

# Частотно-временная синхронизация

## 1 Введение

По эфиру передается модулированное моногармоническое колебание  $x(t)$  с несущей частотой  $f_c$  - косинусоида, амплитуда и фаза которой меняется во времени по определенному закону - в зависимости от передаваемых данных:

$$x(t) = A(t)\cos(2\pi f_c t + \phi(t)) \quad (1)$$

Амплитудный спектр сигнала  $x(t)$  схематично изображен на графике ниже:

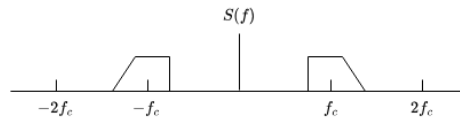


Рис. 1: Спектр полосового сигнала  $x(t)$

Сигнал  $x(t)$  вещественный, поэтому его амплитудный спектр симметричен относительно нулевой частоты. Кроме того, данный сигнал является полосовым - он занимает не все частоты, а лишь определенную *полосу* частот: от  $f_c - B/2$  до  $f_c + B/2$ , где  $B$  - полоса сигнала.

Программно-определяемый радиоприёмник производит запись передаваемого сигнала  $x(t)$  в его полосе. Данную операцию можно представить в виде следующей последовательности действий:

1. Перенос полезной полосы сигнала с несущей частоты на нулевую частоту путем умножения во временной области на комплексную экспоненту  $e^{-j2\pi f_c t}$
2. Фильтрация низкочастотным фильтром для удаления удвоенной отрицательной частоты
3. Оцифровка сигнала с частотой, достаточной для захвата всей полосы сигнала

Оцифровка сигнала состоит из квантования по амплитуде и дискретизации по времени. Дискретизация производится с определенной частотой, именуемой частотой дискретизации (англ.: sample rate).

Амплитудный спектр сигнала после переноса частоты, амплитудно-частотная характеристика (АЧХ) низкочастотного фильтра и амплитудный спектр сигнала после фильтрации схематично изображены на графиках ниже:

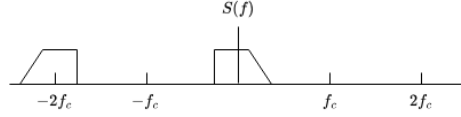


Рис. 2: Спектр полосового сигнала  $x(t)$  после переноса частоты

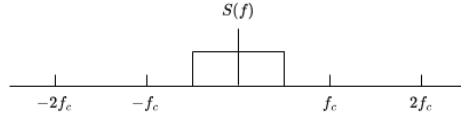


Рис. 3: АЧХ низкочастотного фильтра

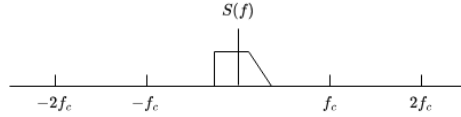


Рис. 4: Спектр сигнала после фильтрации

Отметим, что амплитудный спектр сигнала после фильтрации не является симметричным относительно нулевой частоты, то есть сигнал теперь является комплексным. Таким образом, каждый отсчет оцифрованного сигнала будет состоять из вещественной и мнимой части. Вещественная часть ещё называется синфазной (англ.: in-phase) или I-компонентой, а мнимая - квадратурной (англ.: quadrature) или Q-компонентой, поэтому комплексные отсчеты ещё называют IQ-отсчетами (англ.: IQ-samples). Подробнее про полосовые сигналы можно почитать в статье [2].

Рассмотрим также частный случай, когда полосовой сигнал  $x(t)$  является немодулированным моногармоническим колебанием с частотой, превышающей несущую частоту:  $x(t) = A \cos(2\pi(f_c + f_0)t)$ . После переноса с несущей частоты на нулевую с последующей фильтрацией такой сигнал будет представлять собой комплексную экспоненту с *положительной* частотой:  $e^{j2\pi f_0 t}$ . Другим частным случаем является ситуация, когда полосовой сигнал является немодулированным моногармоническим колебанием с частотой, меньшей несущей частоты:  $x(t) = A \cos(2\pi(f_c - f_0)t)$ . В таком случае после переноса с несущей частоты на нулевую с последующей фильтрацией такой сигнал будет представлять собой комплексную экспоненту с *отрицательной* частотой:  $e^{-j2\pi f_0 t}$ .

