delta \vec{P}_{ouud\delta}^{HaB} = \int_{t_2}^{t_3} \int_{t_2}^{t_3} \Delta \vec{a}_{ouud\delta} \, dt + \sum_{k=t_2}^{t_3} K_k \, \Delta CPHC_{ouud\delta} \,, \tag{1}$$

где: K – коэффициенты алгоритма коррекции, отвечающие за степень влияния данных СРНС на итоговую оценку местоположения; $\triangle CPHC_{ouudo}$ – усредненная погрешность модуля СРНС на участке помехового воздействия.

Из (1) следует, что при $(t_2 - t_1) \to \min$, погрешность, вносимая недостоверными сигналами СРНС будет минимизирована, т.е. $\sum_{k=t_2}^{t_3} K_k \triangle CPHC_{ouud} \to \min$,

что повысит точность позиционирования при полете в зоне помехового воздействия. Проведем анализ известного научно-методического аппарата, используемого бортовыми вычислителями БЛА для идентификации помехового воздействия на бортовой преемник СРНС.

Анализ предыдущих исследований в исследуемой предметной области

Для идентификации искажения навигационного поля используется информация от [2]:

- наземного контрольного сегмента навигационной системы;
- геостационарных спутников;
- аппаратуры контрольных станций и других навигационных систем с последующей передачей информации о целостности навигационных данных потребителям;

- собственно аппаратуры потребителей (способы автономного контроля).

Наибольший интерес для применения в БЛА представляют алгоритмы, реализующие вычисления на борту летательного аппарата, что позволяет определять локальные зоны искажения сигналов спутников навигации и иметь высокую оперативность идентификации помехового воздействия.

В работе [10] описан алгоритм автономного контроля целостности навигационного поля с использованием избыточной навигационной информации от сети спутников. Однако автором не указано время, необходимое для идентификации искажения навигационного поля при различных видах воздействий. В главе «Контроль целостности радионавигационного поля ГЛОНАСС» указано, что минимальное время определения контроля целостности НП составляет до 10 с. За этот временной промежуток, погрешность оценки местоположения БЛА может достигать нескольких сотен метров, что не удовлетворяет требованиям их практического применения.

Результаты моделирования, представленные в [7], идентифицируют факт искажения НП за счет суммирования вектора невязки. При этом погрешность

оценки местоположения БЛА может достигать нескольких сотен метров за счет обработки недостоверных данных, получаемых от модуля СРНС за временной промежуток, превышающий допустимую величину, что не удовлетворяет требованиям эксплуатации БЛА при решении практических задач.

В [8] предложен способ определения факта искажения и идентификации помехового воздействия навигационного поля. Для его реализации требуется накопление статистической информации размером не менее 30 выборок, что занимает временной отрезок 3 с при частоте получения данных от СРНС модуля 10 Гц. Предполагается зависание БЛА для идентификации помехового воздействия. Тем не менее, даже задержка в 3 с может привести к погрешности позиционирования до нескольких сотен метров в условиях сильного искажения сигналов СРНС.

В работе [11] использован алгоритм на основе линейного фильтра Калмана для оценки параметров движения БЛА. Он позволяет идентифицировать отказ одного из датчиков высоты в случае скачкообразного изменения выходного сигнала. В [11] утверждается, что при плавном нарастании погрешности измерений высотомера, используемый алгоритм только с 50% вероятностью отбраковывает недостоверные данные. Таким образом, рассматриваемая работа не применима для случаев, в которых происходит постепенное нарастание погрешностей измерителя и не позволяет на её основе построить алгоритм идентификации отказа бортового модуля СРНС с требуемым для практики качеством.

В работах [12-17], направленных на оценку параметров движения БЛА, не рассматриваются случаи, в которых система навигации идентифицирует недостоверные измерения модуля СРНС. Таким образом, можно сделать вывод о том, что существующий научно-методический аппарат либо не позволяет идентифицировать факт искажения НП за минимальное время с требуемой достоверностью в реальных сценариях применения БЛА, либо не рассматривает подобные сценарии их эксплуатации вовсе.

Для повышения летно-эксплуатационных характеристик малых БЛА при преодолении зон глушения/искажения сигналов навигационных спутников, необходимо разработать метод, позволяющий исключить недостоверные измерения модуля СРНС за временной промежуток, необходимый бортовому вычислителю для идентификации факта искажения с требуемой достоверностью.

Постановка задачи

Пусть коррекция инерциального метода счисления пути выполняется с помощью фильтра Калмана по данным бортового модуля СРНС, тогда алгоритм оценки параметров движения в общем виде можно разбить на два шага: прогноза и коррекции.

Шаг прогноза запишем в следующем виде [18]:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-|k-1} + B_k u_k,
P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k,$$
(2)

II û

где: x_{klk-1} – вектор пространства состояний, содержащий оценку параметров движения инерциальным методом; F – матрица перехода состояний, которая применяется к состоянию за предыдущий квант времени x_{k-k-1} ; u_k – вектор управления; B_k – матрица перехода вектора управления в вектор пространства состояний; $P_{k|k-1}$ – априорная ковариационная матрица; Q_k – матрица шумов шага прогноза.

Шаг коррекции предполагает использование следующих вычислений [18]:

$$\widetilde{y}_{k} = z_{k} - H_{k} \widehat{x}_{k|k-1},
S_{k} = H_{k} P_{k|k-1} H^{T} + R_{k},
K_{k} = P_{k|k-1} H^{T} S_{k}^{-1},
\widehat{x}_{k|k} = \widehat{x}_{k|k-1} + K_{k} \widetilde{y}_{k},
P_{k|k} = (I - K_{k} H_{k}) P_{k|k-1},$$
(3)

где: \tilde{y}_k – вектор невязки измерений; z_k – вектор, содержащий измерения бортового модуля СРНС; H_k – матрица приведения вектора $\hat{x}_{k|k-1}$ к размерностям вектора z_k ; S_k – ковариационная матрица вектора ошибки; R_k – матрица шумов измерителя; K_k – оптимальная по Калману матрица коэффициентов усиления.

На вербальном уровне задача исключения недостоверных данных модуля СРНС из оценки параметров движения БЛА может быть декомпозирована в следующем виде:

- буферизация вектора невязки $K_k \tilde{y}_k$ за промежуток времени, необходимый бортовому вычислителю для идентификации помехового воздействия с требуемой достоверностью;
- после идентификации факта искажения, исключить из $\hat{x}_{k|k}$ вектор $K_k \tilde{y}_k$ за указанный промежуток времени.

