



УДК 621.396.96

А. В. Бархатов, А. С. Козлов

Санкт-Петербургский государственный электротехнический университет "ЛЭТИ" им. В. И. Ульянова (Ленина)

Быстрое вычисление частотно-временной функции в радиолокационной станции на графических процессорах¹

Описан алгоритм быстрого вычисления частотно-временной функции в радиолокационной станции (РЛС). Сделан акцент на особенностях программной реализации алгоритма. Описаны особенности параллельных вычислений функции на процессорах игровых видеокарт. Определено время, затрачиваемое на параллельные вычисления 16 частотно-временных функций в многоканальной РЛС.

Радиолокация, цифровая обработка, функция неопределенности, корреляционная функция, доплеровское смещение частоты, задержка, быстрое преобразование Фурье, графический процессор, параллельные вычисления

В подавляющем большинстве когерентных радиолокационных станций (РЛС) оптимальная обработка цифровых сигналов с неизвестными амплитудой, частотой, задержкой и начальной фазой предусматривает вычисление частотно-временной функции

$$|\Psi(l, d)| = \left| \sum_{n=0}^{N-1} s(n) s_{\text{ref}}^*(n-l) e^{-j2\pi dn/N} \right|, \quad (1)$$

где l – дискретизированная задержка (lag); d – дискретизированный частотный сдвиг; N – количество отсчетов принятого сигнала в обрабатываемом блоке; $n = \overline{0, N-1}$ – номер отсчета сигнала; $s(n)$ – принятый сигнал; $s_{\text{ref}}(n-l)$ – цифровой опорный (referenced) сигнал; "*" – символ комплексного сопряжения.

Дискретизированная задержка l представляет собой задержку сигнала τ , дискретизированную на интервал дискретизации (sampling) T_s : $l = \text{round}(\tau/T_s)$ ($\text{round}(\cdot)$ – операция взятия целой части аргумента). Дискретизированный частотный сдвиг аналогично связан с доплеровским смещением f_D и длительностью анализируемой выборки сигнала NT_s :

$$d = \text{round}(f_D NT_s) = \text{round}[(f_D/f_s)N],$$

где $f_s = 1/T_s$ – частота дискретизации.

Если $s(n)$ совпадает с опорным сигналом $s_{\text{ref}}(n-l)$, то (1) является оценкой функции неопределенности (ФН) сигнала $s_{\text{ref}}(n-l)$ на интервале задержек $0 \dots LT_s$ и частот $-Df_s/N \dots Df_s/N$. На практике в принятом сигнале обычно содержится множество копий $s_{\text{ref}}(n-l)$, сдвинутых по задержке и по частоте.

В зарубежной литературе функцию (1) называют "поверхностью амплитуда-дальность-Доплер" (amplitude-range-Doppler surface) или просто "поверхностью дальность-Доплер" из-за прямой связи задержки с дальностью и обусловленностью частотного сдвига эффектом Доплера. В отечественных источниках ее называют корреляционно-фильтровой функцией, взаимной функцией неопределенности, двумерной взаимно корреляционной функцией, функцией отклика и т. д. Далее будем называть (1) частотно-временной функцией (ЧВФ). На рис. 1 приведен пример ЧВФ, нормированной (normalized) на свое максимальное значение, выраженной в децибелах.

Алгоритм расчета частотно-временной функции. Различные способы расчета ЧВФ, учитыва-

¹ В публикации использованы результаты ОКР "Разработка пассивного когерентного локационного комплекса для охраны важных объектов", выполняемой СПбГЭТУ "ЛЭТИ" по договору с АО «НИИ "Вектор"» в рамках комплексного проекта по созданию высокотехнологического производства при финансовой поддержке работ по проекту Министерством образования и науки Российской Федерации (постановление Правительства Российской Федерации от 9 апреля 2010 г. № 218).

