

# Алгоритм поэлементной синхронизации по отсчетам комплексной огибающей без использования ФАПЧ

О. Г. Мелентьев, И. Н. Максимов

Предложен алгоритм поэлементной синхронизации по отсчетам комплексной огибающей. Проведено имитационное моделирование и сравнение двух алгоритмов, не использующих ФАПЧ, по вероятности ошибок для разных значений отношения энергии бита к спектральной плотности мощности шума.

*Ключевые слова:* поэлементная синхронизация, тактовая синхронизация, алгоритм синхронизации.

## 1. Введение

Поэлементная синхронизация определяет границы единичных элементов в потоке отсчетов принимаемого сигнала и является обязательной частью любой цифровой системы передачи. Общеизвестные алгоритмы поэлементной синхронизации рассмотрены в работах [1–3].

Для упрощения анализа рассмотрим передачу знакопеременного сигнала прямоугольной формы (меандра), представленного на рис. 1. Данный сигнал поступает на фазовый модулятор и передается в эфир.

Если на приеме обеспечить синхронизацию фазы, то в отсутствие шумов после переноса сигнала на нулевую частоту и фильтрации паразитных высокочастотных составляющих мы получим прямоугольный действительный сигнал. После чего производится его дискретизация. Выберем частоту дискретизации такой, что на единичном интервале будем брать по восемь отсчетов сигнала  $n=8$ . Положим, что все отсчеты нормированы и имеют значения  $\pm 1$ , что удобно для рассмотрения числовых примеров. Для простоты пока положим, что моменты взятия отсчетов в приемнике и передатчике синхронны. В таком случае сигнал будет иметь вид, как на рис. 1.

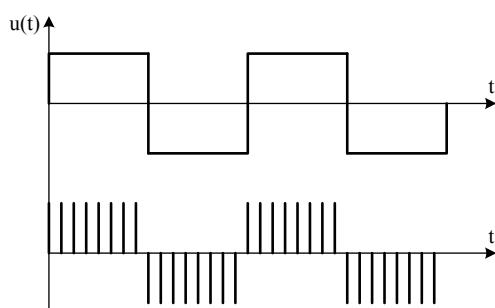


Рис. 1. Прямоугольный действительный сигнал и его отсчеты

## 2. Алгоритм поэлементной синхронизации по действительным отсчетам

Рассмотрим известный алгоритм поэлементной синхронизации по действительным отсчетам сигнала. Алгоритм базируется на анализе сумм отсчетов в четырех последовательно расположенных окнах размером  $n/2 = 4$  элемента. Задача алгоритма – совместить начало первого окна с первым семплом на единичном интервале. Обозначим последовательные окна и суммы отсчетов в них  $S_1, S_2, S_3, S_4$ , а суммы отсчетов в двух последовательных окнах –  $S_{12}, S_{23}$  и  $S_{34}$  соответственно.

Знак суммы  $S_{12}$  определяет значение детектируемого элемента, а знак  $S_{34}$  – значение следующего элемента. Анализ проводится, только когда значения элементов на соседних единичных интервалах меняют знак. При равенстве знаков  $S_{12}$  и  $S_{34}$  алгоритм пропускает шаг подстройки синхронизации.

Алгоритм анализирует области  $S_2$  и  $S_3$ , определяет, где происходит смена значащей позиции, и формирует решение на смещение окна в эту сторону. Последовательности отсчетов и окна синхронизации для двух последовательных шагов анализа алгоритма при рассинхронизации на два периода дискретизации показаны на рис. 2. Так как суммы  $S_{12}$  и  $S_{34}$  имеют разные знаки, то анализ проводится. На рис. 2а  $S_{23}$  равна четырем, что больше нуля, значит, в области анализа преобладает единица, при этом значащий момент находится в области с меньшим значением суммы отсчетов в  $S_3$ .

На следующем шаге смещаем окно синхронизации на восемь отсчетов (рис. 2б). Здесь сумма  $S_{23} = -4$  меньше нуля, значит, в области анализа преобладает минус единица (логический ноль), при этом значащий момент находится в области с большим значением суммы отсчетов, т.е. опять в  $S_3$ .

