Синхронизация псевдослучайных последовательностей на практике: задача распознавания

Мария Беляева (Санкт-Петербург)

Необходимость в распознавании синхронизации псевдослучайных последовательностей возникает, если входящая последовательность содержит ошибки. Предлагается способ решения этой задачи. Исследуются условия его применимости, приводятся результаты моделирования.

Введение

Практические задачи обычно приходится решать в условиях той или иной неопределённости исходных данных. Это означает, что фактические значения многих входных величин, на основании которых принимается решение, неизвестны – имеются только их случайные оценки. Пренебрежение реальной неопределённостью часто приводит к тому, что вполне корректные алгоритмы оказываются бесполезными.

В статье [1] задача синхронизации псевдослучайных последовательностей (ПСП) решается в идеализированной постановке: предполагается, что входящая последовательность не содержит ошибок. В реальных условиях эта задача сложнее и интереснее.

Представим себе, что встроенный в передатчик генератор ПСП с линейным регистром обратной связи длины m вырабатывает последовательность $\{b_k\}$. В приёмнике, имеющем генератор с таким же регистром, надо решить задачу кодовой синхронизации, т.е. добиться, чтобы местный генератор начал вырабатывать ПСП $\{a_k\}$, совпадающую с передаваемой. Иначе говоря, начиная с некоторого момента, должно выполняться равенство $a_k = b_k$.

Такая задача решается, например, в широкополосных системах связи: цифровой информационный сигнал перед передачей скремблируется (суммируется с ПСП по модулю два), а сама ПСП передаётся отдельным пилот-каналом. Задача приёмника – захватить фазу этой последовательности, подстроиться под неё и начать расшифровку (дескремблирование) входящего сигнала.

Некоторые алгоритмы

В работе [1] предлагается подождать, пока местный генератор выработает целый период, т.е. $n = 2^m - 1$ бит, сложить их с n бит, поступивших на вход приёмника, и, воспользовавшись известными свойствами ПСП, найти сдвиг между двумя последовательностями. Тогда достаточно сдвинуть местный генератор на найденное число бит, и синхронизация будет достигнута.

Однако в условиях, для которых он предназначен, данный алгоритм слишком долго работает: для синхронизации надо получить n бит; для регистра из 15 ячеек это $2^{15} - 1$ бит. Существует гораздо более быстрый способ [2]: достаточно поместить m последовательных бит, выработанных генератором передатчика, в регистр местного генератора, и тот сразу начнёт вырабатывать последовательность, идентичную передаваемой. Синхронизация, таким образом, будет достигнута за m бит; для того же регистра это 15 бит.

Кроме того, упомянутые алгоритмы непригодны для практической реализации, поскольку разработаны для «стерильных» условий. В реальности передача любого сигнала сопровождается искажениями, обусловленными как параметрами среды, через которую он проходит, так и аппаратурными шумами. Это приводит к ошибкам приёма символов ПСП. Если из m бит, помещённых в местный генератор, по меньшей мере один бит будет ошибочным, синхронизация по [2] не состоится. Если же принятый набор из n бит содержит хотя бы одну ошибку, то, сложив его с содержимым местного генератора, мы не получим той же (смещённой) последовательности, как это предполагается в алгоритме [1]. Поскольку вероятность появления хотя бы одной ошибки в n бит существенно выше вероятности ошибки в m бит, первый алгоритм.

По-видимому алгоритмы синхронизации ПСП, не учитывающие возникающие ошибки, практического интереса не представляют. А приведённую выше постановку задачи следует дополнить важным условием: последовательность $\{b_k\}$ не известна точно. На приёмной стороне имеется лишь последовательность $\{c_k\}$ принятых бит, такая, что:

$$P\{b_k - c_k\} = 1 - p_{\text{err}},$$

где p_{err} – вероятность битовой ошибки в результате передачи.

