

Алгоритм символьной синхронизации по отсчетам комплексной огибающей с элементами адаптивности для модуляции DQPSK

Д. А. Гончаров, И. Н. Максимов, О. Г. Мелентьев

Сибирский государственный университет телекоммуникаций и информатики (СибГУТИ)

Аннотация: В статье предложен алгоритм символьной синхронизации по отсчетам комплексной огибающей, без использования фазовой автоподстройки, частоты (ФАПЧ) для DQPSK модуляции. Проведено моделирование данного алгоритма, определено среднее время вхождения в синхронизм и вероятность ошибок.

Ключевые слова: символьная синхронизация, вероятность ошибки, время синхронизации, ошибка синхронизации.

Для цитирования: Гончаров Д. А., Максимов И. Н., Мелентьев О. Г. Алгоритм символьной синхронизации по отсчетам комплексной огибающей с элементами адаптивности для модуляции DQPSK // Вестник СибГУТИ. 2024. Т. 18, № 4. С. 76–86. <https://doi.org/10.55648/1998-6920-2024-18-4-76-86>.



Контент доступен под лицензией
Creative Commons Attribution 4.0
License

© Гончаров Д. А., Максимов И. Н.,
Мелентьев О. Г., 2024

Статья поступила в редакцию 09.04.2024;
переработанный вариант – 03.06.2024;
принята к публикации 11.07.2024.

1. Введение

Рассмотрим цифровые системы передачи (ЦСП), в которых данные предаются с помощью последовательно передаваемых символов равной длины. Символы представляют сигналы определённой формы, каждый из которых принадлежит некоторому множеству. Количество сигналов в этом множестве, их форма и способы их формирования называются типом модуляции. Различные типы модуляции имеют множество характеристик, которые в существенной мере определяют облик ЦСП, её характеристики и сложность реализации. Тем не менее все ЦСП должны так или иначе решить три задачи, без решения которых ЦСП работать не будет. А именно: фазовая синхронизация, символьная синхронизация и кадровая синхронизация. Для некоторых типов модуляции эти задачи редуцируются до предела. Например, если данные передаются с помощью амплитудной модуляции, то необходимость в фазовой синхронизации отпадает. Но такая модуляция несёт в себе другие недостатки.

Рассмотрим в качестве примера ЦСП, использующую модуляцию BPSK. В такой системе данные передаются с помощью опорного гармонического колебания на некоторой частоте. Причём на протяжении символов, соответствующих единице, передаваемый сигнал синфазен этому колебанию, а на протяжении символов, соответствующих нулю, находится в противофазе. Для приёма такого сигнала необходимо сначала произвести фазовую синхронизацию с помощью ФАПЧ, для того чтобы воссоздать в приёмнике копию опорного сигнала передатчика. После чего, сравнивая принимаемый и опорный сигнал, сформировать сигнал разности фаз между ними. После этого начинает работать символьная синхронизация, которая найдёт в этом сигнале границы между символами и проинтегрирует его в

промежутках между границами. Далее получившаяся последовательность бит может быть подвергнута кадровой синхронизации.

При создании использующейся в этой работе ЦСП авторы руководствовались различными требованиями, диктуемыми соображениями исследовательского и методического характера. Одним из таких требований было отсутствие в дизайне системы ФАПЧ. Обойтись без ФАПЧ для детектирования символов можно, используя дифференциальную фазовую модуляцию. Для этого надо сравнивать символы не с опорным колебанием, а друг с другом. Но для этого необходимо провести символьную синхронизацию и эти символы получить. Получается некий замкнутый круг, классическим способом преодоления которого является вычисление сигнала разности фаз между текущим сигналом и им же, но задержанным на один символ. С описанием такого способа можно ознакомиться в работе [1]. Заметим, что сигнал разности фаз является действительным, а не комплексным, и со способом тактовой синхронизации по такому сигналу можно ознакомиться в работе [2]. Недостатком использования сигнала разности фаз является то, что он не подвергался интегрированию на промежутках между символами и таким образом сильнее зашумлён, чем сами символы. Авторы предложили способ проведения символьной синхронизации для получения символов без проведения предварительной фазовой синхронизации по исходному комплексному сигналу, описанный в статье [3], что позволяет вообще обойтись без сигнала разности фаз. В представляемой работе данный способ расширен для двухкратной фазовой модуляции DQPSK.