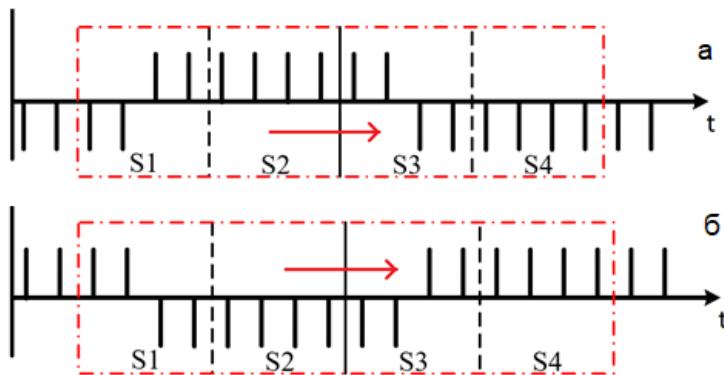


Рис. 2. Рассинхронизация на два периода дискретизации

Область  $S_2$  находится раньше, смещение в эту сторону назовем отрицательным ( $i - cor$ ). Смещение в сторону  $S_3$  – положительное ( $i + cor$ ).

На каждом рабочем цикле алгоритм формирует число по формуле (1), знак которого определяет направление будущего смещения, а величина характеризует степень рассинхронизации между текущим положением окна и предполагаемым.

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

где  $sign(S_{12})$  – функция взятия знака.

Для примера, рассмотренного выше (рис. 2б),  $Sm = sign(-4) \cdot -4 = 4$ .

Эти числа складываются между собой до тех пор, пока сумма (аккумулятор) не превысит по модулю некоторого порога. Если модуль аккумулятора превышает порог, то окно смещается на один отсчет в направлении, соответствующем знаку аккумулятора. Для рассматриваемого примера алгоритм сместит окна на один отсчет вправо, т.е. пропустит один отсчет, при этом рассинхронизация станет равна одному периоду дискретизации.

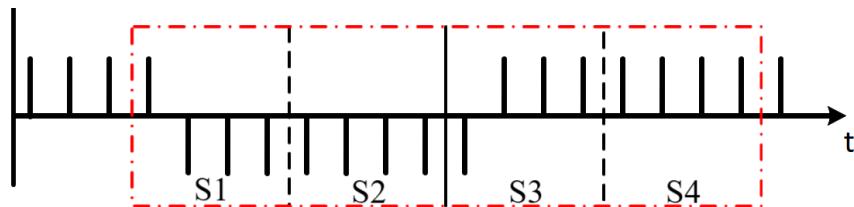


Рис. 3. Рассинхронизация на один период дискретизации

При смещении влево алгоритм вставит один отсчет с нулевым значением. Через некоторое число шагов система достигнет синхронизма, который будет поддерживаться алгоритмом, так как смещение в среднем будет нулевым:  $S_m = \text{sign}(8) \cdot 0 = 0$ . Это режим стационарного состояния (рис. 4).

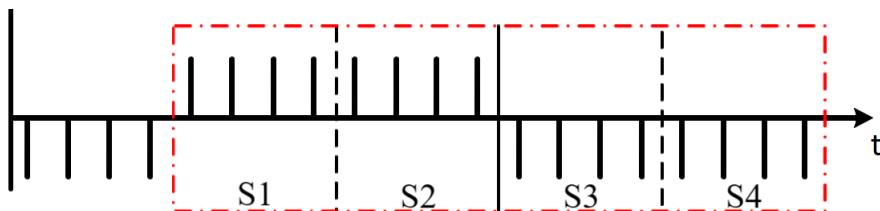


Рис. 4. Режим стационарного состояния

Как уже упоминалось, такая картина будет наблюдаться, если моменты взятия отчетов в приёмнике и передатчике синхронны. Теперь рассмотрим противоположный случай: моменты взятия отчетов в приёмнике и передатчике отстоят друг от друга на половину периода дискретизации. В таком случае сигнал будет иметь вид, как на рис. 5б. То есть момент взятия отсчета приходится точно на фронт импульса. В этом случае значение каждого восьмого отсчета становится близким к нулю.