В системах мобильной связи IS-95 CDMA [3] и CDMA 2000 [4] задача синхронизации решается методом «скользящего коррелятора» (СК): для каждого значения сдвига между ПСП, вырабатываемой местным генератором, и входящей последовательностью вычисляется их взаимная корреляция. Максимум корреляции соответствует точке синхронизации. Как справедливо отмечается [1], это – «долгоиграющий» метод, и годится он только для относительно коротких ПСП; в [5] алгоритм СК назван «расточителем времени».

Разнообразные алгоритмы синхронизации предлагаются во множестве работ (по некоторым данным, их не менее полутора тысяч), минимум одна из них содержит идею алгоритма, который практически гарантированно синхронизирует ПСП, причём делает это в среднем гораздо быстрее, чем алгоритм СК.

Предложенный в [6] остроумный метод RASE (rapid acquisition by sequential estimation) хорошо известен за рубежом – редкая статья о синхронизации обходится без ссылки на него. Метод реализуется в виде цепочки случайного числа экспериментов, каждый из которых осуществляется в

два этапа. На первом этапе приёмник помещает в свой регистр обратной связи m бит, полученных из физического канала (возможно, с ошибками). Второй этап – анализ результатов эксперимента. Если выясняется, что биты в регистре совпали с переданными битами, то синхронизация состоялась. В противном случае эксперимент повторяется для новых m бит из канала до тех пор, пока в регистр приёмника не будет загружен правильный набор.

Наборы не пересекаются – этим обеспечивается независимость экспериментов. Независимые испытания с ненулевой вероятностью успеха (испытания Бернулли) рано или поздно приводят к успеху, поэтому синхронизация неизбежна. Вероятность синхронизации в одном эксперименте оценивается как

$$x = (1 - p_{\text{err}})^m. \qquad (1)$$

Число экспериментов до наступления синхронизации случайно, его среднее значение [7]

$$N_{\text{aver}} = \frac{1}{x}, \qquad (2)$$

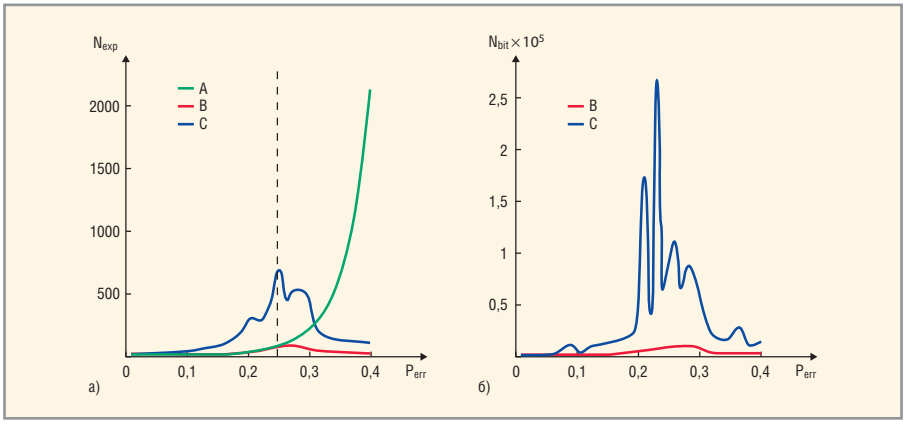

Рис. 1. а) Среднее число экспериментов до достижения фактической синхронизации (A); среднее (B) и максимальное (C) число экспериментов до достижения фактической или ложной синхронизации; б) среднее (B) и максимальное (C) число бит до достижения фактической или ложной синхронизации

(см. кривую A на рисунке 1а).

Итак, синхронизация рано или поздно состоится, остаётся лишь понять, когда это произойдёт. Иначе говоря, в результате каждого эксперимента следует решить, произошла ли синхронизация (понятно, что данная задача имеет смысл лишь в том случае, когда нет возможности понять это непосредственно по содержанию полученной информации). Если да, то процесс заканчивается. В противном случае эксперименты будут продолжены. Эта задача распознавания синхронизации и составляет содержание второго этапа эксперимента. Её формулировка несколько напоминает шутку о стоящих часах, которые показывают верное время дважды в сутки, только неизвестно, когда именно.