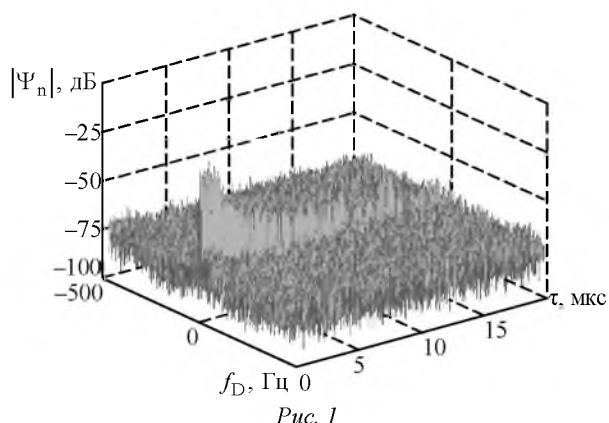


Рис. 1

ющие связь круговой свертки, линейной свертки, дискретного преобразования Фурье (ДПФ), а также свойства ДПФ и быстрого преобразования Фурье (БПФ), описаны в различных источниках. В настоящей статье сосредоточимся на деталях, важных при практической реализации вычисления ЧВФ. Обсуждаемый далее способ и вытекающий из него алгоритм при выполнении определенных условий, указанных ниже, обеспечивают наиболее быстрый расчет ЧВФ [1].

Пусть принятый сигнал s содержит N отсчетов, а опорный сигнал s_{ref} – $N + L$ отсчетов. На рис. 2, а блоки отсчетов условно показаны прямоугольниками.

Блок из N отсчетов сигнала s разбивается на M сегментов по N' отсчетов в каждом. Аналогично разбиваются N отсчетов сигнала s_{ref} и комплексной экспоненты. При этом можно записать:

$$|\Psi(l, d)| = \left| \sum_{m=0}^{M-1} \sum_{n'=0}^{N'-1} s(mN' + n') \times s_{\text{ref}}^*(mN' + n' - l) e^{-j2\pi d(mN' + n')/(MN')} \right|, \quad (2)$$

где $m = 0, \dots, M - 1$ – номер сегмента; n' – номер отсчета внутри сегмента.

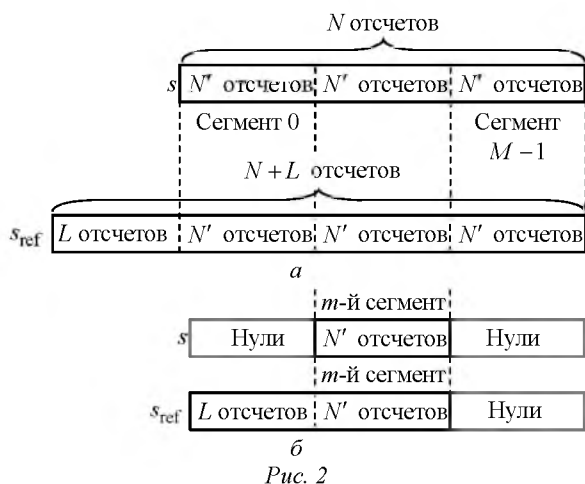


Рис. 2

С учетом того, что

$$\begin{aligned} e^{-j2\pi d(mN' + n')/(MN')} &= \\ &= e^{-j2\pi dmN'/(MN')} e^{-j2\pi dn'/(MN')} = \\ &= e^{-j2\pi dm/M} e^{-j2\pi dn'/(MN')}, \end{aligned}$$

получим

$$|\Psi(l, d)| = \left| \sum_{m=0}^{M-1} \left[e^{-j2\pi dm/M} \sum_{n'=0}^{N'-1} s(mN' + n') \times s_{\text{ref}}^*(mN' + n' - l) e^{-j2\pi dn'/(MN')} \right] \right|.$$