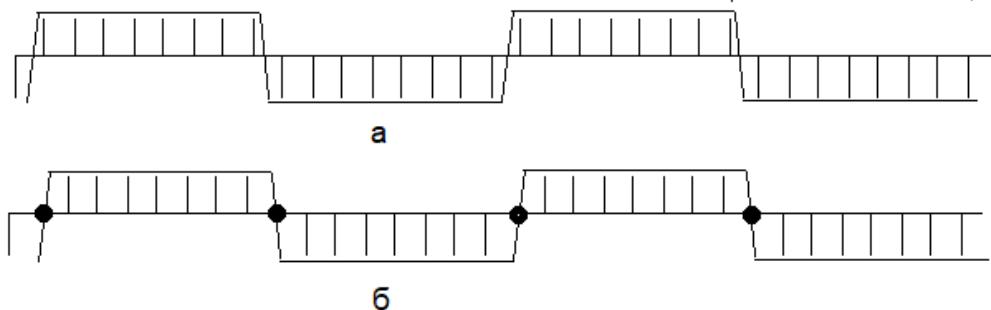


Рис. 5. Варианты взятия отсчетов

Периодические нули могут появляться и в результате фильтрации синхронных отсчетов. Например, при фильтрации последовательности, представленной на рис. 5а, фильтром скользящего среднего с прямоугольной импульсной характеристикой из восьми единичных отсчетов, который в данном случае является согласованным, каждый восьмой выходной отсчет будет нулевым.

Появление периодических нулей приводит к тому, что алгоритм синхронизации не сможет найти наилучшее положение окна и будет стремиться сдвинуть границу окна в сторону нуля.

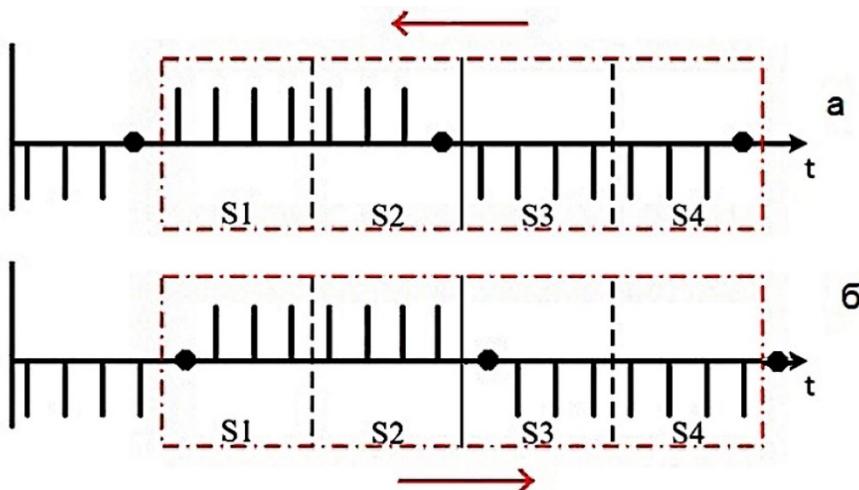


Рис. 6. Колебательный режим

На рис. 6а смещение  $Sm = sign(7) \cdot -1 = -1$  толкает окно влево, но когда окно сместится (смотри рис. 6б), смещение станет положительным  $Sm = sign(7) \cdot 1 = 1$  и после достижения порога окно сместится вправо. Возникает колебательный режим системы синхронизации.

### 3. Алгоритм поэлементной синхронизации по комплексным отсчетам

Рассмотренный выше алгоритм синхронизации по отсчетам действительного сигнала сравним с предлагаемым алгоритмом синхронизации по комплексным отсчетам.

В данном случае суммируем квадратуры восьми последовательных комплексных отсчетов комплексной огибающей с выхода квадратурного смесителя. В результате получаем действительные и мнимые части комплексных сумм:  $S12\_i, S12\_q; S23\_i, S23\_q; S34\_i, S34\_q$ . Теперь вместо проверки противоположности знаков  $S12$  и  $S34$  необходимо проверить разность фаз комплексных чисел  $S12$  и  $S34$ . Если разность фаз больше  $\pi/2$ , то можно производить анализ. В противном случае пропускаем текущий цикл синхронизации. Можно показать, что при разности фаз двух комплексных чисел более чем на  $\pi/2$  сумма произведений их действительных и мнимых частей будет меньше нуля. Вычисляем данную сумму:

$$tr = S12\_i \cdot S34\_i + S12\_q \cdot S34\_q. \quad (2)$$

Если результат отрицательный, то разность фаз более чем  $\pi/2$  и алгоритм проводит анализ.

Как известно, комплексное число может быть представлено точкой или вектором на комплексной плоскости. Идея алгоритма в векторном виде показана на рис. 7.