Для принятия решения будем анализировать «хвосты» двух последователь-

ностей, а именно: ПСП, вырабатываемой местным регистром после загрузки, и входящей последовательностью бит.

Как и любая другая задача принятия решения, распознавание допускает различные подходы. Можно потребовать, например, чтобы вероятность ошибочного решения не превышала заданного значения, и определять длину исследуемых хвостов, исходя из этого требования. Другой вариант состоит в том, чтобы ограничить время исследования (число бит в хвосте): чем оно меньше, тем выше будет вероятность ошибки.

В работе [6] задачу распознавания предлагается решать путём сравнения взаимной корреляции между хвостами двух последовательностей с заранее заданным порогом. Если за время исследования T_e он превышен, синхронизация считается состоявшейся. Однако в описании алгоритма имеются явные логические противоречия. Так, T_e не только не зависит от вероятностей p_f и p_m ошибок ложной и пропущенной синхронизации соответственно, но и задаётся вместе с ними как независимая переменная. Далее, в [6] выбор порога не зависит от p_{err}, хотя очевидно, что должен зависеть, поскольку при состоявшейся синхронизации в хвостах ПСП будет тем больше несовпадений, чем больше шум. Эти и другие противоречия не позволяют использовать данный алгоритм распознавания при реализации эффектной идеи метода RASE.

Предлагаемый ниже алгоритм распознавания синхронизации свободен от указанных противоречий: пороговое условие выбирается в зависимости от вероятности perr и требований к pf и pm; необходимая для анализа длина хвостов (время Te) также зависит от них и определяется в процессе решения.

Распознавание в одном эксперименте

Пусть m входных бит помещены в местный регистр сдвига. Рассмотрим две возможности:

- синхронизация есть. Следовательно, $a_k = b_k$, и различие $a_k \neq c_k$ может возникнуть только из-за наличия ошибок при передаче; в этом случае $b_k \neq c_k$. Вероятность того, что последнее неравенство выполняется, равна вероятности битовой ошибки p_{err}. Побитовое сравнение двух последовательностей есть последовательность независимых испытаний Бернулли с постоянной вероятностью успеха (несовпадения a_k и c_k), равной p_{err};
- синхронизации нет. Приходящие биты могут отличаться от вырабатываемых в двух случаях: может быть $a_k = b_k$, но $b_k \neq c_k$ из-за ошибки передачи, и, следовательно $a_k \neq c_k$, а может быть и $a_k \neq b_k$ потому, что в отсутствие синхронизации они не обязаны быть равными. В последнем случае возможно как $c_k \neq a_k$, так и $c_k = a_k$.

При отсутствии синхронизации $\{b_k\}$ и $\{a_k\}$ суть безошибочные ПСП, получающиеся одна из другой сдвигом. Следовательно, по известному свойству ПСП

$$P\{b_k = a_k\} \approx 0{,}5, k = 1, ..., m.$$

Тогда

$$P\{c_k \neq a_k\} = P\{a_k = b_k, c_k \neq b_k\} + P\{a_k \neq b_k, c_k = b_k\}.$$

Учитывая независимость двух событий в каждой паре фигурных скобок, получим, что побитовое сравнение хвостов при отсутствии синхронизации является последовательностью испытаний Бернулли с вероятностью 0,5.

Теперь достаточно подсчитать число несовпадающих пар бит в хвостах, чтобы определить, какая из двух вероятностей несовпадения бит «вероятнее». Если это 0,5, синхронизации нет, если p_{err} – синхронизация есть. Сравнивать хвосты будем порциями по s бит: если решение не удалось, принять на основе первой порции, добавить к ней вторую, и т.д.

