

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ
ИМ. ПРОФ. М. А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)**

Г.А. Фокин

**ФИЗИЧЕСКИЕ ОСНОВЫ
СИСТЕМ МОБИЛЬНОЙ СВЯЗИ**

УЧЕБНОЕ ПОСОБИЕ

СПб ГУТ)))

**САНКТ-ПЕТЕРБУРГ
2020**

УДК 621.396.93.001.2(076.5)
ББК 3884.16я 73
Б 12

Рецензенты:

доктор технических наук, директор института физики, нанотехнологий и телекоммуникаций СПбПУ, заведующий кафедры «Радиоэлектронные средства защиты информации» Санкт-Петербургского политехнического университета Петра Великого
С. Б. Макаров

доктор технических наук,
почетный профессор Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А.Бонч-Бруевича
М. А. Сиверс

Утверждено редакционно-издательским советом СПбГУТ в качестве учебного пособия

Г.А. Фокин

Б 12 Физические основы систем мобильной связи: учебное пособие / Г.А. Фокин; СПбГУТ. – СПб., 2020. – 300 с.

Даны методические рекомендации и материалы к лекционным, лабораторным и практическим занятиям по дисциплине «Физические основы систем мобильной связи».

Предназначено для бакалавров очной формы обучения по направлению 11.03.02 «Инфокоммуникационные технологии и системы связи».

УДК 621.396.93.001.2(076.5)
ББК 3884.16я 73

© Фокин Г. А., 2020

© Федеральное государственное образовательное бюджетное учреждение высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», 2020

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
1. МНОГОЛУЧЕВОЙ РАДИОКАНАЛ СИСТЕМ МОБИЛЬНОЙ СВЯЗИ.....	5
1.1. ЛЕКЦИЯ. МНОГОЛУЧЕВОЙ РАДИОКАНАЛ СИСТЕМ МОБИЛЬНОЙ СВЯЗИ	5
1.1.1. ВВЕДЕНИЕ.....	5
1.1.2. ПАРАМЕТРЫ ПРОФИЛЕЙ ЗАДЕРЖКИ.....	7
1.1.3. ПАРАМЕТРЫ НАПРАВЛЕНИЯ ПРИХОДА.....	12
1.1.4. ПАРАМЕТРЫ ПРОФИЛЯ ЗАДЕРЖКИ ПО УГЛУ МОЩНОСТИ.....	19
1.1.5. ПАРАМЕТРЫ ОТКЛОНЕНИЙ ПРИНЯТОГО СИГНАЛА.....	20
1.1.6. ПРОСТРАНСТВЕННАЯ КОРРЕЛЯЦИЯ	22
1.1.7. РАЗРЕШЕНИЕ КОМПОНЕНТОВ МНОГОЛУЧЕВОСТИ	25
1.2. ЛАБОРАТОРНАЯ РАБОТА. ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ РАДИОКАНАЛА С ЗАМИРАНИЯМИ	28
2. МЕЖСИМВОЛЬНАЯ ИНТЕРФЕРЕНЦИЯ И ФОРМИРУЮЩИЕ ФИЛЬТРЫ.....	33
2.1. ЛЕКЦИЯ. МЕЖСИМВОЛЬНАЯ ИНТЕРФЕРЕНЦИЯ И ФОРМИРУЮЩИЕ ФИЛЬТРЫ НАЙКВИСТА.....	33
2.2. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ. РАЗРАБОТКА ФОРМИРУЮЩЕГО ФИЛЬТРА НАЙКВИСТА	33
2.3. ЛАБОРАТОРНАЯ РАБОТА. ОЦЕНКА ФУНКЦИОНИРОВАНИЯ ФОРМИРУЮЩЕГО ФИЛЬТРА НАЙКВИСТА.....	33
3. ЦИФРОВЫЕ СИГНАЛЫ ФАЗОВОЙ МАНИПУЛЯЦИИ.....	35
3.1. ЛЕКЦИЯ. ЦИФРОВЫЕ СИГНАЛЫ ФАЗОВОЙ МАНИПУЛЯЦИИ	35
3.2. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ. РАЗРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ ДЛЯ ОЦЕНКИ ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ BPSK, QPSK, DBPSK, DQPSK, OQPSK, QAM В КАНАЛЕ С АБГШ И ЗАМИРАНИЯМИ	35
3.3. ЛАБОРАТОРНАЯ РАБОТА. ОЦЕНКА ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ BPSK, QPSK, DBPSK, DQPSK, OQPSK, QAM В КАНАЛЕ С ЗАМИРАНИЯМИ.....	38
3.3.1. ОЦЕНКА ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ BPSK.....	38
3.3.2. ОЦЕНКА ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ QPSK	42
3.3.3. ОЦЕНКА ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ OQPSK.....	45
3.3.4. ОЦЕНКА ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ QAM	50
4. ЦИФРОВЫЕ СИГНАЛЫ ЧАСТОТНОЙ МАНИПУЛЯЦИИ.....	55
4.1. ЛЕКЦИЯ. ЦИФРОВЫЕ СИГНАЛЫ ЧАСТОТНОЙ МАНИПУЛЯЦИИ	55
4.2. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ. РАЗРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ ДЛЯ ОЦЕНКИ ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ FMSK, GMSK В КАНАЛЕ С АБГШ И ЗАМИРАНИЯМИ ..	55
4.3. ЛАБОРАТОРНАЯ РАБОТА. ОЦЕНКА ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ FMSK, GMSK В КАНАЛЕ С ЗАМИРАНИЯМИ	56
4.3.1. ОЦЕНКА ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ FMSK.....	56
4.3.2. ОЦЕНКА ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ GMSK	61
5. СИГНАЛЫ С РАСШИРЕННЫМ СПЕКТРОМ	67
5.1. ЛЕКЦИЯ. СИГНАЛЫ С РАСШИРЕННЫМ СПЕКТРОМ	67
5.2. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ. РАЗРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ ДЛЯ ОЦЕНКИ ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ DSSS В КАНАЛЕ С АБГШ И ЗАМИРАНИЯМИ	67
5.3. ЛАБОРАТОРНАЯ РАБОТА. ОЦЕНКА ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ DSSS В КАНАЛЕ С ЗАМИРАНИЯМИ	69
6. СИГНАЛЫ С ОРТОГОНАЛЬНЫМ ЧАСТОТНЫМ РАЗДЕЛЕНИЕМ	72
6.1. ЛЕКЦИЯ. СИГНАЛЫ С ОРТОГОНАЛЬНЫМ ЧАСТОТНЫМ РАЗДЕЛЕНИЕМ	72
6.2. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ. РАЗРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ ДЛЯ ОЦЕНКИ ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ OFDM В КАНАЛЕ С АБГШ И ЗАМИРАНИЯМИ	72
6.3. ЛАБОРАТОРНАЯ РАБОТА. ОЦЕНКА ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ OFDM В КАНАЛЕ С ЗАМИРАНИЯМИ	73
ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ.....	79
СПИСОК ИСТОЧНИКОВ	80