Рассмотрим случай $D \ll M$, что означает, что максимальное значение модуля частотного сдвига много меньше частоты повторения сегментов f_M :

$$\left(f_{D_{\text{max}}} = \frac{D}{N} f_s \right) \ll \left(f_M = \frac{f_s}{N'} = \frac{M}{N} f_s \right).$$

При этом максимальное значение модуля показателя экспоненты в (2)

$$\left[2\pi dn'/(MN') \right]_{\text{max}} = 2\pi D(N' - 1)/(MN') \approx 2\pi D/M \approx 0.$$

Тогда $e^{-j[2\pi dn'/(MN')]}_{\text{max}} \approx e^{-j0} = 1$. При меньших значениях d и n' это равенство тем более выполняется.

В этом случае ЧВФ

$$|\Psi(l, d)| = \left| \sum_{m=0}^{M-1} \left[e^{-j2\pi dm/M} \sum_{n'=0}^{N'-1} s(mN' + n') \times s_{\text{ref}}^*(mN' + n' - l) \right] \right|. \quad (3)$$

Обозначим

$$r_m(l) = \sum_{n'=0}^{N'-1} s(mN' + n') s_{\text{ref}}^*(mN' + n' - l) \quad (4)$$

– дискретную линейную свертку m -го сегмента s и m -го сегмента s_{ref}^* (рис. 2, б), вычисленную в ограниченном диапазоне сдвигов $0 \leq l \leq L$.

Спектр (результат ДПФ) круговой (циклической) свертки двух периодических последовательностей равен произведению спектров этих последовательностей [2]. Если выровнять длины периодов последовательностей, дополнив более короткий период нулями, то круговая свертка периодических последовательностей будет равна линейной свертке последовательностей конечной длины. Таким образом, линейную свертку двух конечных последовательностей можно вычислить с помощью ДПФ, предварительно дополнив хотя бы одну из последовательностей нулевыми отсчетами. В рассматриваемом случае одна конечная

последовательность образована отчетами в m -м сегменте s , вторая – отсчетами в m -м сегменте s_{ref}^* . Сокращение вычислительных затрат достигается за счет вычисления ДПФ с помощью алгоритма БПФ. Другими словами, для быстрого вычисления (4) нужно хотя бы один из сегментов (или принятого, или опорного сигналов) дополнить нулевыми отсчетами, выполнить БПФ для каждого из сегментов, перемножить полученные спектры и выполнить обратное БПФ. При программной реализации массивы с отсчетами сигналов должны иметь одинаковую длину, поэтому нулевыми отсчетами дополняются оба сегмента. При этом, во-первых, сегмент принятого сигнала s сначала дополняется L нулевыми отсчетами слева (см. рис. 2, б), во-вторых, количество нулевых отсчетов, которыми дополняются оба сегмента справа, выбирается таким, чтобы длина дополненных массивов не только равнялась или превышала $L + N' + N' - 1$ (чтобы круговая свертка совпадала с линейной), но и была степенью двойки – при этом обеспечивается наилучшая вычислительная эффективность БПФ.

С учетом (4) перепишем (3):

$$|\Psi(l, d)| = \left| \sum_{m=0}^{M-1} r_m(l) e^{-j2\pi dm/M} \right|. \quad (5)$$

Для каждого значения задержки l (5) представляет собой ДПФ вектора

$$[r_0(l), r_1(l), \dots, r_{M-1}(l)]^T$$

(^{"Т"} – символ транспонирования), которое также можно рассчитать по алгоритму БПФ для сокращения вычислений. Таким образом, для каждого значения l сечение ЧВФ вычисляется с помощью БПФ. На рис. 3 проиллюстрировано, как совпадающие по задержке отсчеты $r_m(l)$ объединяются в вектор для последующего расчета l -го сечения ЧВФ.