Необходимо отметить, что любой алгоритм характеризуется теми или иными числовыми параметрами, в зависимости от которых он может работать по-разному. Если планируется практическое применение такого алгоритма, то в задачу его описания необходимо включить способ определения оптимальных значений этих параметров. Что, в свою очередь, невозможно без формулировки критериев оптимальности. Кроме того, в большинстве случаев зависимости между параметрами и результатами работы алгоритма имеют такой характер, что оптимальный результат достигается при разных параметрах в зависимости от внешних условий. В таком случае имеет смысл рассмотреть вопрос об изменении параметров в процессе работы алгоритма. То есть о включении в алгоритм элементов адаптивности. Эти вопросы также рассматриваются в данной работе.

2. Термины и определения

В работе будут использоваться следующие термины:

Символ – отрезок сигнала, который мы получили в результате модуляции, символ отображает объём информации, равный одному или нескольким битам, в зависимости от кратности модуляции.

Семпл – отсчет, получившийся после дискретизации символов. Семплы мы формируем для последующей передачи и обрабатываем при приеме. По группе семплов, при помощи символьной синхронизации, мы должны определить символ.

Глубина дискретизации (n) – отношение частоты дискретизации к количеству символов, передаваемых за секунду или, что то же самое, количество семплов в одном символе. В данной работе рассматривается случай, при котором каждый символ содержит 8 семплов, $n = 8$. Увеличение количества семплов относительно количества символов увеличивает полосу цифрового сигнала и позволяет применить цифровые фильтры для формирования спектра передаваемого и фильтрации получаемого сигналов. Итоговая ширина спектра в эфире будет определяться формой АЧХ и шириной полосы цифрового фильтра, который обрабатывает сигнал перед его передачей на ЦАП.

Окно синхронизации – S , это группа последовательных семплов с количеством, равным $n/2 = 4$, которые мы рассматриваем для определения символов. Предлагаемый в данной работе алгоритм основан на сравнении сигналов в четырёх таких окнах. Эти окна

обозначаются $S1$, $S2$, $S3$, $S4$. Под обозначением S мы подразумеваем сумму семплов в данном окне.

Группа окон – это сумма смежных окон, $S12 = S1 + S2$ и т. п.

$S1_i$ – синфазная составляющая, $S1_q$ – квадратурная составляющая.

Дальше термины и определения будут вводиться по ходу работы.

$S12$ – это сумма семплов первого символа, $S34$ – второго (идущего после первого) символа, $S23$ – 4 последних семпла первого символа и 4 первых семпла второго символа.

3. Алгоритм символьной синхронизации по действительным отсчетам

Для упрощения понимания алгоритма, рассмотрим пример с действительными отсчетами в идеальных условиях (отсутствие шума и затухания). Так как алгоритм производит анализ только когда символы в окнах имеют разные знаки, для уменьшения времени синхронизации в начало каждого кадра добавим меандр. Меандр – это чередование единиц и нулей в битах, это чередование 1 и -1 в символах, а в семплах – это n подряд идущих семплов одного и потом другого знака.

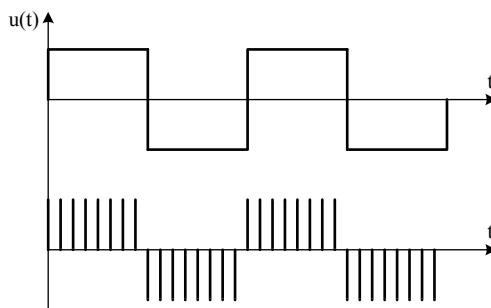


Рис. 1. Прямоугольный действительный сигнал (меандр) и его отсчеты

На приеме после дискретизации мы получаем поток отсчетов. В нашем случае на каждый символ приходится восемь подряд идущих отсчетов. Наша задача определить положение первых семплов каждого символа.

Для определения границ символов мы будем помещать по 16 смежных семплов в окна синхронизации, проводить анализ и, в случае необходимости, на следующем шаге смещать положение окон на 7 или 9 отсчетов. Такое смещение, соответствует относительному смещению окон на ± 1 отсчет.

В сумме окна синхронизации содержат $n \cdot 2 = 16$ семплов, которые в свою очередь можно определить, как два подряд идущих символа. $S12$ – первый символ (детектируемый, ранний), $S34$ – второй символ (следующий, поздний). Само собой, если в $S12$ попали семплы одного символа, то это и есть случай полной синхронизации – мы можем определять символ с минимальной вероятностью ошибки.