ВВЕДЕНИЕ

Учебное пособие предназначено для изучения дисциплины «Физические основы систем мобильной связи» в рамках профессионального цикла с общей трудоемкостью дисциплины 6 зачетных единиц, 216 часов. Задачами учебного пособия «Физические основы систем мобильной связи» является изучение физических основ функционирования систем мобильной связи.

Учебное пособие «Физические основы систем мобильной связи» подготовлено в соответствии с новой рабочей программой дисциплины «Физические основы систем мобильной связи» [1] на основе требований Федерального государственного образовательного стандарта высшего профессионального образования по направлению подготовки 11.03.02 «Инфокоммуникационные технологии и системы связи» профиля «Системы мобильной связи».

В учебном пособии «Физические основы систем мобильной связи» представлены рекомендации и материалы к лекционным, лабораторным и практическим занятиям по дисциплине «Физические основы систем мобильной связи» в соответствии с тематическим планом занятий и количеством аудиторных часов 82. Материалы лекционных, практических и лабораторных занятий основаны на многолетнем опыте преподавания дисциплины на кафедре радиосвязи и вещания и материалах работ [2] – [6].

В результате освоения дисциплины студент должен

1. МНОГОЛУЧЕВОЙ РАДИОКАНАЛ СИСТЕМ МОБИЛЬНОЙ СВЯЗИ

1.1. ЛЕКЦИЯ. МНОГОЛУЧЕВОЙ РАДИОКАНАЛ СИСТЕМ МОБИЛЬНОЙ СВЯЗИ

Цель занятия: изучить особенности моделей распространения радиоволн (РРВ) в многолучевом радиоканале систем мобильной связи (СМС).

1.1.1. ВВЕДЕНИЕ

В Рекомендации МСЭ-R P.1407 [1] приводится описание характера многолучевого распространения и определяются надлежащие параметры для статистического описания явлений многолучевого распространения, а также приводятся примеры явлений корреляции между большим числом трасс распространения лучей и способов их расчета.

В системах мобильной связи, помимо прямых трасс распространения в пределах прямой видимости между передатчиком и приемником, часто возникают многолучевые не прямые трассы, обусловленные отражением от окружающих объектов. Такое многолучевое распространение особенно значительно в городских условиях, где стены зданий и дорожное покрытие обладают высокой отражательной способностью. В результате принимаемый сигнал состоит из нескольких компонентов с разными амплитудами, фазами и направлениями прихода.

Результирующая пространственная изменчивость напряженности поля сигнала может быть представлена в двух следующих режимах:

а) быстрое замирание, скорость которого меняется на расстоянии порядка длины волны, что обусловлено в первую очередь изменением фазы различных компонентов сигнала;

б) медленное замирание, которое меняется на расстояниях большой протяженности, прежде всего за счет изменения уровня потерь из-за затенения окружающими объектами.

Помимо этого, различные компоненты сигнала могут испытывать доплеровский сдвиг частоты различной величины, что обусловлено движением подвижной станции или отражающих объектов, таких как транспортные средства.

Многолучевой канал подвижной связи можно описать с помощью импульсной характеристики, скорость изменения которой зависит от скорости движения подвижной станции и/или рассеивателей. Следовательно, в приемнике должна быть предусмотрена возможность справляться с искажениями, возникающими за счет отраженных сигналов, а также учитывать быстрые изменения характера таких искажений. Подобные характеристики радиоканала подвижной связи описываются с помощью профилей задержки

мощности и доплеровскими спектрами, которые можно получить путем измерений зондирования широкополосного канала.

Из-за многолучевого рассеяния амплитуды сигналов, передаваемых на движущиеся транспортные средства и от них в городе или лесной местности, сильно меняются. К обычным явлениям относятся замирания на 30 дБ или более относительно среднего уровня. Мгновенное значение напряженности поля, измеренного на расстояниях порядка нескольких десятков длин волны, приблизительно следует рэлеевскому распределению. Средние значения для распределений на этих небольших секторах сильно меняются в зависимости от географической зоны, высоты, плотности и расположения холмов, деревьев, зданий и других объектов.

С физической точки зрения параметрами многолучевого распространения являются количество лучей, амплитуда, разница в длине трасс (задержка), доплеровский сдвиг и угол прихода. Значения этих параметров могут быть получены из серии комплексных импульсных характеристик на небольших расстояниях или временных интервалах, которые могут использоваться для расчета функции распределения доплеровской задержки, описывающей явление многолучевости в трехмерном пространстве – дополнительная задержка, доплеровская частота и плотность мощности. Функция распределения доплеровской задержки определяет линейный трансверсальный фильтр, выходной сигнал которого представляет собой сумму ослабленных, с многочисленными задержками и доплеровским сдвигом реплик входного сигнала. Такое представление удобно при аппаратном моделировании в виде динамического трансверсального фильтра. Функция распределения доплеровской задержки используется для расчета профиля задержки и спектра Доплера, который может быть связан с периодом когерентности канала. Кроме того, применив преобразование Фурье к переменной во времени комплексной импульсной характеристике, можно получить переменную во времени комплексную частотную функцию, амплитуда которой как функция частотных характеристик определяет многолучевую избирательность по частоте, которая связана с шириной полосы корреляции. Изменение во времени этой функции определяет характеристики замирания на конкретной частоте.