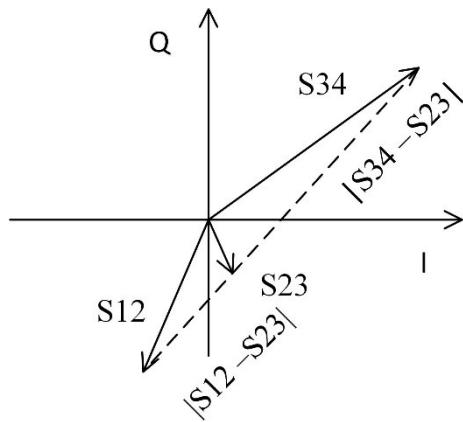


Рис. 7. Идея алгоритма синхронизации по комплексным отсчетам в векторном виде

Вычислим длину вектора  $S_{23}$ , или его модуль:

$$|S_{23}| = r = \sqrt{(S_{23\_i})^2 + (S_{23\_q})^2}. \quad (3)$$

Вычислим длины разности векторов  $(S_{12} - S_{23})$  и  $(S_{34} - S_{23})$ :

$$|S_{12} - S_{23}| = \sqrt{(S_{12\_i} - S_{23\_i})^2 + (S_{12\_q} - S_{23\_q})^2}, \quad (4)$$

$$|S_{34} - S_{23}| = \sqrt{(S_{34\_i} - S_{23\_i})^2 + (S_{34\_q} - S_{23\_q})^2}. \quad (5)$$

Разность усредненных векторов показывает, как много отсчетов с одинаковой фазой в них содержится. Чем меньше разность, тем больше отсчетов данного типа в  $S_{23}$ . Значит, надо смещать окно синхронизации в сторону соседнего элемента. Важно определить, какой из векторов приходит раньше. В нашем случае раньше приходит вектор  $S_{12}$ , поэтому если длина разности  $|S_{12} - S_{23}| < |S_{34} - S_{23}|$ , то векторов  $S_{12}$  в сумме  $S_{23}$  больше и по аналогии с действительным алгоритмом надо смещаться вправо, в сторону  $S_{34}$ , т.е. пропустить один отсчет.

Для уменьшения влияния помех на точность синхронизации вводим инерционный элемент – аккумулятор (*accum*), в который будем добавлять значение  $|S_{23}| = r$ , если преобладает  $S_{12}$ , или отнимать в противном случае. Для уменьшения вычислительной нагрузки целесообразно исключить взятие квадратного корня в выражениях (3), (4) и (5).

Далее выполняем сравнение модуля аккумулятора с порогом и производим смещение окна в соответствии со знаком аккумулятора. Если значение положительное, преобладает  $S_{12}$ , значит, пропускаем один отсчет, если отрицательное – добавляем один отсчет.

Добавление и вычитание отсчета производится изменением длины массива необработанных отсчетов на предыдущей операции чтения из буфера. Если надо пропустить отсчет и длина массива не нулевая – уменьшаем длину на один элемент. Если надо добавить отсчет и длина меньше максимальной, то искусственно увеличиваем количество считываемых элементов на один. После каждой корректировки обнуляем аккумулятор.

Детектор сравнивает разность фаз соседних усредненных комплексных значений векторов  $S_{12}$ , используя выражение (2), и принимает решение о значащей позиции. В комплексной системе не нужен отдельный детектор, так как фазы квадратур соседних символов вычисляются через каждые 8 отсчетов.

#### 4. Структурная схема моделирования

Для получения действительного сигнала необходимо использовать ФАПЧ или вычислять разность фаз отсчетов, сдвинутых на длину символа ( $n=8$ ). Так как в комплексном алгорит-

ме нам ФАПЧ не нужен, то представляется корректным для сравнения выбрать действительный алгоритм, основанный на вычислении разности фаз (DBPSK).