Перед БПФ вектор $[r_0(l), r_1(l), \dots, r_{M-1}(l)]^T$ должен быть дополнен нулевыми отсчетами так,

$r_0(0)$	$r_0(1)$...	$r_0(L)$
$r_1(0)$	$r_1(1)$...	$r_1(L)$
...
$r_m(0)$	$r_m(1)$...	$r_m(L)$
...
$r_{M-1}(0)$	$r_{M-1}(1)$...	$r_{M-1}(L)$

Рис. 3

чтобы длина дополненного вектора, во-первых, была не меньше $2M - 1$ и, во-вторых, была степенью двойки. Для снижения уровня боковых лепестков ЧВФ перед дополнением нулевыми отсчетами вектор следует умножить на оконную функцию (например, Хемминга) длиной M .

На основании изложенного алгоритм вычисления ЧВФ включает следующие операции:

1. Получение векторов s длиной N и s_{ref} длиной $L + N$.
2. Комплексное сопряжение вектора s_{ref} .
3. Разделение вектора s длиной N на M векторов s_m длиной N' (см. рис. 2).
4. Разделение вектора s_{ref}^* на M векторов $s_{\text{ref } m}^*$ длиной $N' + L$ (см. рис. 2).
5. Определение числа точек БПФ $r_m(l)$ (4): $N_{\text{FFT}r} = 2^{\text{nextpow}2(2N'+L-1)}$ (FFT – Fast Fourier Transformation – быстрое преобразование Фурье; $\text{nextpow}2(n)$ – функция, возвращающая минимальное p , такое, что $2^p \geq n$).
6. Дополнение каждого вектора s_m L нулями слева и $N_{\text{FFT}r} - L - N'$ нулями справа.
7. Дополнение каждого вектора $s_{\text{ref } m}^*$ $N_{\text{FFT}r} - L - N'$ нулями справа.
8. БПФ для всех векторов s_m и $s_{\text{ref } m}^*$ (далее – БПФ-8).
9. Перемножение векторов-результатов БПФ-8 для совпадающих m .
10. Обратное БПФ (далее – БПФ-10) для всех M векторов-результатов перемножения из п. 9.
11. Отбрасывание в каждом векторе-результате БПФ-10 последних $N_{\text{FFT}r} - (L + 1)$ отсчетов.
12. Формирование матрицы R_{ml} с размерами $M \times (L + 1)$ из векторов-результатов БПФ-10.
13. Умножение каждого столбца матрицы R_{ml} на оконную функцию длиной M .
14. Определение количества точек БПФ $\Psi(l, d)$ (5): $N_{\text{FFT}\Psi} = 2^{\text{nextpow}2(2M)}$.
15. Дополнение всех столбцов матрицы R_{ml} $N_{\text{FFT}\Psi} - M$ нулями.
16. БПФ (далее – БПФ-16) для каждого столбца матрицы R_{ml} с размерами $N_{\text{FFT}\Psi} \times (L + 1)$.
17. Формирование новой матрицы с размерами $(2D + 1) \times (L + 1)$.

18. Копирование в первые D столбцов сформированной матрицы из п. 17 последних D столбцов матрицы-результата из п. 16, а в последние $D + 1$ столбцов указанной матрицы – первых $D + 1$ столбцов матрицы-результата из п. 16.

Вычисление частотно-временной функции.

Относительно недавно у разработчиков появилась возможность выбора способа реализации цифровой обработки радиолокационных сигналов и, в частности, вычисления ЧВФ: конструировать и программировать специализированный вычислитель (на сигнальных процессорах и программируемых логических схемах) или использовать аппаратные средства персональных компьютеров (центральные процессоры, математические сопроцессоры, графические процессоры). Преимущества и недостатки разных подходов рассмотрены в [3]. В настоящей статье описано вычисление ЧВФ на графических процессорах (далее – видеокартах), а именно на широко распространенных игровых видеокартах.

Видеокарты привлекательны тем, что их архитектура позволяет организовать параллельные вычисления за счет многопроцессорности. Например, используемая в описанной далее реализации относительно недорогая видеокарта ASUS Radeon R9 290 содержит 2560 универсальных вычислительных процессоров. Кроме того, при параллельных вычислениях для увеличения суммарного объема памяти или для повышения производительности можно использовать не одну, а две и более видеокарт.