Определения параметров канала для небольшого измеряемого сектора (или мелкомасштабных параметров) приведены далее. Далее статистические данные о мелкомасштабных параметрах используются для составления кумулятивных функций распределения (CDF). Среднемасштабная функция CDF охватывает отдельный маршрут измерений, имеющий длину порядка от десятков до сотен метров. Совокупность объединенных данных от ряда среднемасштабных маршрутов рассматривается в качестве крупномасштабной или глобальной характеристики, представляющей особенности изучаемой окружающей обстановки, например, холмистой местности, города, пригорода, больших помещений внутри зданий, коридоров и т. д.

1.1.2. ПАРАМЕТРЫ ПРОФИЛЕЙ ЗАДЕРЖКИ

Определения профилей задержки мощности. Соответствующие параметры для статистического описания времени задержки многолучевости могут быть вычислены на основе любого из трех типов профилей задержки мощности: мгновенного профиля задержки мощности; кратковременного профиля задержки мощности; или долговременного профиля задержки мощности, которые представляют собой либо усредненные по времени значения, полученные при использовании стационарного приемника, отображающие происходящие в окружающей среде перемещения, либо усредненные пространственные значения, полученные при использовании приемника, находящегося в движении. Определения профилей задержки мощности получают, как показано на рис. 1.1.

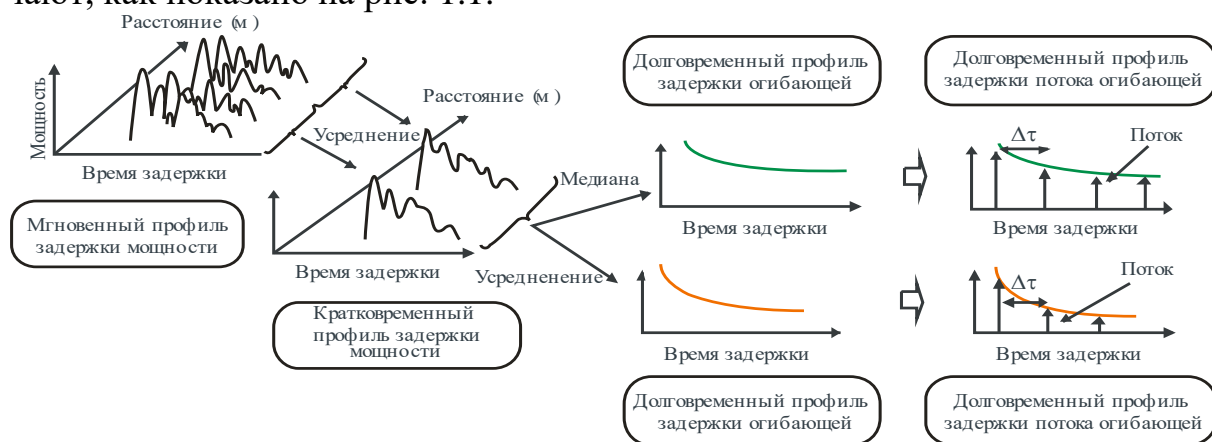


Рис. 1.1. Определение профилей задержки мощности

Мгновенный профиль задержки мощности представляет собой плотность мощности импульсной характеристики в один момент в одной точке.

Кратковременный (мелкомасштабный) профиль задержки мощности получают путем пространственного усреднения мгновенных профилей задержки мощности по нескольким десяткам значений длины волн в пределах диапазона, в котором сохраняются одни и те же компоненты многолучевости, в целях подавления колебаний быстрого замирания. Кроме того, он может быть получен из функции распределения доплеровской задержки, изображенной на рис. 1.2, путем суммирования квадратов абсолютных значений по оси доплеровского сдвига частоты, как показано на рис. 1.3.

Долговременный профиль задержки мощности получают путем пространственного усреднения кратковременных профилей задержки мощности приблизительно на одном и том же расстоянии от базовой станции (БС) в целях подавления колебаний, вызванных замиранием.

Долговременные профили задержки мощности с дискретным избыточным временем задержки, нормированные по временному разрешению $1/B$, где B – ширина полосы, определяются как долговременные профили задержки потока мощности вместо постоянных профилей задержки мощности.

С другой стороны, долговременный профиль задержки огибающей представляет собой срединное значение кратковременных профилей задержки мощности приблизительно на одном и том же расстоянии от базовой станции; он отражает форму профиля задержки в рассматриваемой зоне.