Предполагается, что для определения факта искажения с требуемой достоверностью используется способ, предложенный в [8]. Тогда, при идентификации помехового воздействия, необходимо:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} - \sum_{k} K_k \tilde{y}_k$$
, где: $\sum_{k} K_k \tilde{y}_k \in [t_1; t_2]$,

при ограничениях: $(t_2 - t_1) \le \Delta t_{max}$, где Δt_{max} – максимальное время, необходимое для идентификации факта помехового воздействия с требуемой достоверностью.

Решение задачи

Выразим буферизированные данные вектора невязки через сумму:

$$B_{\Delta t_{\max}} = \sum_{k=0}^{n-1} K_k \tilde{y}_k ,$$

где: $n = f_{cphc} \Delta t_{max}$; f_{cphc} – частота получения данных от модуля СРНС.

(4)

При обнаружении факта искажения, вычтем $B_{at_{max}}$ из $\hat{x}_{k|k-1}$:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} - B_{\Delta t_{\max}}.$$
(5)

До выхода из зоны помехового воздействия, данные СРНС не используются, поэтому уравнения (3) заменяются на $\hat{x}_{k|k} = \hat{x}_{k|k-1}$ т.к. отсутствует необходимость в шаге коррекции.

При выполнении полетного задания заранее неизвестно время начала помехового воздействия, однако, основываясь на способе идентификации помехового воздействия [8] и введенных ограничениях, зададим максимальное время, которое может затратить бортовой вычислитель для идентификации помехового воздействия Δt_{max} . В бортовом вычислителе буферизация $K_k \tilde{y}_k$ выполняется в режиме скользящего окна за промежуток Δt_{max} . Это гарантирует наличие вектора невязки в буфере для восстановления оценки параметров движения при идентификации помехового воздействия.

В качестве целевой платформы для разрабатываемого метода выступает бортовой электронный вычислитель БЛА, который относится к классу встраиваемых систем. Он имеет низкую стоимость, малое энергопотребление, ограниченные вычислительные ресурсы, а также малый объем оперативной и постоянной памяти. С целью минимизации потребляемой оперативной памяти, используем режим буферизации с дискретным сдвигом скользящего окна через промежутки Δt_{max} . Для этого используем двумерный массив:

$$B_{l\times 2} = \begin{pmatrix} b_{11} & b_{12} \\ \vdots & \vdots \\ b_{L1} & b_{L2} \end{pmatrix}$$
(6)

где l – длина вектора пространства состояний $\hat{x}_{k|k}$.

Каждый столбец в $B_{l\times 2}$ соответствует $B_{{}_{\Delta}t_{\max}}$. Время заполнения столбца составляет ${}_{\Delta}t_{\max}$,

Переключение индекса столбца в (6) выполняется через равные промежутки Δt_{max} для обеспечения буферизации в режиме скользящего окна. Так как буферизация выполняется через операцию суммирования, то величина Δt_{max} не влияет на размер требуемой памяти для хранения данных. В любой момент времени, данные в буфере соответствуют временному интервалу [Δt_{max} ; 2 Δt_{max}].

При обнаружении факта искажения, данные в буфере суммируются:

$$B_{tmp} = \sum_{i=0}^{l-1} \left(B_{i,1} + B_{i,2} \right).$$
(7)

Применение (7) для восстановления коррекции изменит параметры в векторе пространства состояний на B_{imp} , что дискретно увеличит рассогласование между заданной навигационной точкой полетного задания и текущей оценкой параметров движения БЛА. С целью уменьшения влияния восстановления коррекции на движение БЛА, увеличим время переходного процесса восстановле-

ния вектора пространства состояний. Для этого разделим B_{tmp} на скалярную величину, определяемую ($\Delta t_{max} \cdot f_{CPHC}$) + ($\Delta t_{idx_update} \cdot f_{CPHC}$). Получим:

$$B_{tmp_{-1}} = \sum_{i=0}^{l-1} \frac{B_{i,1} + B_{i,2}}{(\Delta t_{max} \cdot f_{CPHC}) + (\Delta t_{idx_update} \cdot f_{CPHC})},$$
(8)

где: Δt_{idx} undate – время с момента переключения столбца в буфере.

При обнаружении факта искажения используется $\hat{x}_{k|k} = \hat{x}_{k|k-1} - B_{imp_1}$ в течение $(\Delta t_{max} \cdot f_{CPHC}) + (\Delta t_{idx_update} \cdot f_{CPHC})$ тактов коррекции. Затем коррекция вектора пространства состояний по сигналам СРНС отключается до выхода из зоны искажения.

Опишем предложенный метод с использованием конечного автомата для событий (рис. 1):

– данные для коррекции достоверны;

- данные в буфере исчерпаны;

– данные для коррекции не достоверны.

В условиях наличия достоверных данных для шага коррекции, система находится в нормальном состоянии и выполняет буферизацию вектора невязки в режиме скользящего окна. При получении события «данные для коррекции недостоверны», система извлекает накопленные данные вектора невязки из буфера и выполняет их вычитание из вектора пространства состояний в соответствии с (8). Далее, система переходит в автономный режим навигации до получения события «данные для коррекции достоверны».



Применение метода в существующих цифровых системах требует наличия программной реализации, представленной в приложении 1. Для апробации полученного решения, выполнена его интеграция в существующую систему навигации на базе линейного фильтра Калмана [19]. Анализ времени выполнения программной реализации алгоритма определения параметров движения БЛА с учетом предложенного выше метода показал увеличение времени обработки данных на 4 мкс в контроллере «STM32H745ZI», работающем на частоте 480 МГц, что увеличило время выполнения алгоритма на 11,1%.

Пример

Покажем эффективность предложенного метода на примере БЛА «Альбатрос СКФУ», имитационная модель которого представлена в [20]. Будем считать, что известны следующие параметры:

- время начала искажения сигналов СРНС, которое составляет 120 с;
- величина искажения сигналов СРНС при помеховом воздействии, которая увеличивалась инкрементно с частотой 10 Гц на величину 0,000045° для долготы и 0,000027° для широты;
- время буферизации вектора невязки в режиме скользящего окна $\Delta t_{\text{max}} = 20 \text{ c};$
- время идентификации помехового воздействия $\triangle(t_2 t_1) = 3$ с.