Первым этапом вычисления ЧВФ на видеокарте является подготовка данных для параллельных вычислений БПФ-8. Подготовка заключается в копировании отсчетов принятого и опорного сигналов в память видеокарты, разделении их на сегменты и дополнении нулями. Для минимизации количества операций по работе с памятью видеокарты (поскольку они являются самыми медленными) следует не копировать отсчеты в память и затем перемещать их там, дополняя нулями (при этом на каждый элемент обрабатываемого массива придется как минимум две операции копирования), а предварительно заполнить память нулями и, используя простой алгоритм, скопировать каждый элемент всего один раз. На рис. 4 это упрощенно проиллюстрировано для сигнала s (здесь подразумевается, что данные в память видеокарты копируются из оперативной памяти компьютера).

Несмотря на то, что при таком варианте приходится копировать каждый сегмент по отдельности (в отличие от копирования целиком всего блока от-

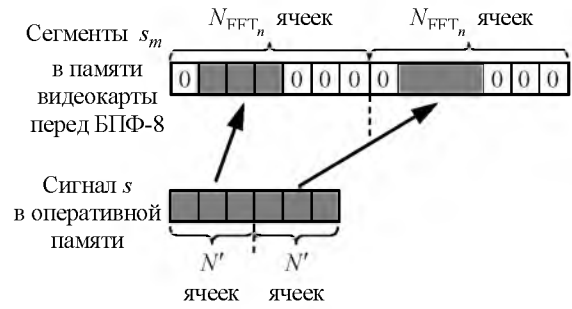


Рис. 4

счетов сигнала), временные потери оказываются невелики, так как передача данных осуществляется по технологии прямого доступа к памяти.

В рассмотренной реализации компьютерная программа вычисления ЧВФ создана с помощью OpenCL (фреймворк для написания компьютерных программ, связанных с параллельными вычислениями, содержащий свой язык программирования и интерфейс программирования приложений). БПФ (шаги 8 и 16 алгоритма вычисления ЧВФ), а также обратное БПФ (шаг 10 алгоритма) осуществлялись с помощью библиотеки cIFFT OpenCL (при этом настройка БПФ сводилась к указанию количества точек, типа данных, буферов для хранения входных и выходных данных). Требовалось обеспечить параллельное вычисление ЧВФ для 16 приемных каналов. Количество необходимой для обработки 16 каналов памяти превышало объем памяти одной видеокарты, поэтому параллельные вычисления выполнялись на двух видеокартах. При этом пришлось создать 2 контекста (рабочих пространства) для библиотеки cIFFT, каждый из которых содержал по одной видеокарте. На скорости вычислений использование двух контекстов вместо одного не сказывалось.

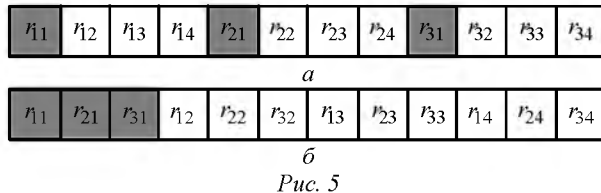
Перемножение спектров (шаг 9 алгоритма) выполнялось ядром OpenCL. OpenCL не содержит операций для работы с комплексными числами, поэтому они были реализованы с помощью специально написанных inline-функций. Для представления комплексного числа в OpenCL использован тип данных float2, содержащий два числа с плавающей точкой.

После выполнения обратного БПФ-10 выходной буфер данных (здесь и далее – для каждого приемного канала) содержал M векторов по N_{FFT} комплексных отсчетов в каждом. В процессе реализации шагов 11–15 алгоритма эти векторы преобразовывались в матрицу R_{ml} с размерами $N_{FFT} \Psi \times (L + 1)$.