Также уточним, что после оцифровки с частотой дискретизации  $F_s$  непрерывная комплексная экспонента  $e^{j2\pi f_0 t}$  превратится в комплексную экспоненту дискретного времени  $e^{j2\pi \frac{f_0}{F_s} k}$ .

В реальности невозможно точно перенести полосовой сигнал с несущей частоты на нулевую частоту: всегда будет иметь место небольшая частотная расстройка. Именно поэтому необходимо выполнять частотную синхронизацию. Кроме того, время распространения сигнала между приемником и передатчиком, как правило, неизвестно, поэтому нужно также выполнять временную синхронизацию: определять, где начало сигнала, а где его конец.

Установление частотно-временной синхронизации обычно выполняют совместно. Основная идея заключается в том, что часть информации, которая передается модулированным сигналом, уже известна приемнику. Таким образом, приемник ищет в принятом сигнале известные ему паттерны и по ним определяет частотную расстройку, а также смещение по времени.

## 2 Симуляционная платформа

В рамках текущего задания кандидату предлагается реализовать один из алгоритмов частотно-временной синхронизации и промоделировать помехоустойчивость этого алгоритма в канале с аддитивным белым гауссовским шумом (АБГШ) методом Монте-Карло. Результатом моделирования помехоустойчивости является зависимость вероятности неуспешной синхронизации от отношения сигнал-шум (SNR).

В скрипте *timeFreqSyncTest.m* реализована симуляционная платформа, которая позволяет промоделировать помехоустойчивость упрощенного алгоритма синхронизации. Скрипт перебирает диапазон SNR, для каждого значения формирует модель принятого сигнала с текущим SNR, далее выполняет синхронизацию и определяет, была ли она выполнена успешно. Синхронизация считается успешной, если временная синхронизация установлена с точностью до 2 дискретных отсчета. Для каждого значения SNR набирается достаточная статистика экспериментов, после чего оценивается искомая вероятность неуспешной синхронизации.

При моделировании принятого сигнала операции, связанные с переносом частоты, фильтрацией и дискретизацией, опускаются. Вместо этого формируется модель комплексного сигнала уже после оцифровки, а влияние канала распространения и переноса частоты моделируется путем добавления шума и частотного сдвига.

Модель комплексного сигнала состоит из нескольких частей, каждая из которых представляет собой некий комплексный сигнал во временной области. В скрипте *timeFreqSyncTest.m* эти части называются *freqSeq*, *synchSeq* и *dataSeq*, а результирующий сигнал во временной области описывается конкатенацией этих трех частей:

```
txSignal = [freqSeq synchSeq dataSeq];
```

- *freqSeq* представляет собой комплексную экспоненту с частотой, заданной переменной *toneFreq*. Данная комплексная экспонента может быть использована для установления первоначальной грубой временной и частотной синхронизации.
- *synchFreq* представляет собой псевдослучайную последовательность, которая известна как на передающей стороне, так и на приемной. По корреляции с этой известной последовательностью приемник может выполнить точную временную синхронизацию.
- *dataSeq* представляет собой последовательность передаваемых данных. В текущей модели эта часть формируемого сигнала никак не используется.

Реализованный в скрипте *timeFreqSyncTest.m* упрощенный алгоритм синхронизации никак не использует *freqSeq*. Вместо этого сразу выполняется корреляция с *synchSeq*, после чего временная синхронизация выполняется путем нахождения максимума корреляции.

Величина частотной расстройки задается переменной *freqError*. При частотной расстройке в 1КГц упрощенный метод синхронизации работает достаточно хорошо. Зависимость вероятности неуспешной синхронизации от SNR приведена на графике ниже.

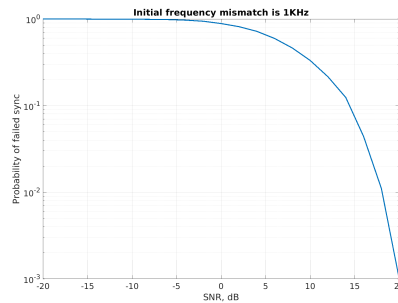


Рис. 5: Зависимость вероятности неуспешной синхронизации от SNR при частотной расстройке 1КГц

Как можем видеть из графика, при  $\text{SNR} > 10\text{дБ}$  вероятность неуспешной синхронизации составляет менее 10 %, что является неплохим результатом.