Требуется определить: величину погрешности вектора местоположения $\triangle \vec{P}_{ouuo}^{Hab}$ после преодоления зоны помехового воздействия.

Ограничения: время идентификации факта помехового воздействия не превышает 20 с.

На рис. 2 приведены результаты моделирования без применения программной реализации предложенного метода. На 123 с, система отключает коррекцию до конца моделирования и выполняет оценку параметров движения только инерциальным методом счисления пути. Анализ результатов показал, что максимальная погрешность позиционирования достигает 59,094 м для северного направления и 6,142 м для восточного.

Скорость нарастания погрешности позиционирования максимальна на промежутке времени между фактическим началом помехового воздействия и идентификацией данного факта бортовым вычислителем. Компенсация погрешностей вертикального канала выполнялась с помощью барометрического высотомера, поэтому ухудшения точности определения высоты при помеховом воздействии на приемник СРНС не наблюдается.

На рис. 3 представлены результаты моделирования с применением программной реализации предложенного метода в контуре системы навигации. Параметры моделирования идентичны случаю на рис. 2.



Рис. 2. Результаты имитационного моделирования БЛА «Альбатрос СКФУ» без применения предложенного метода: а) погрешность оценки вектора скорости; б) погрешность оценки вектора местоположения



Рис. 3. Результаты имитационного моделирования БЛА «Альбатрос СКФУ» с применением предложенного метода: а) погрешность оценки вектора скорости; б) погрешность оценки вектора местоположения

Вычитание вектора невязки из вектора пространства состояний фильтра Калмана начинается на 123 с моделирования и продолжается до 146 с. После

завершения восстановления вектора пространства состояний, программная реализация метода переключилась в блокированный режим до конца проведения моделирования. Для данного случая, максимальная погрешность позиционирования составила 3,337 м и 0,869 м для северного и восточного направлений соответственно.

С целью сравнения полученного улучшения точности позиционирования в условиях помехового воздействия, результат моделирования без буферизации и с её применением для БЛА «Альбатрос СКФУ» представлены в виде диаграммы (рис. 4).



Рис. 4. Сравнение погрешностей позиционирования БЛА «Альбатрос СКФУ» после идентификации помехового воздействия

Повышение точности позиционирования с применением программной реализации предложенного метода в рассматриваемом сценарии составило $|\triangle x| = 55,757 \text{ м}, |\triangle y| = 5,273 \text{ м}.$

Моделирование показало инвариантность предложенного метода к следующим параметрам: величине помехового воздействия на приемник СРНС, времени его идентификации и переходной характеристики алгоритма коррекции.

На рис. 5 представлены результаты имитационного моделирования с применением программной реализации предложенного метода, в которых величина погрешности СРНС изменялась с 3-мя промежуточными точками в промежутке $\triangle P_{CPHC}^{ouud} \in [-0,005^\circ; +0,005^\circ]$. Помеховое воздействие начиналось на 50 с, идентификация факта искажения длилась 10 с. Анализ результатов показывает, что при помеховом воздействии, погрешность оценки местоположения не превышает 5 м на 90 с.



Рис. 5. Варьирование смещения координаты: а) погрешность оценки вектора скорости; б) погрешность оценки вектора местоположения

Покажем инвариантность метода к переходной характеристике фильтра. Используем граничные величины смещения $\triangle P_{CPHC}^{ouud}$, а именно $-0,005^{\circ}$ для долготы и $+0,005^{\circ}$ для широты. Переходная характеристика фильтра Калмана задается матрицей коэффициентов усиления *K*, которая зависит от соотношения матриц шумов *Q* и *R*. Поэтому, для варьирования переходной характеристики изменялось значение матрицы $Q \in [0,00001; 0,01]$ с шагом 0,1.

Временные интервалы моделирования аналогичны предыдущему случаю. Анализ рис. 6 показывает инвариантность предложенного метода к передаточной функции алгоритма.

Выполним варьирование временного интервала, затраченного на идентификацию помехового воздействия. В качестве смещения координаты, использовались величины, аналогичные случаю варьирования переходной характеристики. Задержка факта идентификации помехового воздействия изменялась от 12,5 до 35 с включительно.

На рис. 7 представлены случаи, в которых предложенный метод не справился с компенсацией смещения сигналов СРНС. Это связано с тем, что время идентификации превышает время буферизации, которое составляет 20 с. Для остальных случаев наблюдается инвариантность к времени определения факта искажения.

Абсолютное значение погрешности вектора местоположения для рассмотренных случаев не превышает 10 м на момент завершения восстановления вектора пространства состояний.





Рис. 6. Варьирование переходной характеристики алгоритма: а) погрешность оценки вектора скорости; б) погрешность оценки вектора местоположения



Рис. 7. Варьирование времени идентификации помехового воздействия: a) – погрешность оценки вектора скорости; б) – погрешность оценки вектора местоположения

Выводы

Представленный метод позволяет повысить точность позиционирования БЛА при преодолении зон помехового воздействия путем исключения недосто-

верных данных бортового модуля СРНС из контура системы навигации за промежуток времени между фактическим началом помехового воздействия и идентификацией данного факта бортовым вычислителем.

Элементом новизны метода является учет временного интервала между фактическим началом помехового воздействия на бортовой приемник СРНС и идентификацией данного факта бортовым вычислителем, а также выполнение буферизации вектора невязки в режиме скользящего окна для исключения недостоверных данных СРНС после идентификации факта помехового воздействия.

Описание последовательности действий при идентификации факта помехового воздействия использует конечный автомат состояний, позволяющий осуществить переход в автономный режим навигации с учетом промежутка времени, необходимого бортовому вычислителю для определения факта искажения (рис. 1).

Для апробации предложенного метода использована система навигации БЛА «Альбатрос СКФУ». Анализ полученных результатов показал уменьшение погрешности позиционирования на 55,757 м для северного и на 5,273 м для восточного направлений при переходе в автономный режим навигации и задержке идентификации факта помехового воздействия, которая составила 3 с (рис. 2-4). Показана инвариантность метода к величине и знаку помехового воздействия (рис. 5), переходной характеристики используемого алгоритма (рис. 6), а также времени, необходимого для идентификации факта помехового воздействия бортовым вычислителем, если задержка идентификации не превышает временного интервала буферизации вектора невязки (рис. 7).