Отсчеты матрицы содержались во входном буфере перед выполнением БПФ-16. В рассмотренной реализации оба буфера (выходной буфер после

обратного БПФ-10 и входной буфер перед БПФ-16) не могли быть размещены в памяти видеокарты одновременно. В связи с этим создавался промежуточный буфер меньшего размера, в который копировались $L + 1$ отсчетов каждого из M векторов, после чего выходной буфер обратного БПФ-10 уничтожался. Затем создавался и заполнялся нулями буфер для матрицы R_{ml} , после чего отсчеты из промежуточного буфера копировались в новый буфер по аналогии с тем, как это делалось при загрузке данных в память видеокарты перед выполнением БПФ-8 (см. рис. 4).

Матрицу R_{ml} следует ориентировать в памяти видеокарты так, как показано на иллюстративном примере (рис. 5) для матрицы размером 3×4 , чтобы друг за другом располагались отсчеты одной строки (рис. 5, а), а не столбца (рис. 5, б), несмотря на то, что БПФ выполняется для столбцов матрицы. Заливкой на рис. 5 выделены ячейки с элементами первой строки. Для операции выполнения БПФ расположение элементов в памяти не играет роли, но при последующем извлечении данных (в том числе из памяти видеокарты) для выполнения шагов 17, 18 алгоритма это важно, и размещение по строкам заметно экономит время.



Время вычисления частотно-временной функции. Описанным способом выполнялось параллельное вычисление 16 ЧВФ. Блок комплексных отсчетов для каждого из 16 принятых сигналов s имел длину $N = 2\,308\,198$ отсчетов. Опорный сигнал s_{ref} содержал $N + L = 2\,308\,198 + 4\,572 = 2\,312\,770$ отсчетов. Каждый сигнал разделялся на $M = 1262$ сегментов по $N' = 1829$ комплексных отсчетов в каждом (перекрывающиеся сегменты опорного сигнала содержали по $N' + L = 1829 + 4\,572 = 6401$ отсчету). Количество точек БПФ-8 $N_{FFTF} = 16\,384$. На шаге 8 алгорит-

ма для всех сигналов (16 принятых и одного опорного) суммарно выполнялось 21 454 БПФ на 16 384 точках. На шаге 10 алгоритма для всех сигналов выполнялось 20 192 обратных БПФ на 16 384 точках. Количество точек БПФ-16 $N_{FFTF} = 4096$ (матрица R_{ml} на этом шаге алгоритма имела размеры 4096×4573). На шаге 16 алгоритма для всех приемных каналов суммарно выполнялось 73 168 БПФ на 4 096 точках. Размеры выходной матрицы, представляющей ЧВФ, 1537×4573 . Расчет 16 ЧВФ на двух видеокартах ASUS Radeon R9 290 с учетом загрузки данных в память видеокарт и извлечения результатов в оперативную память компьютера занял 660 мс.

Полученный результат – время, затрачиваемое на расчет ЧВФ, позволяет, на взгляд авторов, рассматривать описанный вариант аппаратно-программной реализации цифровой обработки как приемлемый для использования в реальной практике. В частности, при интервале дискретизации порядка 100 нс блок отсчетов сигнала длиной, аналогичной приведенной ранее (несколько миллионов отсчетов), соответствует времени накопления несколько сот миллисекунд, соизмеримому со временем обработки на видеокартах. Можно сформулировать по-другому. В рассмотренном примере две видеокарты обработали около 40 млн отсчетов за 660 мс. Поскольку каждый отсчет является 64-разрядным, обрабатываемый поток составил 3.8 Гбит/с, что также является неплохим показателем. Рассмотренный вариант реализации вычисления ЧВФ позволяет увеличивать количество обрабатываемых приемных каналов без повышения времени вычислений за счет добавления видеокарт. С точки зрения стоимости рассмотренный вариант также привлекателен, так как обычно специализированный вычислитель стоит существенно дороже видеокарт. Программный код, написанный в OpenCL, не привязан к конкретной аппаратуре, что позволяет использовать видеокарты различных производителей. Во многих случаях гражданского применения, когда использование на РЛС компьютеров допускается техническими требованиями, применение видеокарт для цифровой обработки сигналов является реальной альтернативой.