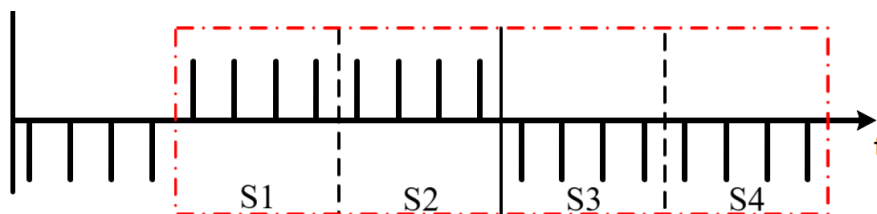


Рис. 2. Состояние синхронизма на примере меандра

Если знаки $S12$ и $S34$ равны, то алгоритм пропускает шаг подстройки синхронизации, так как не сможет узнать в какую сторону смещаться.

$S_{12} = 8$ и $S_{34} = -8$ – знаки разные. Алгоритм подстройки начинает работать.

В S_{23} при полной синхронизации мы увидим значение ноль, так как в S_2 находятся $\frac{n}{2} = 4$ семпла первого символа, а в S_3 – второго символа. $S_2 = 4$, $S_3 = -4$, $S_{23} = 0$. Это значит, что мы сейчас находимся в устойчивом состоянии синхронизма, нам не нужно смещать окна и можем детектировать символ. В случае DBPSK детектируемый бит определяется знаком суммы S_{12} .

Далее окна синхронизации смещаются на n семплов в потоке семплов. В S_{12} переходит символ, который был в окне S_{34} на предыдущем шаге, а в S_{34} приходит следующий символ. И каждый шаг снова повторяет вышеупомянутые действия.

Если получилось так, что в S_{12} попали семплы из разных символов, то мы говорим об рассинхронизации на некоторое количество семплов (Cor). В окне S_{12} ($Cor = 1$) присутствует 7 семплов одного символа и 1 семпл другого символа (см. рис.3).

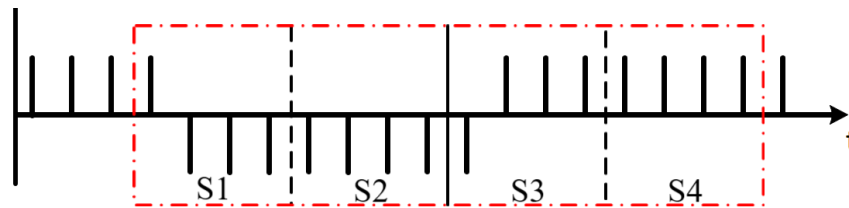


Рис. 3. Рассинхронизация на один семпл

Окно S_{23} нам не покажет ноль. Это значит, что граница двух символов смещена. В одном из рассматриваемых окон (S_2 или S_3) будут семплы одинакового знака, а вот в другом окне будет один семпл противоположного знака.

Если $S_{23} > 0$, то граница в окне с меньшим значением суммы. В нашем случае (рис. 3) $S_{23} = -2 < 0$, т. е. преобладают минус единицы, значит символ меняет знак в области с большим значением суммы (там, где будут и положительные и отрицательные отсчеты). В данном случае надо сместить границу в сторону окна S_3 , вправо, на более поздний срок. Такое смещение назовем положительным. Можно сформулировать логику работы алгоритма по-другому. $S_{12} = -6$, $S_{34} = +6$, а $S_{23} = -2$. Видим, что S_{23} ближе к S_{12} , чем к S_{34} . Значит для выравнивания нужно двигать окна в сторону S_{34} , т. е. на более поздний срок, пропустить отсчет и сместиться на 9 отсчетов.

В алгоритме предусмотрен аккумулятор, который накапливает значения смещений, полученных на каждом шаге подстройки, до достижения некоторого порога (por). Величина и знак смещения вычисляются по формуле

$$Sm = \text{sign}(S_{12}) \cdot S_{23} \quad (1)$$

где произведение знака группы окон S_{12} – $\text{sign}(S_{12})$ и суммы S_{23} показывают нам, направление смещения окон синхронизации.

В рассматриваемом примере знак S_{12} и значение S_{23} отрицательные, поэтому в аккумулятор добавится значение положительного смещения равное $Sm = 2$. При достижении аккумулятором порогового значения (влияние порога показано далее), алгоритм сместит все окна на $n + 1 = 9$ семплов, и сумма S_{23} станет равна нулю (Рис. 2.).

Алгоритм будет работать аналогично при рассинхронизации на 2 или 3 семпла.