Применение предлагаемого метода возможно в системах, которые выполняют комплексирование данных математической модели и измерителей с использованием вектора невязки, который является разностью между прогнозируемыми параметрами системы и измерениями, получаемыми от используемых датчиков. Т.к. вектор невязки является разностью, то его буферизация не накладывает ограничений на динамику изменения параметров вектора пространства состояний рассматриваемой системы.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 20-37-90010

Приложение 1

Программа, написанная на языке Си, реализующая метод восстановления коррекции (состоит из двух файлов: «Lib_A_IRBS.c» и «Lib_A_IRBS.h»)

```
Системы управления, связи и безопасности
                                                           Nº4. 2021
Systems of Control, Communication and Security
                                                        ISSN 2410-9916
/*#### |Begin| --> Секция - "Прототипы локальных функций" ########################
 IRBS ALWAYS INLINE void
prvReturnActualCorrection(
 void
                        *pvI,
 __LIB_A_MN_FPT
                  *aReturnInnov);
static void
prvPutNormal(
 void
                        *pvI,
 ___LIB_A_MN_FPT___
                  *aReturnInnov);
static void
prvGetNormal(
 void
                        *pvI,
 ___LIB_A_MN_FPT___
                  *aReturnInnov);
static void
prvGetStart(
 void
                        *pvI,
 LIB A MN FPT
                  *aReturnInnov);
static void
prvGetRun(
                        *pvI,
 void
 __LIB_A_MN_FPT
                  *aReturnInnov);
static void
prvGetStop(
 void
                        *pvI,
 __LIB_A_MN_FPT___
                  *aReturnInnov);
static void
prvGetReset(
 void
                        *pvI,
                  *aReturnInnov);
  LIB A MN FPT
/*#### |End | <-- Секция - "Прототипы локальных функций" ########################
/*#### |Begin| --> Секция - "Описание глобальных функций" #######################
void
IRBS StructInit(
 irbs init s *pInnovRollbackObj s)
{
 memset(
    (void*)pInnovRollbackObj s,
    0.
    sizeof(irbs init s));
}
/*-----*//**
 * @author Mickle Isaev
 * @date
          07-май-2021
 * @brief Инициализация управляющей структуры, которая содержит параметры,
             необходимые для работы программной реализации алгоритма вос-
становления
 *
              вектора невязки
 *
```

Системы управления, связи и безопасности Systems of Control, Communication and Security

```
ISSN 2410-9916
 * @param[out]
              *pI:
                            Указатель на управляющую структуру
 * @param[in]
             *pxInit: Указатель на структуру инициализации
 * @return Статус инициализации – PHAL INIT FNC STATUS SUCCESS
PHAL INIT FNC STATUS ERROR
*/
phal_init_fnc_status_e
IRBS Init(
 irbs_s
              *pI,
 irbs init s *pxInit)
{
 phal init fnc status e eInitStatus =
    PHAL_INIT_FNC_STATUS_SUCCESS;
 pI->aStateFnc[IRBS NORMAL] = prvGetNormal;
 pI->aStateFnc[IRBS_START] = prvGetStart;
 pI->aStateFnc[IRBS_RUN] = prvGetRun;
pI->aStateFnc[IRBS_STOP] = prvGetStop;
 pI->aStateFnc[IRBS RESET] = prvGetReset;
 /*-----*/
 if (pxInit->pMemMain == NULL)
 {
    eInitStatus = PHAL_INIT_FNC_STATUS ERROR;
 }
 if (pxInit->pMemBackup == NULL)
 {
    eInitStatus = PHAL INIT FNC STATUS ERROR;
 }
 if (pxInit->pMemTmp == NULL)
 {
    eInitStatus = PHAL_INIT_FNC_STATUS ERROR;
 }
 if (pxInit->ulVectLength == 0u)
 {
    eInitStatus = PHAL INIT FNC STATUS ERROR;
 /*-----*/
 pI->ulVectLength
                  = pxInit->ulVectLength;
 pI->ulBackupCntMaxVal = pxInit->ulBackupCntMaxVal;
 /*-----*/
 pI->pMemMain = pxInit->pMemMain;
 pI->pMemBackup = pxInit->pMemBackup;
 pI->pMemTmp = pxInit->pMemTmp;
 /*_____*/
 IRBS SetModeNormal(pI);
 return (eInitStatus);
}
/*-----*//**
 * @author Mickle Isaev
 * @date
         07-май-2021
 * @brief
         Функция отправляет новое значение вектора невязки в буфер
```