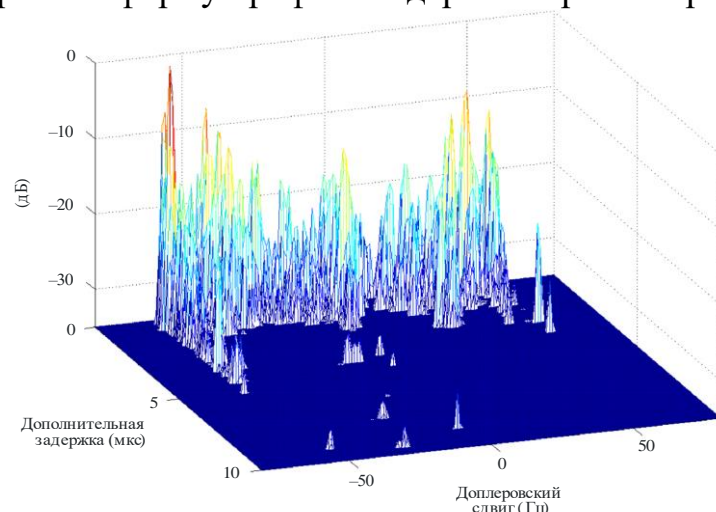


Рис. 1.2. Функция распространения доплеровской задержки

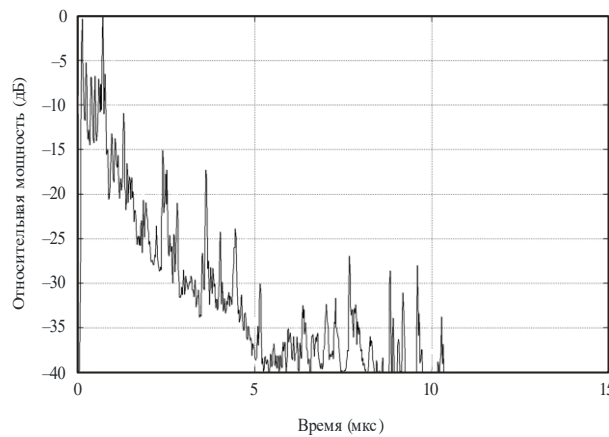


Рис. 1.3. Относительная мощность как функция временной характеристики

Определения статистических параметров. Ниже приводятся соответствующие параметры для статистического описания явлений многолучевости. Средняя задержка – это взвешенное по мощности среднее значение дополнительной задержки, которое определяется первым моментом профиля задержки мощности (квадрат амплитуды импульсной характеристики).

Среднеквадратичный разброс задержки – это взвешенное по мощности стандартное отклонение дополнительных задержек, которые определяются вторым моментом профиля задержки мощности. Этот параметр является мерой изменчивости средней задержки.

Окно задержки – это длина средней части профиля задержки мощности, в которой содержится определенный процент (обычно 90%) общей мощности, передаваемой в соответствии с данной импульсной характеристикой.

Интервал задержки – это часть импульсной характеристики между двумя значениями дополнительной задержки от момента, когда амплитуда

этой характеристики первый раз превышает заданный порог, до момента, когда амплитуда в последний раз падает ниже этого порога.

Количество компонентов многолучевости или сигналов представляет собой число пиковых уровней профиля задержки мощности, амплитуда которых находится в пределах А дБ наиболее высокого пикового уровня и превышает значение минимального уровня шума.

Определения статистических параметров приведены со ссылкой на рис. 1.4 и 1.5. Следует отметить, что на рисунках профили задержки мощности представлены по шкале децибел, в то время как в уравнениях суммирования мощности приводятся линейные единицы мощности.

Общая мощность, p_m , импульсной характеристики определяется как:

$$p_m = \int_{t_0}^{t_3} p(t) dt, \quad (1.1)$$

где: $p(t)$ – плотность мощности импульсной характеристики в линейных единицах мощности; t – задержка относительно опорного времени; t_0 – момент, когда $p(t)$ в первый раз превышает уровень отсечки; t_3 – момент, когда $p(t)$ в последний раз превышает уровень отсечки.

Время средней задержки, T_D , определяется первым моментом профиля задержки мощности:

$$T_D = \frac{\int_0^{\tau_e} \tau p(\tau) d\tau}{\int_0^{\tau_e} p(\tau) d\tau} - \tau_a, \quad (1.2)$$

где: τ – переменная величина дополнительной временной задержки, равная $t - t_0$; τ_a – время прихода первого принятого компонента многолучевого сигнала (первый пик в профиле); $\tau_e = t_3 - t_0$.

В дискретной форме с временным разрешением $\Delta\tau = 1/B$ уравнение (1.2) приобретает вид:

$$T_D = \frac{\sum_{i=1}^N \tau_i p(\tau_i)}{\sum_{i=1}^N p(\tau_i)} - \tau_M, \quad (1.3)$$

где $\tau_i = (i-1)\Delta\tau = (i-1)/B$, ($i = 1, 2, \dots, N$), $i = 1$ и N – индексы первого и последнего из образцов профилей задержки, лежащих выше порогового уровня, соответственно, а M – индекс первого принятого компонента многолучевого сигнала (первый пик в профиле).

Величины задержек могут определяться из следующего соотношения

$$t_i (\text{мкс}) = 3,3 r_i, \text{ км}, \quad (1.4)$$

где r_i – сумма расстояний от передатчика до отражателя многолучевых сигналов и от отражателя до приемника или общее расстояние от передатчика до приемника для t_{LOS} .

Среднеквадратичный разброс задержек, S , определяется как корень квадратный из второго центрального момента:

$$S = \sqrt{\frac{\int_0^{\tau_e} (\tau - T_D - \tau_a)^2 p(\tau) d\tau}{\int_0^{\tau_e} p(\tau) d\tau}}. \quad (1.5)$$

В дискретной форме с временным разрешением Δt уравнение (1.5) приобретает вид:

$$S = \sqrt{\frac{\sum_{i=1}^N (\tau_i - T_D - \tau_M)^2 p(\tau_i)}{\sum_{i=1}^N p(\tau_i)}}. \quad (1.6)$$

Окно задержки, W_q , представляет собой среднюю часть профиля задержки мощности, содержащую определенный процент, q , общей мощности:

$$W_q = t_2 - t_1, \quad (1.7)$$

причем границы t_1 и t_2 определяются соотношением:

$$\int_{t_1}^{t_2} p(t) dt = \frac{q}{100} \int_{t_0}^{t_3} p(t) dt = \frac{q}{100} P_m, \quad (1.8)$$

а мощность за пределами окна делится на две равные части $\left(\frac{100-q}{200}\right) P_m$.

Интервал задержки, I_{th} , определяется как временной интервал между моментом t_4 , когда амплитуда профиля задержки мощности в первый раз превышает заданный порог I_{th} , и моментом t_5 , когда амплитуда в последний раз падает ниже этого порога:

$$I_{th} = t_5 - t_4. \quad (1.9)$$

Количество компонентов многолучевости или сигналов может быть получено из профиля задержки и представлено как число пиковых уровней, амплитуда которых находится в пределах A дБ – наиболее высокого пикового уровня и превышает значение минимального уровня шума, как показано на рисунке 1.4.