Оговорим стационарное состояние алгоритма при смещении на 4 семпла ($Cor = 4$). В окнах S_{12} и S_{34} окажутся равные половины двух символов, а в окне S_{23} – символ целиком. Определить направление смещения в данном случае невозможно. Выход из данного состояния возможен только при воздействии шума или искусственного вмешательства.

Вышеописанный алгоритм требует предварительной фазовой синхронизации. Для работы без ФАПЧ можно использовать версию с комплексными числами.

4. Комплексный алгоритм символьной синхронизации для DBPSK модуляции

При комплексном алгоритме соседние символы меандра также изменяют свою фазу относительно предыдущего на 180 градусов, но могут иметь вид как: 1, -1 так и $0 + i1$, $0 - i1$. Сумма окна в свою очередь теперь тоже будут иметь комплексные значения, например, $S1 = 4 + i0$ или $S34 = 0 + 8i$.

При относительной фазовой модуляции мы изменяем фазу текущего символа относительно предыдущего на 180 градусов при передаче единицы или не изменяем при передаче нуля. Данная операция эквивалентна умножению предыдущего символа на комплексную экспоненту угла поворота.

$$X_i = X_{i-1} \cdot e^{j\varphi}$$

Значит для получения информации о разнице фаз соседних символов, достаточно поделить одно комплексное число на другое.

$$e^{j\varphi} = \frac{X_i}{X_{i-1}}$$

Причем для DBPSK нам достаточно определить знак действительной части разницы фаз, так как на модуляторе, меняя фазу на 180, мы просто умножали на -1. И, если действительная часть от разницы фаз будет меньше 0, значит фаза была изменена.

Формула для деления комплексных чисел приведена ниже:

$$\frac{a + j \cdot b}{c + j \cdot d} = \frac{a \cdot c + d \cdot b}{c^2 + d^2} + j \cdot \frac{b \cdot c - a \cdot d}{c^2 + d^2} \quad (2)$$

Так как в нашем случае информация об амплитуде не используется, деление на $c^2 + d^2$ можно исключить. Действительная часть будет равна сумме произведений действительных частей символов и мнимых.

$$tr_i = S12_i \cdot S34_i + S12_q \cdot S34_q$$

По знаку этой формулы мы также определяем, одинаковые ли символы идут подряд в окнах S12 и S34. Если $tr_i < 0$, то анализируемые символы разных знаков. При этом, алгоритм синхронизации проводит вычисления, модифицирует значение аккумулятора, сравнивает с порогом и при превышении производит коррекцию. В противном случае, вычисления не производятся и все окна смещаются на $n = 8$ семплов для следующего шага.

Любое комплексное число можно представить вектором на комплексной плоскости. На рисунке 4 представлены три вектора комплексных сумм окон: S12, S23, S34, а также разности векторов.

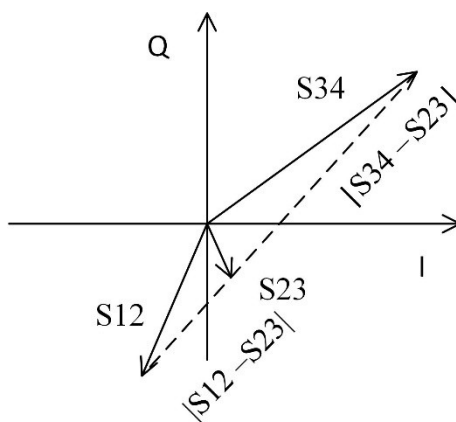


Рис. 4. Идея алгоритма синхронизации по комплексным отсчетам в векторном виде

Длины векторов можно вычислить по следующим формулам

$$\begin{aligned} |S_{12}|=r &= \sqrt{(S_{23_i})^2 + (S_{23_q})^2}, \\ |S_{12} - S_{23}|=r &= \sqrt{(S_{12_i} - S_{23_i})^2 + (S_{12_q} - S_{23_q})^2}, \\ |S_{34} - S_{23}|=r &= \sqrt{(S_{34_i} - S_{23_i})^2 + (S_{34_q} - S_{23_q})^2}. \end{aligned}$$

Для интерпретации рисунка воспользуемся аналогией с действительным алгоритмом. Из рисунка видно, что длина вектора разности $S_{12}-S_{23}$ меньше, чем длина вектора $S_{34}-S_{23}$. Значит, сумма S_{23} содержит больше векторов, преобладающих в сумме ранее пришедшего символа S_{12} . Поэтому, для достижения баланса нужно смещать окна синхронизации в сторону позднего символа S_{34} , т. е. пропустить 1 отсчет или сместить окно на 9 отсчетов.