Передающая часть модели одинакова для обоих алгоритмов. Для обеспечения минимального времени вхождения в синхронизм значащие позиции символов должны меняться на каждом единичном элементе, поэтому комплексная огибающая должна быть меандром. Учитывая логику DBPSK-модуляции: «Если на вход поступает логическая единица, то фаза сигнала на текущем единичном интервале меняется на  $\pi$  относительно сигнала на предыдущем интервале. При поступлении нуля фаза не меняется», необходимо перекодировать исходную информационную последовательность перед подачей на квадратурный модулятор. Правило перекодирования следующее: если во входной последовательности  $X_i$  – единица, то в выходной  $Y_i$  меняем значащую позицию относительно предыдущего выходного элемента. При поступлении нуля значащая позиция не меняется. Данное преобразование можно легко реализовать, используя логическую операцию «исключающее ИЛИ»:  $Y_i = Y_{i-1} \oplus X_i$ . Учитывая преобразование, меандру на выходе будет соответствовать входная информационная последовательность «все единицы».

Далее производим биполярное преобразование – заменяем логические нули на минус единицы и увеличиваем частоту дискретизации в  $n=8$  раз.

Для имитации канала передачи генерируем нормальный комплексный шум с математическим ожиданием, равным 0, и среднеквадратическим отклонением, равным 1. Сгенерированные таким образом отсчеты шума умножаем на коэффициент, соответствующий отношению энергии бита к спектральной плотности мощности шума  $EbNo$  и количеству отсчетов на единичном интервале, и складываем с отсчетами DBPSK-сигнала:

$$IQn = I + \sqrt{n} 10^{\frac{EbNo}{20}} \cdot noise \quad (6)$$

Структурную схему приемника можно представить следующим образом

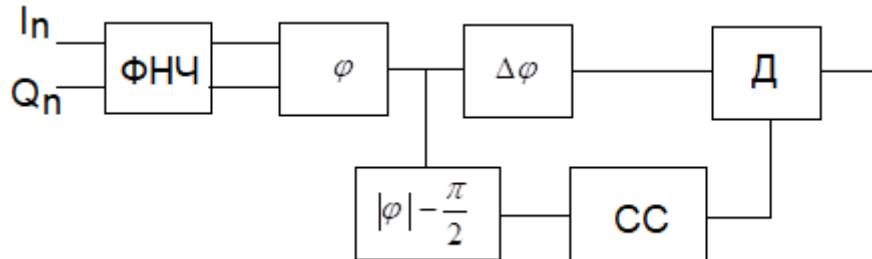


Рис. 8. Структурная схема модели приемника

Комплексный сигнал с шумом ( $I_n, Q_n$ ) поступает на фильтр скользящего окна низких частот (ФНЧ), импульсная характеристика которого имеет прямоугольную форму из  $n=8$  единичных отсчетов.

Далее для каждого фильтрованного комплексного отсчета вычисляем фазу как арктангенс отношения мнимой и действительной составляющих:

$$\phi_i = a \tan\left(\frac{Q_i}{I_i}\right).$$

Получаем последовательность действительных чисел в диапазоне от  $-\pi$  до  $\pi$ . Модули элементов данной последовательности, смещенные на  $-\pi/2$ , поступают в блок поэлементной синхронизации (СС).

$$F = |\phi| - \frac{\pi}{2}. \quad (7)$$

Так как мы используем DBPSK, находим разность фаз отсчетов, сдвинутых на длину символа ( $n=8$ ):

$$\Delta\phi = \phi_i - \phi_{i-n}. \quad (8)$$

При вычислении разности фаз возможны значения, превышающие по модулю  $\pi$ . В таких случаях проводим корректировку. Если  $|\Delta\phi| > \pi$ , то

$$\Delta\phi = \Delta\phi - \text{sign}(\Delta\phi) \cdot 2 \cdot \pi. \quad (9)$$

Скорректированный сигнал подается на детектор ( $\mathcal{D}$ ), который принимает решение, сравнивая модуль разности фаз с  $\pi/2$ . Если сдвиг фаз более  $\pi/2$ , то принимается решение о единице, иначе – ноль.

В модели приемника с алгоритмом синхронизации по комплексным отсчетам реализуется только рассмотренный выше алгоритм, который включает в себя все необходимые функции. Так как при реализации комплексного алгоритма выражения (3), (4) и (5) использовались без корня, то для выравнивания условий в системе синхронизации по действительным отсчетам выражение (2) было модифицировано:

$$Sm = \text{sign}(S12) \cdot \text{sign}(S23) \cdot S23^2. \quad (10)$$

## 5. Результаты моделирования

Сравнение алгоритмов производилось в диапазоне отношений  $EbNo$  от 2 до 12 дБ. Моделирование показало, что в колебательном режиме оба алгоритма имеют одинаковую вероятность ошибок, в стационарном режиме комплексный алгоритм обеспечивает меньшую вероятность ошибок.