На рис. 1.4 показан профиль задержек мощности, иллюстрирующий следующие параметры: окно задержки, W_{90} , содержащее 90% принимаемой мощности, располагается между двумя вертикальными пунктирными линиями (t_1 и t_2), интервал задержки, I_{15} , в котором уровень сигнала превышает

порог "на 15 дБ ниже пикового значения", лежит в пределах между t_4 и t_5 . Начало и конец профиля, превышающего минимальный уровень шума, обозначают t_0 и t_3 .

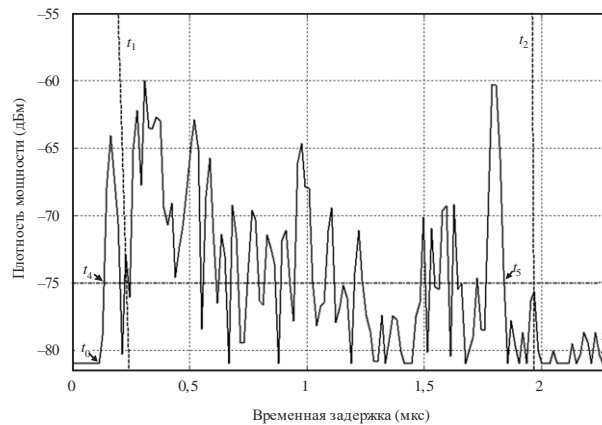


Рис. 1.4. К определению статистических параметров явлений многолучевости

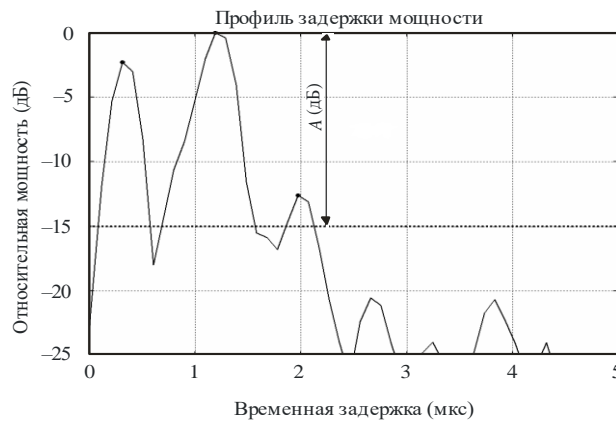


Рис. 1.5. Профиль задержки мощности, иллюстрирующий компоненты многолучевости выше порогового уровня

Рекомендованные параметры. При анализе данных рекомендуется рассматривать окна задержки для 50, 75 и 90% мощности, интервалы задержки для порогов на 9, 12 и 15 дБ ниже максимального значения. Следует отметить, что влияние шума и побочных сигналов в системе (от РЧ до обработки данных) может быть весьма значительным. Поэтому важно точно определить пороговый уровень шумов и/или побочных сигналов для системы и обеспечить для этого уровня необходимый запас надежности. Рекомендуется использовать запас надежности, равный 3 дБ, а для того чтобы обеспечить целостность результатов, рекомендуется минимизировать используемое в качестве критерия приемлемости отношение "пиковый уровень сигнала/уровень побочного сигнала" до, к примеру, 15 дБ (без запаса надежности в 3 дБ), если статистические данные не содержат параметров импульсной характеристики. Порог, используемый для идентификации количества компонентов многолучевости, зависит от динамического диапазона измерительного оборудования – типичное значение на 20 дБ ниже пикового уровня профиля задержки.

1.1.3. ПАРАМЕТРЫ НАПРАВЛЕНИЯ ПРИХОДА

На рис. 1.6 показан угол азимута θ_h и возвышения прихода θ_v .

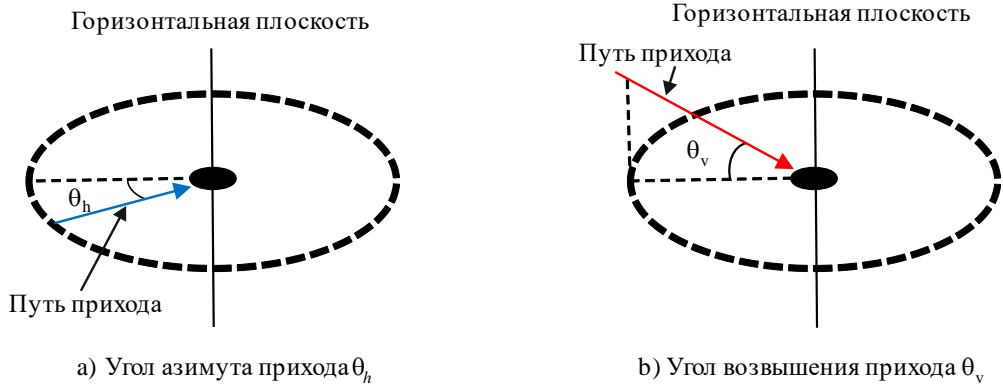


Рис. 1.6. Угол азимута прихода θ_h и угол возвышения прихода θ_v

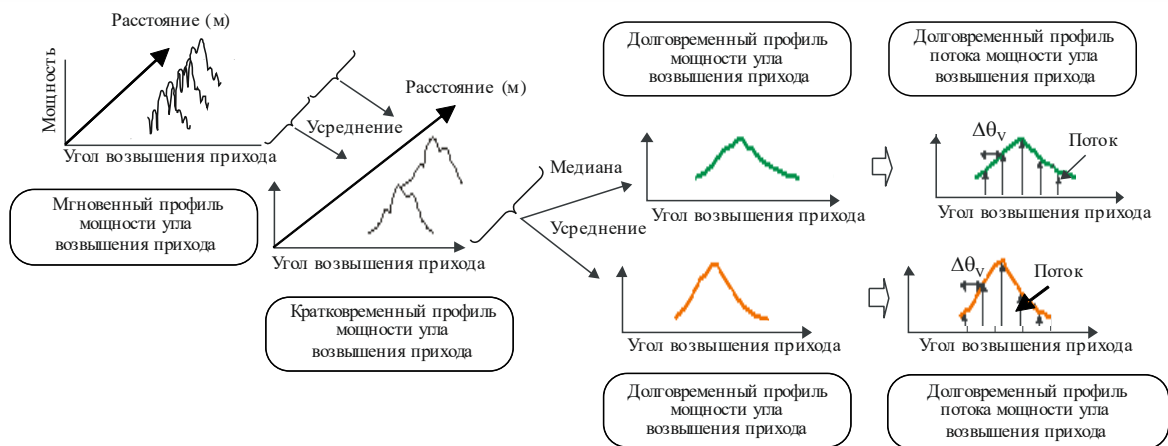
Определение профилей мощности угла азимута или возвышения прихода. Соответствующие параметры для статистического описания угла азимута или возвышения прихода многолучевости могут быть вычислены на основе любого из трех типов профилей мощности угла азимута или возвышения прихода: мгновенного профиля мощности угла азимута или возвышения прихода; кратковременного профиля мощности угла азимута или возвышения прихода; или долговременного профиля мощности угла азимута или возвышения прихода, которые представляют собой либо усредненные по времени значения, полученные при использовании стационарного приемника, отображающие происходящие в окружающей среде перемещения, либо усредненные пространственные значения, полученные при использовании приемника, находящегося в движении.