### 3 Постановка задачи

Ситуация с синхронизацией обстоит намного хуже при увеличении частотной расстройки: при частотной расстройке в 10КГц упрощенный метод синхронизации просто перестает работать - вероятность неуспешной синхронизации составляет 100% вне зависимости от SNR, как показано на графике

ниже. Причина этого заключается в том, что *synchSeq* просто не коррелируется при большой частотной расстройке.

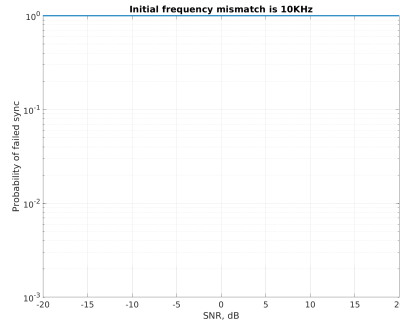


Рис. 6: Зависимость вероятности неуспешной синхронизации от SNR при частотной расстройке 10КГц

Для решения возникшей проблемы необходимо произвести доработку алгоритма. Перед тем, как выполнять корреляцию с *synchSeq*, необходимо выполнить грубую временную и частотную синхронизацию по *freqSeq*.

Дело в том, что *freqSeq* - это комплексная экспонента, следовательно, разность фаз между двумя её последовательными отсчетами постоянна. В то же время разность фаз между двумя последовательными отсчетами произвольного сигнала случайна.

Таким образом, можно предложить следующий алгоритм. Необходимо проходить по всем отсчетам сигнала и находить разность фаз между текущим отсчетом и предыдущим через умножение текущего отсчета на комплексное сопряжение предыдущего. Далее необходимо посчитать скользящее среднее от полученного сигнала, причем размер окна скользящего среднего должен равняться размеру *freqSeq*. При попадании на произвольный сигнал скользящее среднее выдаст число с маленькой амплитудой, а при попадании на комплексную экспоненту будет получен ярковыраженный пик. Пример такой ситуации изображен на графике ниже. Подробнее о данном методе можно почитать в статье [1].

Положение пика позволяет установить грубую временную синхронизацию, после чего можно грубо оценить частотную расстройку. Для этого достаточно извлекать из принятого сигнала участок, отвечающий за *freqSeq* (грубо временная синхронизация уже установлена), взять от этого участка преобразование Фурье и посмотреть, насколько сдвинулась частота этой комплексной экспоненты - это и даст нам грубую оценку частотной расстройки. Далее частотную расстройку можно компенсировать путем умножения принятого сигнала на комплексную экспоненту с нужной частотой, в результате чего *synchSeq* снова будет коррелироваться - дальнейшую временную синхронизацию можно оставить как в упрощенном алгоритме.

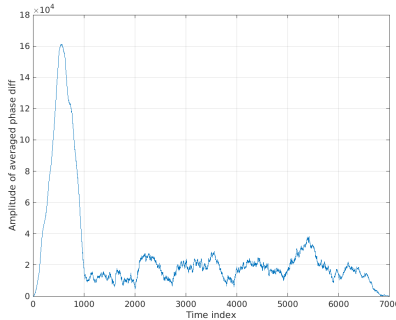


Рис. 7: Пример пика при детектировании комплексной экспоненты

## 4 Требования к выполнению

Таким образом, тестовое задание заключается в том, чтобы доработать алгоритм синхронизации в скрипте *timeFreqSyncTest.m* так, чтобы при достаточно большом SNR синхронизация выполнялась успешно даже для больших частотных расстройек, например, 10КГц.

К решению предъявляются следующие требования:

- Изменения необходимо вносить в приложенный скрипт *timeFreqSyncTest.m*
- Код необходимо оформлять в соответствии с тем, как он оформлен в скрипте *timeFreqSyncTest.m* - качество оформления также будет оцениваться
- Будьте готовы объяснить происходящее в вашем коде во время дальнейшего этапа интервью
- Написание небольшого отчета приветствуется, но не является обязательным

## Список литературы

- [1] Harald Kröll и др. “Low-complexity frequency synchronization for GSM systems: Algorithms and implementation”. В: *2012 IV International Congress on Ultra Modern Telecommunications and Control Systems*. IEEE. 2012, с. 168—173.
- [2] Бахурин Сергей. *Полосовые радиосигналы. Комплексная огибающая и универсальный квадратурный модулятор*. [https://ru.dsplib.org/content/quadrature\\_modulator/quadrature\\_modulator.html](https://ru.dsplib.org/content/quadrature_modulator/quadrature_modulator.html) [Accessed: 09.10.24]. 2015.