Nº4. 2021

Системы управления, связи и безопасности Systems of Control, Communication and Security

```
Nº4. 2021
ISSN 2410-9916
```

```
* @param[in,out] *pI:
                                  Указатель на управляющую структуру
 * @param[in]
             *pNewInnov: Указатель на массив, который содержит вектор
                                       невязки для буферизации
 * @return None
 */
void
IRBS PutNewCorrection(
 irbs s
                            *pI,
  _LIB_A_MN FPT
                      *pNewInnov)
 irbs innovation status e eStatus = pI->eStatus;
 if (eStatus == IRBS NORMAL)
 {
     prvPutNormal(
           (void*) pI,
           pNewInnov);
 }
}
/*-----*//**
 * @author
            Mickle Isaev
 * @date
           07-май-2021
 *
 * @brief Функция возвращает актуальное значение вектора невязки из буфера.
                 Данная функция должна быть вызвана сразу после
IRBS PutNewCorrection()
 * @param[in,out] pI: Указатель на управляющую структуру
 * @param[out] aReturnCorrection: Актуальное значение вектора невязки,
                           которое должно быть применено в пользовательском
коде
 *
 * @return Текущее состояние конечного автомата (см.
<irbs innovation status e>)
*/
irbs innovation_status_e
IRBS GetActualCorrection(
 irbs s
                            *pI,
  _LIB_A_MN_FPT_
                     *aReturnCorrection)
{
 irbs innovation status e eStatus = pI->eStatus;
 /* Если указатель на функцию равен NULL, то нечего вызывать.
  * (для состояния IRBS BLOCK указатель на функцию равен NULL и,
  * соответственно, не вызывается функция-обработчик) */
 if (pI->aStateFnc[eStatus] != NULL)
 {
      InterfaceAssert(pI->aStateFnc[eStatus]);
     /* Вызов функции, которая соответствует текущему состоянию системы отката
*/
     pI->aStateFnc[eStatus](
           (void*) pI,
           aReturnCorrection);
 /* if (pI->aStateFnc[eStatus] != NULL) */
 return (pI->eStatus);
ł
/*#### |End | <-- Секция – "Описание глобальных функций" ########################
```

```
<u>№4. 2021</u>
ISSN 2410-9916
```

```
Systems of Control, Communication and Security
/*#### |Begin| --> Секция - "Описание локальных функций" ########################
 IRBS ALWAYS INLINE void
prvReturnActualCorrection(
 void
                              *pvI,
  LIB A MN FPT
                        *aReturnInnov)
 irbs_s *pI = (irbs_s*) pvI;
 /* Значение, которое необходимо вернуть в качестве корректирующей величины
  * всегда находится по адресу <pMemTmp> */
 size t i;
 for (i = 0u; i < pI->ulVectLength; i++)
 {
      aReturnInnov[i] = pI->pMemTmp[i];
 }
}
static void
prvPutNormal(
 void
                              *pvI,
  _LIB_A_MN_FPT___
                        *aReturnInnov)
 irbs s *pI = (irbs s*) pvI;
 size t i;
 for (i = 0u; i < pI->ulVectLength; i++)
 {
      /* Прибавить новый вектор в существующему */
      pI->pMemMain[i] += aReturnInnov[i] * (( LIB A MN FPT ) 1000.0);
      /* Копировать во временный вектор (для вызова функции
       * IRBS InnovationRollbackObj_Get()) */
      pI->pMemTmp[i] = aReturnInnov[i];
 }
 /* Инкремент счетчика с проверкой переполнения */
 pI->ulActualCnt =
      (pI->ulActualCnt + 1u) % pI->ulBackupCntMaxVal;
 /* Если было переполнение */
 if (pI->ulActualCnt == 0u)
 {
      for (i = 0u; i < pI->ulVectLength; i++)
      {
            /* Копировать в вектор для хранения */
            pI->pMemBackup[i] = pI->pMemMain[i];
            /* Очистить основной вектор */
            pI->pMemMain[i] = (__LIB_A_MN_FPT__) 0.0;
      }
      /* for (i = 0u; i < pI->innovVectLength; i++) */
 }
 /* if (pI->actualCnt == 0u) */
}
static void
prvGetNormal(
 void
                              *pvI,
  LIB A MN FPT
                       *aReturnInnov)
{
 prvReturnActualCorrection(pvI, aReturnInnov);
}
```

Системы управления, связи и безопасности

```
static void
prvGetStart(
 void
                              *pvI,
   LIB A MN FPT
                       *aReturnInnov)
 irbs_s *pI = (irbs_s*) pvI;
 pI->ulActualCnt += pI->ulBackupCntMaxVal / 2u;
 /* Вычислить Величину отката за шаг времени */
 size t i;
 for (i = 0u; i < pI->ulVectLength; i++)
 {
      /* Вычислить величину коррекции для одного шага из основного вектора не-
вязки */
     pI->pMemTmp[i] =
            -((pI->pMemMain[i]
               + pI->pMemBackup[i])
              / ((__LIB_A_MN_FPT_) pI->ulActualCnt));
     pI->pMemTmp[i] *= ((__LIB_A_MN_FPT__) 0.001);
 }
 pI->eStatus = IRBS RUN;
 prvReturnActualCorrection(pvI, aReturnInnov);
}
static void
prvGetRun(
 void
                              *pvI,
  LIB_A_MN_FPT___
                      *aReturnInnov)
 irbs s *pI = (irbs s*) pvI;
 pI->ulActualCnt--;
 if (pI->ulActualCnt < 1u)
 {
     pI->eStatus = IRBS STOP;
 }
 prvReturnActualCorrection(pvI, aReturnInnov);
}
static void
prvGetStop(
 void
                              *pvI,
 __LIB_A_MN_FPT___ *aReturnInnov)
{
 irbs s *pI = (irbs s*) pvI;
 size t i;
 for (i = 0u; i < pI->ulVectLength; i++)
 {
      /* Очистка векторов */
                             = ( LIB A MN FPT ) 0.0;
     pI->pMemBackup[i]
     pI->pMemMain[i] = (__LIB_A_MN_FPT__) 0.0;
                             = ( LIB A MN FPT ) 0.0;
     pI->pMemTmp[i]
 }
 pI->eStatus = IRBS BLOCK;
 prvReturnActualCorrection(pvI, aReturnInnov);
```

}

```
static void
prvGetReset(
void
              *pvI,
 LIB A MN FPT *aReturnInnov)
irbs s *pI = (irbs s*) pvI;
size t i;
for (i = 0u; i < pI->ulVectLength; i++)
   /* Очистка памяти буфера */
  pI->pMemBackup[i] = ( LIB A MN FPT ) 0.0;
  pI->pMemMain[i] = (__LIB_A_MN_FPT__) 0.0;
  pI->pMemTmp[i]
              = ( LIB A MN FPT ) 0.0;
}
pI->eStatus = IRBS NORMAL;
prvReturnActualCorrection(pvI, aReturnInnov);
}
/*#### |End | <-- Секция – "Описание локальных функций" #########################
Файл Lib A IRBS.h
#ifndef LIB A IRBS INNOVATION ROLLBACK SYSTEM H
#define LIB A IRBS INNOVATION ROLLBACK SYSTEM H
#include <stdint.h>
#include <stddef.h>
#include <string.h>
/*==== |End | <-- Секция - "RTOS libraries =================*/
/*==== |Begin| --> Секция - "MK peripheral libraries" ====================*/
#include "Lib A MN standart definitions.h"
#include "PHAL definitions init fnc status.h"
#include "Lib A interfaces.h"
/*==== |End | <-- Секция - "Extern libraries" ================================*/
typedef enum
PHAL INIT FNC STATUS ERROR
                 = 0u,
PHAL INIT FNC STATUS SUCCESS
                 = 1u,
} phal init fnc status e;
```

Системы управления, связи и безопасности Systems of Control, Communication and Security