Определения профилей мощности угла азимута и угла возвышения прихода даны на рис. 1.7.

Мгновенный профиль мощности угла азимута или возвышения прихода представляет собой плотность мощности импульсной характеристики в один момент в одной точке.



а) Определение профилей мощности угла азимута прихода



б) Определение профилей мощности угла возвышения прихода

Рис. 1.7. Определение профилей мощности угла азимута или возвышения прихода

Кратковременный профиль мощности угла азимута или возвышения прихода получают путем пространственного усреднения мгновенных профилей мощности угла азимута или возвышения прихода по нескольким десяткам значений длины волн в пределах диапазона, в котором сохраняются одни и те же компоненты многолучевости, в целях подавления колебаний быстрого замирания.

Долговременный профиль мощности угла азимута или возвышения прихода получают путем пространственного усреднения кратковременных профилей мощности угла азимута или возвышения прихода приблизительно на одном и том же расстоянии от базовой станции (БС) в целях подавления колебаний, вызванных замиранием.

Долговременные профили мощности угла азимута или возвышения прихода с дискретным углом, нормированным по угловому разрешению антенны, определяются как долговременные профили потока мощности угла азимута или возвышения прихода вместо постоянных профилей мощности угла азимута или возвышения прихода.

С другой стороны, долговременный профиль огибающей угла азимута или возвышения прихода представляет собой срединное значение кратковременных профилей потока мощности угла азимута или возвышения прихода приблизительно на том же расстоянии от базовой станции и характеризует форму профиля мощности угла азимута или возвышения прихода в рассматриваемой зоне.

Определения статистических параметров. Определения соответствующих параметров, предназначенных для статистического описания угла азимута или возвышения прихода многолучевости, приведены ниже.

Средний угол азимута или возвышения прихода представляет собой средневзвешенное по мощности значение измеренных направлений азимута

или возвышения прихода и выражается первым моментом спектра мощности азимута или возвышения. (Он может также называться угловым профилем мощности азимута или возвышения.)

Профиль мощности угла азимута или возвышения прихода – это угловая характеристика мощности в пределах плоскости азимута или возвышения.

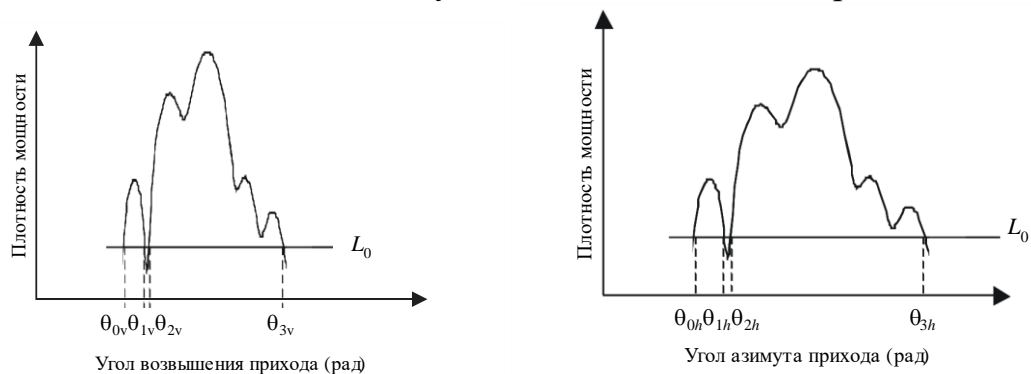
Среднеквадратичный разброс угла азимута или возвышения – это взвешенное по мощности стандартное изменение направления азимута или возвышения прихода, которое выражается вторым моментом профиля мощности угла азимута или возвышения. Используется для измерения среднего угла азимута или возвышения прихода.

Угловое окно азимута или возвышения – это ширина средней части профиля мощности угла азимута или возвышения прихода, содержащей четко определенную процентную долю общей мощности, выявленной в результате этого измерения профиля мощности угла азимута или возвышения прихода.

Угловой интервал (или угловое расстояние) азимута или возвышения – определяется как ширина импульсной характеристики (или ширина углового профиля азимута или возвышения) между двумя значениями направления прихода. Он указывает первый угол азимута или возвышения, при котором амплитуда профиля мощности угла азимута или возвышения прихода превышает данное пороговое значение, а также последний угол азимута или возвышения, при котором эта амплитуда становится ниже этого порогового значения. Используемое пороговое значение зависит от динамического диапазона измерительного оборудования – типичное значение, равное 20 дБ, ниже пикового значения профиля мощности угла азимута или возвышения прихода.

Допустим, что принимаемая мощность в направлении азимута или возвышения, θ_h , θ_v , равна $p(\theta_h)$, $p(\theta_v)$.

Общая мощность, p_{0h} p_{0v} , профиля мощности угла азимута или возвышения прихода определяется как уровень мощности ниже порогового значения L_0 , отделяющего сигнал от шума, как это показано на рис. 1.8.