Пусть требуется, чтобы вероятности ложной и пропущенной синхронизации в эксперименте удовлетворяли условиям

$$p_f \le 1 - \beta$$
$$p_m \le 1 - \beta \qquad (3)$$

Пусть частота несовпадений в первых N порциях, т.е. в Ns бит, равна p_{nc}, тогда вероятность несовпадений находится внутри доверительного интервала (g', g'') с доверительной вероятностью β. Здесь, согласно [8],

$$g' = p_{nc} - t_\beta \sqrt{\frac{p_{nc}(1-p_{nc})}{Ns}},$$

$$g'' = p_{nc} + t_\beta \sqrt{\frac{p_{nc}(1-p_{nc})}{Ns}}, \qquad (4)$$

а t_β зависит от выбранного значения β, например,

$$t_\beta = 1{,}643 \text{ при } \beta = 0{,}9,$$
$$t_\beta = 2{,}576 \text{ при } \beta = 0{,}99.$$

Если выполняется неравенство

$$g' < p_{err} < g'' < 0{,}5, \qquad (5)$$

то с вероятностью β синхронизация состоялась. Если же верно неравенство

$$p_{err} < g' < 0{,}5 < g'', \qquad (6)$$

то с той же вероятностью синхронизации нет. Если ни то, ни другое неравенство не выполняется, добавим к уже рассмотренным битам следующую порцию и повторим расчёты. Процесс сходится, ибо с увеличением числа анализируемых бит доверительный интервал уточняется, и рано или поздно одно из значений p_{err} и 0,5 окажется внутри, а второе – вне его.

Решение о наличии синхронизации при её фактическом отсутствии (ложная синхронизация) будет принято, если истинная вероятность несовпадений 0,5 окажется вне доверительного интервала. Решение об отсутствии синхронизации при её наличии (пропущенная синхронизация) – если вне доверительного интервала окажется истинная вероятность p_{err}. Таким образом, последовательно наращивая хвосты, мы добьёмся выполнения условия (3). Чем p_{err} ближе к 0,5, тем труднее их различить и тем более длинные хвосты потребуются для этого. При p_{err} = = 0,5 всё это, естественно, не работает, ибо частота несовпадений как при синхронизации, так и без неё будет близка к 0,5.

Если ни одно из неравенств (5), (6) не выполняется, то добавляется следующая порция бит и анализ повторяется.

Теперь предположим, что требования к вероятностям ложной и пропущенной синхронизации различны, т.е. должно выполняться

$$p_f \le 1 - \beta_1$$
$$p_m \le 1 - \beta_2. \qquad (7)$$

Подставив поочередно β_1 и β_2 в (4), получим формулы для расчёта доверительных интервалов (g_1', g_1'') и (g_2', g_2''). Очевидно, первое неравенство в (7) будет выполнено, если интервал (g_1', g_1'') накроет величину p_{err}, а второе – если интервал (g_2', g_2'') накроет 0,5. Таким образом, решение о наличии синхронизации следует принимать, если

$$g_1' < p_{err} < g_1'' < 0{,}5,$$

а об её отсутствии – если

$$p_{err} < g_2' < 0{,}5 < g_2''.$$

Распознавание в цепочке экспериментов

Итак, вероятности ошибок в одном эксперименте не превысят заданных значений. Однако для практического применения более важны интегральные характеристики данного способа, а именно, вероятности принятия ошибочного решения в целом и длительность вычислений.