#if defined (GNUC) /* inline*/ #ifndef IRBS INLINE #define __IRBS_INLINE inline #endif /* static inline */ #ifndef __IRBS_STATIC INLINE #define IRBS STATIC INLINE static inline #endif /* always inline */ #ifndef IRBS ALWAYS INLINE #define __IRBS_ALWAYS_INLINE inline __attribute__((always_inline)) static #endif /* force inline */ #ifndef IRBS FORCE INLINE #define __IRBS_FORCE_INLINE inline __attribute__((always_inline)) #endif #else #define __IRBS_INLINE #define __IRBS_STATIC_INLINE
#define __IRBS_ALWAYS_INLINE
#define __IRBS_FORCE_INLINE static static #endif typedef enum IRBS NORMAL = 0u, IRBS START, IRBS RUN, IRBS STOP, /* Данное состояние необходимо для принудительного сброса буфера при * переключении в режим NORMAL */ IRBS RESET, /*------*//** * Режим блокировки * @note В данном режиме система не записывает новые данные. * Выдает нулевое значение вектора инновации */ IRBS BLOCK, /*-----*//** * Количество состояний системы отката (должен быть расположен в конце * перечисляемого типа) */ IRBS STATE MAX NUMB, } irbs innovation status e; typedef struct {

```
__LIB_A_MN_FPT__ *pMemMain;
__LIB_A_MN_FPT__ *pMemBackup;
__LIB_A_MN_FPT__ *pMemTmp;
 uint32 t ulBackupCntMaxVal;
            ulVectLength;
 size t
 uint32 t ulActualCnt;
 irbs innovation status e eStatus;
 /* Массив указателей на функции-обработчики каждого состояния системы отката
*/
 void (*aStateFnc[IRBS STATE MAX NUMB])(
                                *pvI,
      void
       _LIB_A_MN_FPT__ *aNewInnov)
} irbs s;
typedef struct
{
 __LIB_A_MN_FPT__ *pMemMain;
__LIB_A_MN_FPT__ *pMemBackup;
__LIB_A_MN_FPT__ *pMemTmp;
 /* Длина вектора невязки */
 size t
                  ulVectLength;
 /* Количество тиков, в течение которых выполняется буферизация данных в
  * режиме скользящего окна */
 uint32 t ulBackupCntMaxVal;
} irbs init s;
/**
 * @brief Функция устанавливает "Нормальный" режим работы алгоритма буфериза-
ЦИИ
 *
                   вектора невязки
 *
 * @param[in]
                  pxI: Указатель на управляющую структуру алгоритма
 * @return None
 */
  IRBS ALWAYS INLINE void
IRBS SetModeNormal(
 irbs s *pxI)
{
 if (pxI->eStatus != IRBS NORMAL)
 {
      /* Система выполнит сброс значений в буфере из состояния <RESET> и автома-
тически
       * переключится в режим <NORMAL> */
      pxI->eStatus = IRBS_RESET;
 }
}
/**
 * @brief Функция устанавливает режим восстановления коррекции для алгоритма
                   буферизации вектора невязки
 * @param[in]
                  pxI: Указатель на управляющую структуру алгоритма
 * @return None
 */
  IRBS ALWAYS INLINE void
IRBS SetModeRollback(
 irbs s *pxI)
```

```
{
/* Активация режима отката возможна только сиз состояния <Normal>, в противном
 * случае игнорировать переключение состояния */
if (pxI->eStatus == IRBS NORMAL)
{
  pxI->eStatus = IRBS START;
}
}
/*#### |Begin| --> Секция - "Определение глобальных переменных" ################/
/*#### |End | <-- Секция - "Определение глобальных переменных" ################/
/*#### |Begin| --> Секция - "Прототипы глобальных функций" ######################
extern void
IRBS StructInit(
irbs init s *pInnovRollbackObj s);
extern phal init fnc status e
IRBS Init(
irbs s
          *pI,
irbs init s *pInnovRollbackObjInit s);
extern void
IRBS PutNewCorrection(
irbs s
                *pI,
____LIB_A_MN_FPT___ *pNewInnov);
extern irbs innovation status e
IRBS GetActualCorrection(
irbs s
                 *pI,
 LIB A MN FPT *aReturnCorrection);
/*#### |End | <-- Секция - "Прототипы глобальных функций" ######################/
/* LIB A IRBS INNOVATION ROLLBACK SYSTEM H */
#endif
```

Литература

1. Селиванова Л. М., Шевцова Е. В. Одноканальные инерциальные навигационные системы: Учебное пособие. – М.: МГТУ им. Н.Э. Баумана, 2012. – 49 с.

2. Бранец В. Н., Шмыглевский И. П. Применение кватернионов в задачах ориентации твердого тела. М.: Наука, 1973. – 319 с.

3. Макаренко С. И. Анализ средств и способов противодействия беспилотным летательным аппаратам. Часть 3. Радиоэлектронное подавление

систем навигации и радиосвязи // Системы управления, связи и безопасности. 2020. № 2. С. 101-175. DOI: 10.24411/2410-9916-2020-10205.

4. Ташков С. А., Булочников Д. Ю., Шатовкин Р. Р. Исследование существующих способов автономного контроля целостности навигационных данных // NovaInfo.Ru. 2018. Т 1. № 91. С. 41-46.

5. Хусаинов Н. Ш. Разработка алгоритма автономного контроля целостности для бортовой части системы радионавигации // Известия Южного федерального университета. Технические науки. 2007. №. 1. С. 188-194.

6. Марарескул Д. И., Ермолаев М. В., Малышев А. А., Полякова А. А. Контроль навигационного поля системы ГЛОНАСС // Системы управления, космическая навигация и связь. 2017. №. 1. С. 277-279.

7. Иванов А. В., Негуляева А. П., Москвитин С. П. Автономный контроль целостности навигационных данных спутниковых радионавигационных систем способами сравнения и невязок // Вестник Тамбовского государственного технического университета. 2016. Т. 22. № 3. С. 358-367.

8. Сагдеев К. М., Линец Г. И., Мельников С. В., Исаев А. М., Исаев М. А. Способ определения факта искажения навигационного поля и идентификации помехового воздействия на приемник РБЛА // Инфокоммуникационные технологии. 2020. Т. 18. № 2. С. 166-177.

9. Сурков В. О. Анализ методов контроля целостности спутниковых радионавигационных систем в навигационных системах подвижных наземных объектов // Молодой учёный. 2016. Т. 13. №. 2. С. 219-221

10. Бойков Д. В. Алгоритм автономного контроля целостности навигационного поля // Технические науки в России и за рубежом. 2014. № 3. С. 27-32.

11. Kolosov K., Miller A., Miller B. Robust Data Fusion of UAV Navigation Measurement with Application to the Landing System // Remote Sensing. 2020. DOI: 10.3390/rs12233849.