СПИСОК ЛИТЕРАТУРЫ

1. Bistatic radar: emerging technology / ed. by M. Cherniakov. Chichester: John Wiley & Sons, 2008. 406 p.
2. Сергиенко А. Б. Цифровая обработка сигналов: учеб. пособие. 3-е изд. СПб.: БХВ-Петербург, 2011. 768 с.

3. Altera. May 2013. Radar Processing: FPGA or GPUs? WP-01197-2.0 White paper. URL: <http://www.altera.com/literature/wp/wp-01197-radar-fpga-or-gpu.pdf> (дата обращения: 03.09.2015).

A. V. Barkhatov, A. S. Kozlov
Saint Petersburg Electrotechnical University "LETI"

Radar amplitude-range-doppler surface fast calculation on graphics processing units

The paper describes the algorithm of fast calculation of the amplitude-range-Doppler surface. The features of the software implementation of the algorithm are drawn. The nuances of parallel computing of the surface on graphics processing units of the gaming graphics card are shown. The result, the sixteen surfaces parallel computing time for multi-channel radar, is presented.

Radar, digital processing, ambiguity function, cross-correlation, Doppler shift, delay, fast Fourier transformation, graphics processing units, parallel computing

Статья поступила в редакцию 7 сентября 2015 г.

УДК 621.396.9

С. В. Штанько, Д. А. Лесняк
Военно-космическая академия им. А. Ф. Можайского

Алгоритмы защищенного информационного обмена в радиоканалах космической навигационной системы

Для повышения защищенности информационного обмена по радиоканалам космической навигационной системы с целью предотвращения несанкционированного использования системы предложено использовать криптографические методы защиты сигнала высокой точности. Эту задачу позволяет решить криптографическая система, реализующая защищенные протоколы аутентификации, передачи ключей и информационного обмена.

Космическая навигационная система, межспутниковые каналы, алгоритм Диффи-Хеллмана, криптографическая защита

В настоящее время околоземное космическое пространство является сферой ведения военно-космической деятельности и применения космических средств различного назначения, а также рассматривается как стратегическая космическая зона.

На современном этапе развития радиотехнических систем особое место занимают космические навигационные системы (КНС).

Пространственная электромагнитная доступность, являющаяся свойством любых радиоканалов, создает условия для реализации угроз несанкционированного доступа к радиоканалам космического аппарата (КА). Нормальное функционирование КНС может быть нарушено в результате различных деструктивных воздействий.

Применение существующих протоколов защиты информации в радиоканалах навигационной системы ограничено особенностями ее функционирования, связанными в первую очередь с тем, что в ней существуют только однонаправленные каналы передачи информации "навигационный космический аппарат (НКА) – наземная

аппаратура потребителя (НАП)", что накладывает существенные ограничения на процедуры аутентификации и обмена ключевой информацией.

Одним из способов смены ключей является генерация новых ключей на основе совершения циклической операции со старым ключом (например, шифрование на старом ключе очередного числа, поступающего от датчика псевдослучайных чисел одинаковой для всех абонентов структуры) [1]. В этом случае, если наземный абонент не может расшифровать полученную информацию, он генерирует новый ключ и пытается расшифровать информацию с его помощью. Это позволяет абоненту восстановить текущий ключ системы независимо от того, сколько циклов смены ключа он пропустил [2]. Недостатком рассмотренного способа является возможность вскрытия структуры датчика псевдослучайных чисел, если вероятный нарушитель сможет получить в свое распоряжение экземпляр НАП. В дальнейшем, вскрыв один ключ, он также может получать последующие ключи с использованием датчика псевдослучайных чисел.