Пусть P_F – вероятность того, что достигнутая синхронизация является ложной, а P_M – вероятность хотя бы одной пропущенной синхронизации во всей цепочке экспериментов. Очевидно,

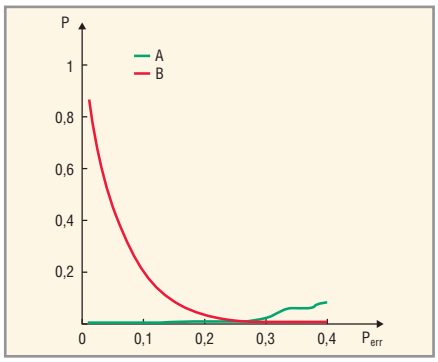

Рис. 2. Вероятности синхронизации в эксперименте: ложной (A) и фактической (B)

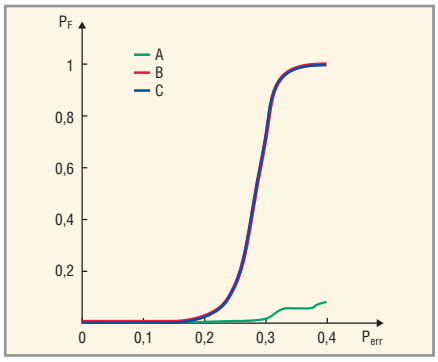

Рис. 3. Вероятность ложной синхронизации в эксперименте (A); вероятность того, что синхронизация в конце цепочки экспериментов окажется ложной (B – по формуле (11), C – по результатам моделирования)

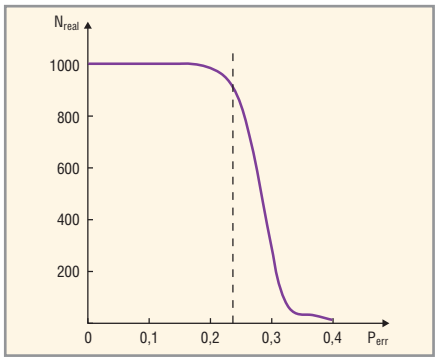

Рис. 4. Число цепочек (из 1000), закончившихся фактической синхронизацией

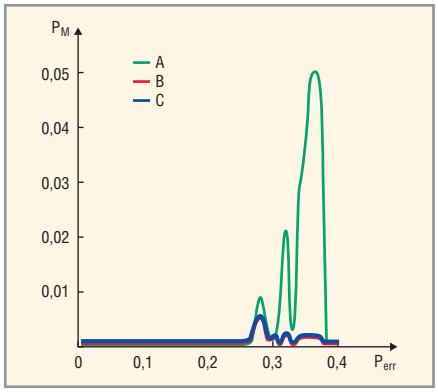

Рис. 5. Вероятность пропуска синхронизации в эксперименте (A); вероятность хотя бы одного пропуска синхронизации в цепочке экспериментов (B – по формуле (10), C – по результатам моделирования)

$$P_M = \sum_{i=1}^{\infty} P\left\{\begin{array}{l}\text{синхронизация пропущена хотя бы в одном из } i \text{ экспериментов,}\\ \text{всего в цепочке } i \text{ экспериментов}\end{array}\right\}.$$

Учитывая независимость экспериментов и то, что в последнем эксперименте цепочки синхронизация не может быть пропущена, получим

$$P_M = \sum_{i=1}^{\infty}(P_1 \times P_2), \qquad (8)$$

где

$$P_1 = P\left\{\begin{array}{l}\text{синхронизация пропущена хотя бы в одном из } (i-1) \text{ экспериментов,}\\ \text{эти эксперименты – не последние в цепочке}\end{array}\right\},$$

$P_2 = P\{i\text{-й эксперимент – последний в цепочке}\}$.

Очевидно, что

$P\{(i-1)$ экспериментов – не последние в цепочке$\} = P_1 +$

$$+ P\left\{\begin{array}{l}\text{синхронизация не пропущена ни в одном из } (i-1) \text{ экспериментов,}\\ \text{эти эксперименты – не последние в цепочке}\end{array}\right\}.$$

Отсюда, учитывая независимость экспериментов,

$$P_1 = P_3^{i-1} - P_4^{i-1}, \qquad (9)$$

где $P_3 = P\{$эксперимент – не последний в цепочке$\}$,

$$P_4 = P\left\{\begin{array}{l}\text{эксперимент – не последний в цепочке}\\ \text{в нём не пропущена синхронизация}\end{array}\right\}.$$