12. Chunshi F., Zheng Y. Highly Efficient Sigma Point Filter for Spacecraft Attitude and Rate Estimation // Mathematical Problems in Engineering. 2009. DOI: 10.1155/2009/507370

13. Shin E.H. Estimation Techniques for Low-Cost Inertial Navigation. Department of Geomatics Engineering. DOI: 10.11575/PRISM/2386

14. Lin S., Garratt M. A., Lambert A. J. Monocular vision-based real-time target recognition and tracking for autonomously landing an UAV in a cluttered shipboard environment // Autonomous Robots. No. 41. 2017. P. 881-901. DOI: 10.1007/s10514-016-9564-2.

15. Feng K., Li J., Zhang X., Shen C., Bi Y., Zheng T., Liu J. A New Quaternion-Based Kalman Filter for Real-Time Attitude Estimation Using the Two-Step Geometrically-Intuitive Correction Algorithm // Sensors. 2017. Vol. 17. № 9. DOI: 10.3390/s17092146.

16. Jian Li, Xinguo W., Guangjun Z. An Extended Kalman Filter-Based Attitude Tracking Algorithm for Star Sensors // Sensors. 2017. Vol. 17. № 8. DOI: 10.3390/s17081921.

17. Manish S, Sanjeeb P. P., Basanta J., Aman S., Rom K. P. Automatic Pose Estimation of Micro Unmanned Aerial Vehicle for Autonomous Landing // Intelligent Computing Methodologies. 2020. Vol. 16. № 3. P. 3-15. DOI: 10.1007/978-3-030-60796-8_1.

18. Raitoharju M., Nurminen H., Piché R. Kalman filter with a linear state model for PDR+WLAN positioning and its application to assisting a particle filter // EURASIP Journal on Advances in Signal Processing. 2015. DOI: 10.1186/s13634-015-0216-z.

19. Isaev M. A., Linets G. I., Nikulin V. I., Melnikov S. V., Isaev A. M. Increasing fast estimation of assessment of navigation parameters for the UAV based on a linear Kalman filter // AIP Conference Proceedings 2318. 2021. P. 1-8. DOI: 10.1063/5.0035917.

20. Исаев А. М., Линец Г. И., Мельников С. В., Исаев М. А. Программноаппаратный комплекс имитационного моделирования полета мультироторного БЛА // Инфокоммуникационные технологии. 2020. Т. 18. № 2. С. 177-187.

References

1. Selivanova L. M., Shevcova E. V. Odnokanal'nye inercial'nye navigacionnye sistemy [Single channel inertial navigation systems]. Moscow, Bauman Moscow State Technical University Publ., 2012. 49 p. (in Russian).

2. Branets V. N., Shmyglevsky I. P. Primenenie kvaternionov v zadachakh orientatsii tverdogo tela. [Application of quaternions in solid orientation problems]. Moscow Nauka Publ., 1973. 320 p. (in Russian).

3. Makarenko S. I. Counter Unmanned Aerial Vehicles. Part 3. Electronic Warfare against Navigation and Radio Connection Subsystems of Unmanned Aerial Vehicles. *Systems of Control, Communication and Security*, 2020, no. 2, pp. 101-175 (in Russian). DOI: 10.24411/2410-9916-2020-10205.

4. Tashkov S. A., Bylochnikov D. U., Shatovkin R. R. Issledovanie sushchestvuyushchih sposobov avtonomnogo kontrolya celostnosti navigacionnyh dannyh [Investigation of methods for autonomous control of the integrity of navigation data.]. *NovaInfo.Ru*, 2018, vol. 1, no 91, pp. 41-46 (in Russian).

5. Xysainov N. S. Razrabotka algoritma avtonomnogo kontrolya celostnosti dlya bortovoj chasti sistemy radionavigacii [Development of an algorithm for autonomous integrity control for the onboard part of the radio navigation system]. *Izvestiya SFedU. Engineering sciences*, 2007, vol. 1, pp. 188-194 (in Russian).

6. Marareskul D. I. Kontrol' navigacionnogo polya sistemy GLONASS [Control of the navigation field of the GLONASS system]. *Materialy XXI MNPK, posvyashchennoj pamyati general'nogo konstruktora raketno-kosmicheskih sistem akademika. M.F. Reshetneva* [Materials of the XXI MNPK dedicated to the memory of the general designer of rocket and space systems, Acad. M.F. Reshetneva], 2007, vol. 1, pp. 277-279 (in Russian).

7. Ivanov A. V., Negulaeva A. P., Moscwitin C. P. Avtonomnyj kontrol' celostnosti navigacionnyh dannyh sputnikovyh radionavigacionnyh sistem sposobami sravneniya i nevyazok [Autonomous control of the integrity of navigation data of

satellite radio navigation systems by means of comparison and innovation] *Vestnik TGTU*, 2016, vol. 22, pp. 358-367 (in Russian).

8. Sagdeev K. M., Linec G. I., Melnikov S. V., Isaev A. M., Isaev M. A. Method for determining the navigation field distortion and the dentification of interference on the RUAV receiver. *Infokommunikacionnye tekhnologii*, 2020, vol. 18, no. 2, pp. 166-177 (in Russian).

9. Surkov V. O. Analiz metodov kontrolya celostnosti sputnikovyh radionavigacinnyh sistem v navigacionnyh sistemah podvizhnyh nazemnyh ob"ektov [Analysis of methods for monitoring the integrity of satellite radio navigation systems in navigation systems of mobile ground objects]. *Molodoj uchyonyj* [Young scientist], 2016, vol. 13, no. 2, pp. 219-221 (in Russian).

10. Boikov D. V. Algoritm avtonomnogo kontrolya celostnosti navigacionnogo polya [Algorithm for autonomous control of the integrity of the navigation field]. *Tekhnicheskie nauki v Rossii i za rubezhom: materialy 3 mezhdunarodnoj nauchnoj konferencii* [Engineering Sciences in Russia and Abroad: Proceedings of the 3rd International Scientific Conference], 2014, pp. 27-32 (in Russian).

11. Kolosov K., Miller A., Miller B. Robust Data Fusion of UAV Navigation Measurement with Application to the Landing System. *Remote Sensing*, 2020. DOI: 10.3390/rs12233849.

12. Chunshi F., Zheng Y. Highly Efficient Sigma Point Filter for Spacecraft Attitude and Rate Estimation. *Mathematical Problems in Engineering*, 2009. DOI: 10.1155/2009/507370

13. Shin E. H. Estimation Techniques for Low-Cost Inertial Navigation. Department of Geomatics Engineering. DOI: 10.11575/PRISM/2386

14. Lin. S., Garratt M. A, Lambert A. J. Monocular vision-based real-time target recognition and tracking for autonomously landing an UAV in a cluttered shipboard environment. *Autonomous Robots*, 2017, no. 41, pp. 881-901. DOI: 10.1007/s10514-016-9564-2.