б) Общая мощность профиля мощности угла возвышения прихода а) Общая мощность профиля мощности угла азимута прихода

Рис. 1.8. Общая мощность

Общая мощность профиля мощности угла азимута и угла возвышения прихода определяются выражениями:

$$P_{0h} = \int_{\theta_{0h}}^{\theta_{3h}} p(\theta_h) d\theta_h, \quad P_{0v} = \int_{\theta_{0v}}^{\theta_{3v}} p(\theta_v) d\theta_v, \quad (1.10)$$

где θ_h, θ_v – измеряются в радианах со стороны азимута или возвышения основного сигнала (который, как предполагается, является стационарным во время проведения измерений); $p(\theta_h), p(\theta_v)$ – мощность профиля мощности угла азимута или возвышения прихода выше порогового уровня L_0 ; ниже L_0 , $p(\theta_h), p(\theta_v) = 0$; L_0 – уровень, незначительно (на 3 дБ, согласно рекомендации) превышающий минимальный уровень шума; θ_{0h}, θ_{0v} – угол азимута или возвышения прихода, когда $p(\theta_h), p(\theta_v)$ впервые превышает пороговый уровень L_0 в $\theta_{\max h}(-\pi, \pi), \theta_{\max v}(-\pi/2, \pi/2)$; θ_{3h}, θ_{3v} – угол азимута или возвышения прихода, когда $p(\theta_h), p(\theta_v)$ в последний раз превышает пороговый уровень L_0 в $\theta_{\max h}(-\pi, \pi), \theta_{\max v}(-\pi/2, \pi/2)$.

В дискретной форме выражение (1.10) для общей мощности профиля мощности угла азимута и угла возвышения прихода приобретают следующий вид:

$$P_{0h} = \sum_{i=1}^N p(\theta_{ih}), \quad P_{0v} = \sum_{i=1}^N p(\theta_{iv}). \quad (1.11)$$

где $i = 1$ и N – это индексы первого и последнего образцов профиля мощности угла азимута или возвышения выше порогового уровня соответственно.

Средний угол азимута или возвышения прихода, T_{Ah}, T_{Av} , определяется первым моментом профиля мощности угла азимута или возвышения:

$$T_{Ah} = \frac{1}{P_{0h}} \int_{\theta_{0h}}^{\theta_{3h}} \theta_h p(\theta_h) d\theta_h, \quad T_{Av} = \frac{1}{P_{0v}} \int_{\theta_{0v}}^{\theta_{3v}} \theta_v p(\theta_v) d\theta_v. \quad (1.12)$$

В дискретной форме при разрешении угла азимута или возвышения, $\Delta\theta_h, \Delta\theta_v$, уравнения (1.12) принимают следующий вид:

$$T_{Ah} = \frac{\sum_{i=1}^N \theta_{ih} p(\theta_{ih})}{\sum_{i=1}^N p(\theta_{ih})}, \quad T_{Av} = \frac{\sum_{i=1}^N \theta_{iv} p(\theta_{iv})}{\sum_{i=1}^N p(\theta_{iv})}, \quad (1.13)$$

где $\theta_{iv} = (i-1)\Delta\theta_v$, ($i = 1, 2, \dots, N$), $i = 1$ и N – это индексы первого и последнего образцов профиля мощности угла азимута или возвышения выше порогового уровня соответственно.

Среднеквадратичный разброс угла азимута или возвышения прихода, S_{Ah}, S_{Av} , определяется следующим образом:

$$S_{Ah} = \sqrt{\frac{1}{P_{0h}} \int_{\theta_{0h}}^{\theta_{3h}} (\theta_h - T_{Ah})^2 p(\theta_h) d\theta_h}, \quad S_{Av} = \sqrt{\frac{1}{P_{0v}} \int_{\theta_{0v}}^{\theta_{3v}} (\theta_v - T_{Av})^2 p(\theta_v) d\theta_v}. \quad (1.14)$$

В дискретной форме при разрешении угла азимута или возвышения, $\Delta\theta_h, \Delta\theta_v$, уравнения (1.14) принимают следующий вид:

$$S_{Ah} = \sqrt{\frac{\sum_{i=1}^N (\theta_{ih} - T_{Ah})^2 p(\theta_{ih})}{\sum_{i=1}^N p(\theta_{ih})}}, \quad S_{Av} = \sqrt{\frac{\sum_{i=1}^N (\theta_{iv} - T_{Av})^2 p(\theta_{iv})}{\sum_{i=1}^N p(\theta_{iv})}}, \quad (1.15)$$

где $i = 1$ и N – это индексы первого и последнего образцов углового профиля мощности выше порогового уровня соответственно.

Угловое окно азимута или возвышения, θ_{wh}, θ_{wv} – это ширина средней части углового профиля мощности, содержащая определенный процент, q , общей мощности, как показано на рис. 1.9.

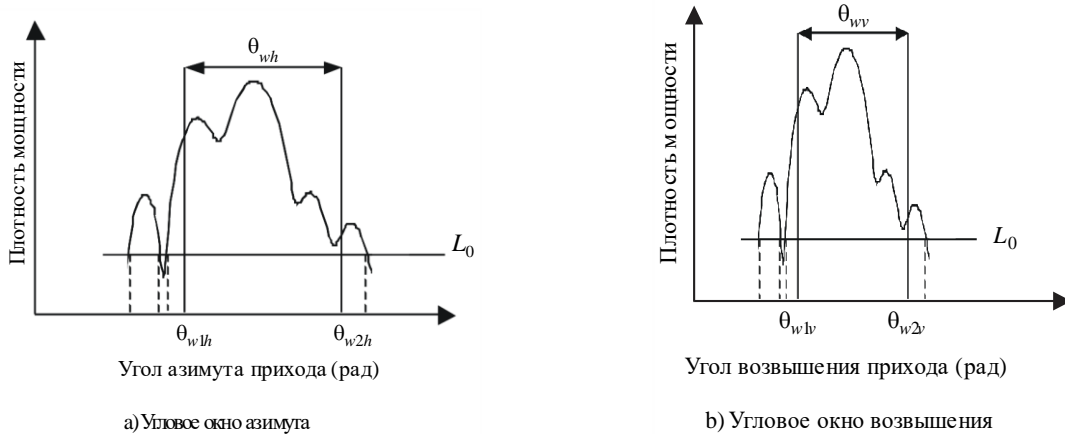


Рис. 1.9. Угловое окно азимута или возвышения

Угловое окно азимута и возвышения определяются выражениями:

$$\theta_{wh} = \theta_{w2h} - \theta_{w1h}, \quad \theta_{wv} = \theta_{w2v} - \theta_{w1v}, \quad (1.16)$$