5. Применение комплексного алгоритма синхронизации для DQPSK сигналов

Рассмотрим применение алгоритмов для определения фаз, символьной синхронизации и детектирования применительно к DQPSK.

Отличие двух модуляций (DBPSK и DQPSK) заключается в нескольких аспектах.

Во-первых, каждый символ в DBPSK содержит один бит информации 1 или 0. В DQPSK – каждый символ несет в себе один из четырех дибитов, их значения: 00, 01, 10, 11.

Во-вторых, в модуляции DBPSK мы создаем символы двумя действиями, поворачивая их на комплексной плоскости (меняем фазу символа) на 180 или не поворачиваем в зависимости от входного бита, что тождественно умножению на $\exp(i \cdot 180)$ или на $(-1 + i \cdot 0)$. Если символ равен единице, то мы поворачиваем его на 180 градусов относительно предыдущего символа. Самый первый символ может иметь любую начальную фазу. Для упрощения мы берем начальную фазу, равную нулю.

В DQPSK, в зависимости от значения дибита, мы поворачиваем на 180°, если дибит равен 11, на 90°, если 01, на -90°, если 10 и не меняем фазу, если 00.

Очевидно, что для демодуляции символа нам нужно определить фазу точнее, чтобы различать сдвиги на 90 и на 180 градусов.

Алгоритм подстройки окон синхронизации и выражения для вычисления длин векторов остаются неизменными.

При детектировании DBPSK мы определяли символ по знаку действительной разности фаз двух символов. В DQPSK для определения символа и последующего детектирования, нам нужно иметь синфазную и квадратурную составляющие разности фаз.

Следует заметить, что для синхронизации мы используем два символа в S_{12} и S_{34} , ранний и поздний. А для демодуляции и детектирования мы используем ранний символ (S_{12}) и предыдущий ему символ, который на прошлом шаге был в S_{12} . Обозначим действительную часть предыдущего символа Sp_{12_i} , а мнимую – Sp_{12_q} .

Найдем действительную и мнимую части фазового сдвига, воспользовавшись формулой деления комплексных чисел (2), сократив знаменатели:

$$\begin{aligned} tr_i &= S_{12_i} \cdot Sp_{12_i} + S_{12_q} \cdot Sp_{12_q}, \\ tr_q &= S_{12_q} \cdot Sp_{12_i} - S_{12_i} \cdot Sp_{12_q}. \end{aligned}$$

Для детектирования принятых бит используем следующие соотношения:

Если $tr_i \geq |tr_q|$, то значение битов 00.

Если $tr_i \leq -|tr_q|$, то значение битов 11.

Если $tr_q > |tr_i|$, то значение битов 01.

Иначе значение битов 10.

Все приведенные примеры соответствуют идеализированным условиям, либо отсутствию шума, либо большому отношению сигнал-шум.

При увеличении шума появляется возможность выхода из стационарных состояний, как из состояния рассинхронизации ($\text{Cог} = 4$), так и из состояния синхронизма.

Вероятности этих событий зависят от отношения сигнал – шум и значения порога подстройки синхронизации.

В реальных условиях алгоритм двигает окна синхронизации не на каждом шаге анализа, а после длительной интеграции результатов сравнения, накапливая их значения в аккумуляторе, до превышения порога.

Изменяя значения порогов, мы можем влиять на время вхождения в синхронизм и вероятность его сохранения.

6. Включение элементов адаптивности в алгоритм

В предыдущем разделе был указан такой параметр работы алгоритма как порог аккумулятора. Представляет интерес характер зависимости между этим параметром и временем вхождения в синхронизм и вероятностью его сохранения. Моделирование (более подробно описанное в следующем разделе) подтвердило интуитивно ожидаемый вывод о том, что, при заданном уровне шума, чем выше порог, тем больше время синхронизации, но, при этом, меньше вероятность выпасть из синхронизма. При этом, чем выше шум, тем выше вероятность выпасть из синхронизма. Так как улучшение одной характеристики алгоритма ухудшает другую, то, очевидно, вопрос выбора оптимального порога не может быть решён однозначно. Этот выбор будет определяться требованиями к системе связи, в которой алгоритм будет работать. То есть надо задаться максимально допустимым уровнем шума и требуемым временем вхождения или максимальной вероятностью выпада из синхронизма. И в этом видится преимущество алгоритмов, в которых есть элементы адаптивности. Если адаптивный алгоритм может оценить уровень шума и этот уровень шума будет меньше, чем максимально допустимый, то алгоритм может уменьшить порог аккумулятора, уменьшая длительность вхождения в синхронизм без риска увеличить вероятность срыва синхронизации. Остаётся только выбрать адекватный способ оценки уровня шума. Обзор возможных способов оценки шумов представлен в работе [4]. Принимая во внимание, что в нашем случае рассматриваются только типы модуляции, которые характеризуются постоянным уровнем амплитуды сигнала, то любые вариации этого уровня от символа к символу будут обусловлены только шумом. Учитывая, что данные амплитуды уже вычислены в процессе синхронизации, целесообразно использовать следующий алгоритм:

На каждом этапе, нужно сохранять амплитуду окон $S1$ и $S2$ в массив.

Так как $S1$ и $S2$ представляют из себя комплексные числа, амплитуду находим по следующим формулам:

$$A_k = \sqrt{S1_i^2 + S1_q^2},$$

$$A_{k+1} = \sqrt{S2_i^2 + S2_q^2},$$

где k – это размер массива на момент вычисления формулы.

Через определенное время (опускаем конкретное значение, оно будет определено дальше), проводим исследование. Сортируем массив и отбрасываем половину с наименьшими значениями амплитуд. Это делается для отбрасывания окон с границами символов. На границе символов семплы разных символов уменьшают общее значение амплитуды, таким образом их можно найти и удалить. После удаления размер массива k уменьшится вдвое

$$k = \frac{k}{2}.$$

Теперь находим среднее в этом массиве – суммируем все элементы и делим на их количество:

$$Avg = \frac{\sum_{i=0}^k A_i}{k}.$$

На данном этапе была получена средняя амплитуда, которую вычитаем из всех оставшихся элементов в массиве.

$$B_k = A_k - Avg,$$

B – массив с отклонением от Avg .

Далее оцениваем дисперсию шума и нормируем её:

$$Disp_{abs} = \sqrt{\frac{\sum_{i=0}^k (B_i)^2}{k}},$$

$$Disp_{norm} = \frac{Disp_{abs}}{Avg},$$

где $Disp_{abs}$ и $Disp_{norm}$ – дисперсии абсолютная и нормированная.

7. Заключение, вывод результатов моделирования

В результате моделирования DQPSK и алгоритма синхронизации, мы можем получить статистику, которая характеризует состояние системы во время работы.

Основными вероятностно-временными показателями данной статистики являются, вероятность ошибок (Per Error), вероятность нахождения в состоянии синхронизма (Probability Syn), время нахождения в состояниях рассинхронизации (Ind Live).

Параметры, по которым мы получаем показатели: соотношение сигнал-шум (E_b/N_0), состояния рассинхронизации на заданное количество семплов (Cor), порог аккумулятора (por).

По итогу мы можем увидеть, что в зависимости от величины порога изменяются вероятность нахождения в синхронизме и время нахождения в рассинхронизации. Значения порогов на рисунках приведено в миллионах ($1m = 1\,000\,000$). Чем больше порог, тем дольше мы будем прибывать в каждом состоянии – дольше будем идти в синхронизм, но дольше задержимся в синхронизме. И наоборот.