15. Feng K, Li J, Zhang X, Shen C, Bi Y, Zheng T, Liu J. A New Quaternion-Based Kalman Filter for Real-Time Attitude Estimation Using the Two-Step Geometrically-Intuitive Correction Algorithm. *Sensors*, 2017, vol. 17, no. 9. DOI: 10.3390/s17092146.

16. Jian Li, Xinguo W., Guangjun Z. An Extended Kalman Filter-Based Attitude Tracking Algorithm for Star Sensors. *Sensors*, 2017, vol. 17, no. 8. DOI: 10.3390/s17081921.

17. Manish S, Sanjeeb P. P., Basanta J., Aman S., Rom K. P. Automatic Pose Estimation of Micro Unmanned Aerial Vehicle for Autonomous Landing. *Intelligent Computing Methodologies*, 2020, vol. 16, no. 3, pp. 3-15. DOI: 10.1007/978-3-030-60796-8_1.

18. Raitoharju M., Nurminen H., Piché R. Kalman filter with a linear state model for PDR+WLAN positioning and its application to assisting a particle filter. *EURASIP Journal on Advances in Signal Processing*, 2015. DOI: 10.1186/s13634-015-0216-z

19. Isaev M. A., Linets G. I., Nikulin V. I., Melnikov S. V., Isaev A. M. Increasing fast estimation of assessment of navigation parameters for the UAV based

on a linear Kalman filter. *AIP Conference Proceedings 2318*, 2021, pp. 1-8. DOI: 10.1063/5.0035917 (in Russian).

20. Isaev A. M., Linets G. I., Isaev M. A., Melnikov S. V. Software and hardware complex for simulation Modeling of a multirotor UAV flight. *Infokommunikacionnye tekhnologii*, 2020, vol. 18, no. 2, pp. 177–187 (in Russian).

Статья поступила 16 июня 2021 г.

Информация об авторах

Исаев Михаил Александрович – соискатель ученой степени кандидата технических наук, Аспирант кафедры инфокоммуникаций. Северо-Кавказский федеральный университет. Область научных интересов: системы навигации малых беспилотных летательных аппаратов, системы управления малых беспилотных летательных аппаратов, имитационное моделирование. E-mail: mrraptor26@gmail.com

Линец Геннадий Иванович – доктор технических наук, доцент. Заведующий кафедрой «Инфокоммуникаций». Северо-Кавказский федеральный университет. Область научных интересов: синтез мультисервисных сетей; исследование влияния свойств самоподобного трафика на количественные и качественные показатели современных сетей; повышение эффективности функционирования беспилотных летательных аппаратов мультироторного типа. Е-mail: kbytw@mail.ru

Адрес: 355017, Россия, г. Ставрополь, ул. Пушкина, д. 1.

Method for recovering estimation of the motion parameters of an unmanned aerial vehicle when identifying the fact of interference on the on-board receiver of satellite navigation signals

M. A. Isaev, G. I. Linets

Relevance. Navigation tasks on the overwhelming number of small unmanned aerial vehicles (UAVs) are solved by combining signals from a satellite radio navigation system (SRNS) and an inertial navigation system. An effective method for countering UAVs is the creation of zones of local suppression of signals from navigation satellites, which distorts the assessment of UAV motion parameters performed by the on-board computer in real time, and as a result, deviates the actual flight path from the specified one. Known works describing the algorithms for the operation of UAVs in a jamming environment do not consider the time interval between the actual onset of jamming on the SRNS receiver and the identification of this fact by an onboard computer, which does not allow ensuring the accuracy of maintaining a given flight path required for UAV operation when overcoming jamming zones. The aim of the work is to increase the accuracy of the estimation of motion parameters when overcoming the zones of interference by eliminating inaccurate data of the on-board module of the SRNS from the navigation system contour for the time interval between the actual start of the interference and the identification of this fact by the on-board computer. Methods. The problem was solved by buffering the residual vector, which is the difference between the estimated state of the UAV motion parameters using a mathematical model and the parameters obtained from the onboard module of the SRNS. When performing a flight task under conditions of reliable signals from navigation satellites, the proposed method buffers the residual vector in a sliding window mode. After identifying the fact of interference, the data of the SRNS module are not used, and the buffered value of the residual vector is subtracted from the estimate of the state space vector, thereby eliminating unreliable SRNS data and restoring the accuracy of the estimation of the UAV motion parameters. In the future, before leaving the zone of interference, the inertial dead reckoning method is used to update the parameters of the UAV movement. Novelty. Element of novelty in this work is the procedure for buffering the residual vector to exclude inaccurate data of the on-board SRNS module after identifying the fact of interference for the time period necessary for the on-board computer to identify the fact of interference. **Result.** The use of the proposed method in the navigation system of the UAV "Albatros NCFU" made it possible to ensure the accuracy of positioning of the UAV when overcoming interference zones no worse than 3.337 m for the northern and 0.869 m for the eastern directions after identifying the fact of interference, which reduced the error by 55.757 m and 5.273 m respectively. **Practical relevance.** The solution of the problem, taking into account the specified features, made it possible to reduce the positioning error of small UAVs and ensure its invariance for the time interval required by the onboard computer to determine the fact of interference. The presented method is implemented in the C programming language in accordance with the C11 standard and is suitable for use in embedded systems that are characterized by low computing resources, RAM and ROM.

Keywords: unmanned aerial vehicle, UAV, UAV, inertial navigation system, interference effect, GNSS receiver, Kalman filter.

Information about Authors

Mikhail Alexandrovich Isaev – Doctoral Student. The postgraduate student of the North Caucasus Federal University of Infocommunications Department. Field of research: UAV navigation system, UAV control system, hardware in the loop model-ing. E-mail: mrraptor26@gmail.com

Gennadiy Ivanovich Linets – Advanced Doctor of Engineering Sciences, Associate Professor. Head of Infocommunications Department. Field of research: synthesis of multiservice networks; study of the influence of the properties of self-similar traffic on the quantitative and qualitative indicators of modern networks; improving the efficiency of multi-rotor unmanned aerial vehicles. E-mail: kbytw@mail.ru

Address: Russia, 355017, Stavropol, Pushkin Street, 1.