Далее,

$$P_3 = P\left\{\begin{array}{l}\text{в эксперименте нет фактической синхронизации,}\\ \text{нет ложной синхронизации}\end{array}\right\} +$$

$$+ P\left\{\begin{array}{l}\text{в эксперименте есть фактическая синхронизация,}\\ \text{она пропущена}\end{array}\right\}$$

Первое слагаемое в последнем равенстве преобразуем к виду:

$P\{$в эксперименте нет фактической синхронизации$\}$ –

$$- P\left\{\begin{array}{l}\text{в эксперименте нет фактической синхронизации,}\\ \text{есть ложная синхронизация}\end{array}\right\} = 1 - x - p_f(1-x),$$

второе равно $p_m x$. В свою очередь,

$$P_4 = P\left\{\begin{array}{l}\text{в эксперименте нет фактической синхронизации,}\\ \text{нет ложной синхронизации}\end{array}\right\} = (1-x)(1-p_f).$$

Далее,

$$P_2 = P\left\{\begin{array}{l}\text{в эксперименте есть фактическая синхронизация,}\\ \text{она не пропущена}\end{array}\right\} +$$

$$+ P\left\{\begin{array}{l}\text{в эксперименте нет фактической синхронизации,}\\ \text{есть ложная синхронизация}\end{array}\right\} = (1-p_m)x + p_f(1-x).$$

После подстановки P_3, P_4 в (9), полученное при этом значение P_1 а также P_2 – в (8) получим значение P_1, тогда:

$$P_M = \sum_{i=1}^{\infty}\left(\left[(1-x)(1-p_f)+p_m x\right]^{i-1} - \left[(1-x)(1-p_f)\right]^{i-1}\right)\left[(1-p_m)x + p_f(1-x)\right].$$

Наконец, по формуле для суммы геометрической прогрессии получим:

$$P_M = \frac{p_m x}{x + p_f(1-x)}. \qquad (10)$$

Аналогичным образом получим:

$$P_F = \frac{p_f(1-x)}{p_f(1-x) + x(1-p_m)}. \qquad (11)$$

Полученные равенства можно использовать для расчёта конкретных значений P_M, P_F после предварительной оценки ошибок p_m и p_f.

Описание модели

Начальное состояние передающего регистра является случайным, его биты искажаются с вероятностью P_{err} и помещаются в местный регистр. Эксперименты продолжаются до наступления синхронизации, а является ли она фактической или ложной, выясняется сравнением начальных состояний регистров. В конце каждой реализации подсчитывается число пропущенных синхронизаций. По результатам всех реализаций рассчитываются средние значения и фактические вероятности.

Результаты моделирования

Расчёты проводились для трёх различных регистров с 15 ячейками, результаты практически совпали. Выполнялось 1000 реализаций. Длина порции бит принята $s = 100$ [8]. Вероятности ложной и пропущенной синхронизации в каждом эксперименте не должны превышать 0,1.

На рис. 1а график B характеризует зависимость среднего числа экспериментов от вероятности P_{err} битовой ошибки при синхронизации ПСП с вероятностями ошибок в одном эксперименте, не превышающих заданных значений. До некоторой точки, соответствующей примерно $P_{err} = 0{,}24$, он совпадает с теоретическим значением (2) (график A), а затем резко с ним расходится. Многие цепочки обрываются, «не дожив» до фактической синхронизации, поскольку ложная синхронизация наступает раньше.

Если вероятность (1) фактической синхронизации уменьшается с ростом P_{err} (график B рис. 2), то вероятность ложной синхронизации (график A) растёт, оставаясь при этом не больше 0,1. Происходит это потому, что два значения (P_{err} и 0,5), которые надлежит разделить с помощью доверительного интервала, сближаются; растёт в среднем и длина доверительного интервала. Вероятность ошибок при этом увеличивается.