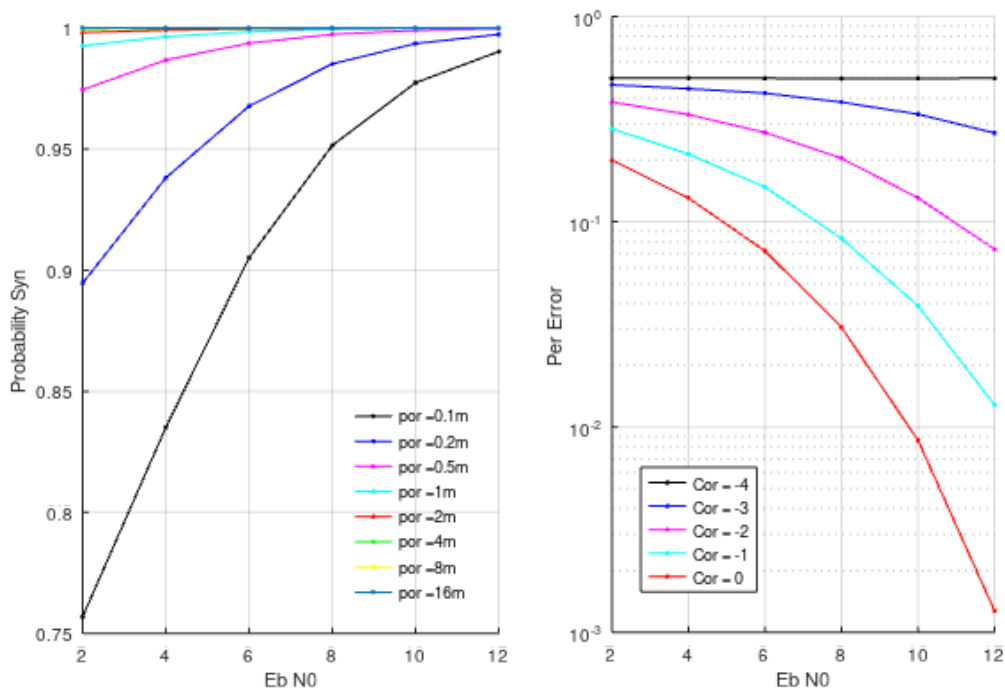


Рис. 5. Вероятность нахождения в синхронизме от отношения E_b/N_0 при разных порогах, вероятность ошибок в разных состояниях от отношения E_b/N_0

Также стоит отметить, что для каждого состояния мы имеем свои вероятности ошибок, где при полной рассинхронизации ($\text{Cог} = -4$) мы имеем вероятность ошибок 50%. При рассинхронизации всего на один семпл вероятность ошибок увеличивается на порядок по сравнению с $\text{Cог} = 0$.

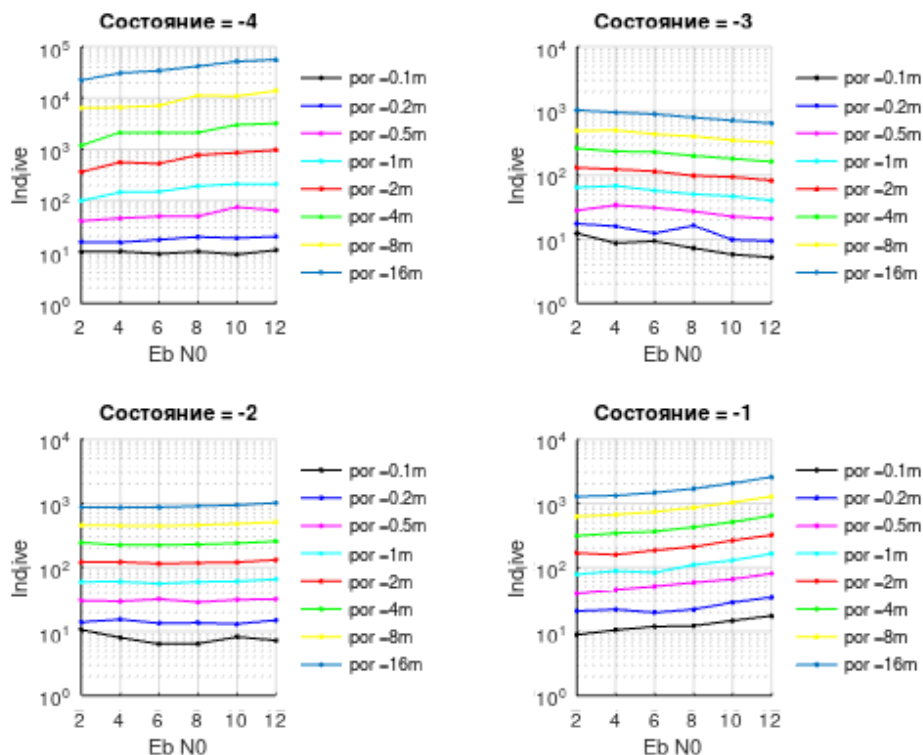


Рис. 6. Время нахождения в разных состояниях в зависимости от por и от отношения E_b/N_0

Из графиков рис. 6 видно, что при $\text{cог} = 4$ семпла, время нахождения в рассинхронизации может превышать аналогичное время при старте из других состояний на порядок и более. В данном случае целесообразно использовать оценку дисперсии и адаптивное понижение порога для уменьшения времени вхождения в синхронизм, с последующим повышением после шага подстройки. Возможно, так же провести подстройку сразу на 2 или 4 семпла.