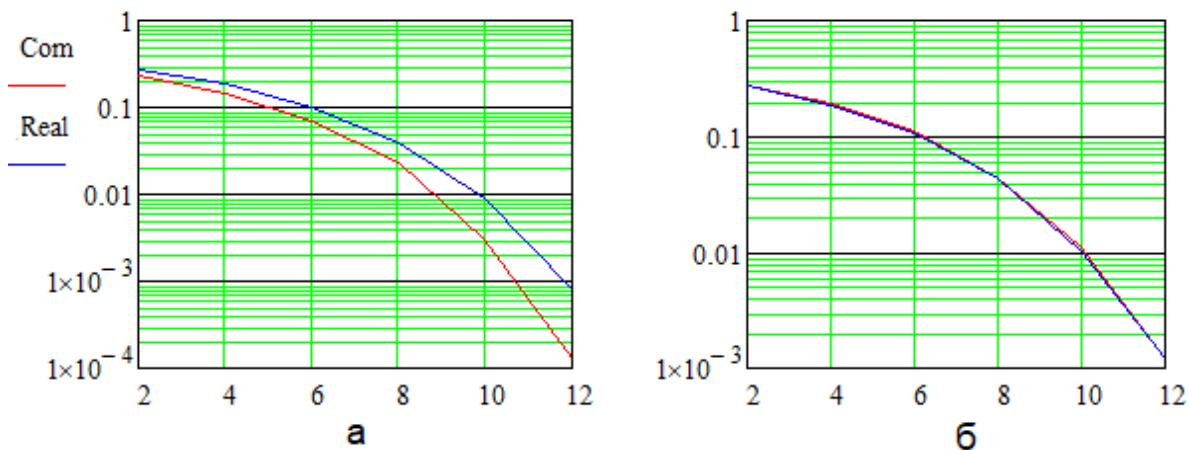


Рис. 9. Зависимости вероятности ошибок от отношения  $EbNo$ :  
а – в стационарном режиме, б – в колебательном режиме

Считая, что оба режима синхронизации равновероятны, усредним вероятности ошибок в разных режимах и проведем сравнение алгоритмов. Из рис. 10 видно, что выигрыш комплексного алгоритма растет с увеличением отношения  $EbNo$  от 5.7 % при 2 дБ и до 50 % при 12 дБ.



Рис. 10. Выигрыш комплексного алгоритма по вероятности ошибок в процентах

Кроме выигрыша по вероятности ошибок комплексный алгоритм показал значительно меньшее время вхождения в синхронизм (на порядок) и, как предполагалось, нечувствительность к фазовым сдвигам и незначительным частотным расстройкам.

## Литература

1. Галкин В. А. Цифровая мобильная радиосвязь: учебное пособие для вузов. М.: Горячая линия – Телеком, 2012. 592 с.
2. Склар Б. Цифровая связь. Теоретические основы и практическое применение: пер. с англ. М.: Издательский дом «Вильямс», 2003. 1104 с.
3. Чичварин Н. В. Системы синхронизации в каналах связи [Электронный ресурс]. [https://ru.bmstu.wiki/Системы\\_синхронизации\\_в\\_каналах\\_связи](https://ru.bmstu.wiki/Системы_синхронизации_в_каналах_связи) (дата обращения: 03.06.2016).

*Статья поступила в редакцию 20.11.2021;  
переработанный вариант – 09.12.2021.*

### Мелентьев Олег Геннадьевич

д.т.н., профессор кафедры инфокоммуникационных систем и сетей СибГУТИ (630102, Новосибирск, ул. Кирова, 86), тел. (383) 2-698-240, e-mail: melog.aes@gmail.com.

### Максимов Иван Николаевич

ведущий инженер кафедры инфокоммуникационных систем и сетей СибГУТИ, тел. (383) 2-698-240, e-mail: ivan.maximov@mail.ru.

## Symbol synchronization algorithm using complex envelop samples without PLL

### O. Melent'ev, I. Maximov

Symbol synchronization algorithm using complexes envelop samples without PLL was proposed. Two different symbol synchronization algorithms were compared. A dependency of the bit error rate on white gaussian noise is given.

*Keywords:* symbol synchronization, clock synchronization, synchronization algorithm.