По мере сближения вероятностей p_f и x наступление ложной синхронизации прежде фактической становится всё вероятнее, это объясняет характер кривой B на рис. 1а.

График C на рис. 1 представляет максимальное число экспериментов за все реализации. По графикам B и C среднего и максимального числа бит до принятия решения в цепочке экспериментов легко рассчитать время синхронизации, если задать битовую скорость передачи.

На рис. 3 графики вероятности ложной синхронизации: A – вероятность p_f, практически совпадающие графики B и

C – вероятность P_F, рассчитанная по результатам моделирования и по формуле (11). При $P_{err} > 0{,}3$ синхронизация становится ненадёжной, почти наверное она окажется ложной. Это подтверждает рис. 4, где показано число тех реализаций, в которых была достигнута фактическая синхронизация. До $P_{err} \approx 0{,}23$ все или почти все цепочки заканчиваются фактическими синхронизациями, затем доля ложных синхронизаций возрастает, приближаясь к 90% и более после 0,3.

На рис. 5 приведены графики вероятности пропуска синхронизации: A – вероятность p_m, B и C – фактическая и рассчитанная по формуле (10) вероятность P_M; последние две кривые практически совпадают. Резкие колебания p_m, начиная с $P_{err} \approx 0{,}3$, объясняются уменьшением числа фактических синхронизаций: p_m оценивается как отношение числа цепочек, содержащих пропуск синхронизации, к общему числу фактических синхронизаций; поэтому чем меньше знаменатель (рис. 4), тем хуже оценка.

Можно рекомендовать применение алгоритма RASE вместе с предлагаемым способом распознавания для регистра с 15 ячейками при $P_{err} < 0{,}2$. При более вероятных битовых ошибках этот способ ненадёжен.

Заключение

Представлен способ распознавания синхронизации в алгоритме RASE, свободный от противоречий исходного способа распознавания [6]. Его работоспособность подтверждена как аналитическими зависимостями (10), (11), так и результатами моделирования. Исследовательская модель позволяет определять условия применимости алгоритма RASE и подбирать требования к вероятностям ошибок в одном эксперименте, позволяющие добиться характеристик синхронизации.

Время распознавания синхронизации можно сократить путём «запараллеливания» анализа последовательностей. Достаточно смоделировать много одинаковых регистров обратной связи и проводить анализ вырабатываемых ими последовательностей одновременно. Каждый регистр освобождается, как только выяснилось, что синхронизация в нём не произошла, и загружается новыми m битами. Процесс останавливается, как только произойдёт синхронизация в одном из регистров. Расчёты показали, что использование 50 регистров вместо одного позволяет сократить среднее время синхронизации в шесть и более раз. Целесообразность такого усовершенствования определяется требованиями к решению задачи и вычислительными возможностями.

Литература

1. *Калугин Е.* Поиск и синхронизация псевдослучайных последовательностей. Современная электроника. 2009. № 9. С. 30–31.
2. *Кирьянов К.Г., Меднов А.С., Акулов В.В.* Синхронизация генераторов псевдослучайных последовательностей. Техника средств связи, ЭКОС. Серия РИТ, 1990. Вып. 1. С. 56–64.
3. *Yang S.C.* CDMA RF System Engineering. Artech House, 1998.
4. *Yang S.C.* 3G CDMA2000 Wireless System Engineering. Artech House. Boston – London, 2004.
5. *Прокис Дж.* Цифровая связь. Радио и связь, 2000.
6. *Ward R.B.* Acquisition of Pseudonoise Signals by Sequential Estimation. IEEE Trans. Commun. Tech., December,1965.
7. *Феллер В.* Введение в теорию вероятностей и её приложения. Т. 1. Мир, 1967.
8. *Вентцель Е.С.* Теория вероятностей. Государственное издательство физико-математической литературы. Москва, 1962.