Литература

1. *Бернард С.* Цифровая связь. Теоретические основы и практическое применение. 2003
2. *Mohamed K. Nezami* RF Architectures and Digital Signal Processing Aspects of Digital Wireless Transceivers, 2004
3. *Мелентьев О. Г., Максимов И. Н.* Алгоритм поэлементной синхронизации по отсчетам комплексной огибающей без использования ФАПЧ. Вестник СибГУТИ. 2021. №4 С.87-94.
4. *Мелентьев О. Г., Шевнина И. Е.* Модификация алгоритма оценки отношения сигнал/шум. Вестник СибГУТИ 2018. № 2, С.3-7.

Мелентьев Олег Геннадьевич

д.т.н., профессор кафедры инфокоммуникационных систем и сетей, Сибирский государственный университет телекоммуникаций и информатики (СибГУТИ, 630102, Новосибирск, ул. Кирова, д. 86), тел. (383) 2-698-240, e-mail: melog.aes@gmail.com, ORCID ID: 0000-0003-1646-6844.

Максимов Иван Николаевич

ведущий инженер кафедры инфокоммуникационных систем и сетей, Сибирский государственный университет телекоммуникаций и информатики (СибГУТИ, 630102, Новосибирск, ул. Кирова, д. 86), тел. (383) 2-698-240, e-mail: ivan.maximov@mail.ru, ORCID ID: 0009-0006-5814-9352

Гончаров Даниил Андреевич

ассистент кафедры инфокоммуникационных систем и сетей, Сибирский государственный университет телекоммуникаций и информатики (СибГУТИ, 630102, Новосибирск, ул. Кирова, д. 86), тел. 8-923-223-9630, e-mail: denn102@bk.ru, ORCID ID: 0009-0002-7869-1203

Авторы прочитали и одобрили окончательный вариант рукописи.

Авторы заявляют об отсутствии конфликта интересов.

Вклад соавторов: каждый автор внес равную долю участия как во все этапы проводимого теоретического исследования, так и при написании разделов данной статьи.

The algorithm of symbolic synchronization based on complex envelope counts with adaptivity elements for DQPSK modulation

D. Goncharov, I. Maximov, O. Melentyev
Siberian State University of Telecommunications and Informatics

Abstract: An algorithm for symbol synchronization based on complex envelope samples for DQPSK modulation is proposed. The algorithm does not use phase locked loop (PLL). The simulation of this algorithm is carried out, the average time of entering into synchronism and the probability of errors are determined.

Keywords: symbol synchronization, error probability, synchronization time, synchronization error.

For citation: Goncharov D. A., Maximov I. N., Melentyev O. G. The algorithm of symbolic synchronization based on complex envelope counts with adaptivity elements for DQPSK modulation. Vestnik SibGUTI, 2024, vol. 18, no. 4, pp. 76-86.
<https://doi.org/10.55648/1998-6920-2024-18-4-76-86>.



Content is available under the license
Creative Commons Attribution 4.0
License

© Goncharov D. A., Maximov I. N.,
Melent'ev O. G., 2024

The article was submitted: 09.04.2024;
revised version: 03.06.2024;
accepted for publication 11.07.2024.

References

1. Bernard S. Digital communication. Theoretical foundations and practical application. 2003.
2. Mohamed K. Nezami RF Architectures and Digital Signal Processing Aspects of Digital Wireless Transceivers, 2004.
3. Melentyev O. G., Maksimov I. N. Algorithm of element-by-element synchronization based on complex envelope counts without using PLL. Vestnik SibGUTI. 2021. no. 4, pp. 87-94.
4. Melentyev O. G., Shevnina I. E. Modification of the algorithm for estimating the signal-to-noise ratio. Vestnik SibGUTI. 2018. no. 2, pp. 3-7.

Oleg G. Melentyev

Doctor of Technical Sciences, Professor of the Department of Infocommunication Systems and Networks, Siberian State University of Telecommunications and Informatics (SibSUTIS, 630102, Novosibirsk, Kirova St. 86), phone: (383) 2-698-240, e-mail: melog.aes@gmail.com, ORCID ID: 0000-0003-1646-6844.

Ivan N. Maximov

Leading Engineer of the Department of Infocommunication Systems and Networks, Siberian State University of Telecommunications and Informatics (SibSUTIS, 630102, Novosibirsk, Kirova St. 86), phone (383) 2-698-240, e-mail: ivan.maximov@mail.ru, ORCID ID: 0009-0006-5814-9352

Daniil A. Goncharov

Assistant of the Department of Infocommunication Systems and Networks, Siberian State University of Telecommunications and Informatics (SibSUTIS, 630102, Novosibirsk, Kirova St. 86), phone 8-923-223-9630, e-mail: denn102@bk.ru, ORCID ID: 0009-0002-7869-1203