где границы окна $\theta_{w1h}, \theta_{w1v}$ и $\theta_{w2h}, \theta_{w2v}$ определяются следующим образом:

$$\int_{\theta_{w1h}}^{\theta_{w2h}} p(\theta_h) d\theta_h = \frac{q}{100} \int_{\theta_{0h}}^{\theta_{3h}} p(\theta_h) d\theta_h = \frac{q}{100} P_{0h}, \quad \int_{\theta_{w1v}}^{\theta_{w2v}} p(\theta_v) d\theta_v = \frac{q}{100} \int_{\theta_{0v}}^{\theta_{3v}} p(\theta_v) d\theta_v = \frac{q}{100} P_{0v}, \quad (1.17)$$

а мощность за пределами окна делится на две равные части $\left(\frac{100-q}{200}\right) P_{0h}$,

$\left(\frac{100-q}{200}\right) P_{0v}$.

Угловой интервал азимута или возвышения (угловое расстояние азимута или возвышения), A_{thh}, A_{thv} , определяется как разность между углом азимута или возвышения, θ_{4h}, θ_{4v} , когда амплитуда профиля мощности угла азимута или возвышения приема впервые превышает заданный порог,

L_{th} , и углом азимута или возвышения, θ_{5h} , θ_{5v} , когда она в последний раз опускается ниже этого порога, как показано на рис. 1.10.

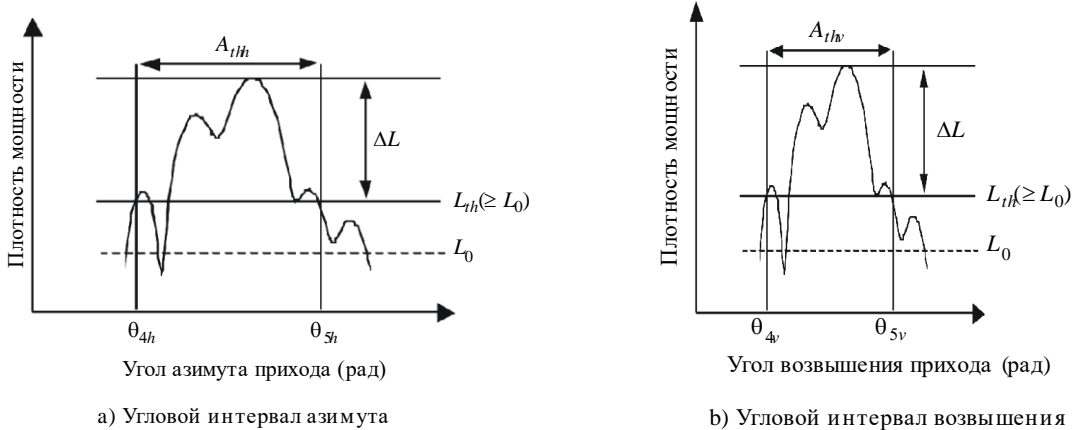


Рис. 1.10. Угловой интервал азимута или возвышения

Угловой интервал азимута и возвышения определяются выражением:

$$A_{thh} = \theta_{5h} - \theta_{4h}, \quad A_{thv} = \theta_{5v} - \theta_{4v}. \quad (1.18)$$

Интервал пространственной корреляции. В частности, для каналов с большим количеством входов и выходов (MIMO) коэффициент пространственной корреляции рассчитывается как комплексная передаточная функция с дисперсией угла азимута или возвышения от профиля мощности угла азимута или возвышения. Коэффициент пространственной корреляции угла азимута или возвышения, $R_h(d)$, $R_v(d)$, вычисляется следующим образом:

$$R_h(d) = \frac{\int_{\theta_{0h}}^{\theta_{3h}} p(\theta_h) \exp\left(-\frac{j2\pi d \sin\theta_h}{\lambda}\right) d\theta_h}{\int_{\theta_{0h}}^{\theta_{3h}} p(\theta_h) d\theta_h}, \quad R_v(d) = \frac{\int_{\theta_{0v}}^{\theta_{3v}} p(\theta_v) \exp\left(-\frac{j2\pi d \sin\theta_v}{\lambda}\right) d\theta_v}{\int_{\theta_{0v}}^{\theta_{3v}} p(\theta_v) d\theta_v}, \quad (1.19)$$

где: d – расстояние для различных интервалов; λ – длина волны.

Как показано на рис. 1.11, расстояние пространственной корреляции d_c определяется как расстояние первой отсечки, при котором $|R_h(d)|$, $|R_v(d)| = x\%$ от $|R_h(d=0)|$, $|R_v(d=0)|$.

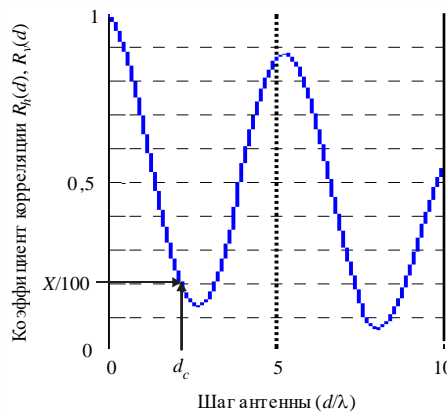


Рис. 1.11. Расстояние пространственной корреляции

Расстояние пространственной корреляции угла азимута и возвышения определяются выражениями:

$$|R_h(d_c)|/|R_h(0)| = x/100, \quad |R_v(d_c)|/|R_v(0)| = x/100. \quad (1.20)$$

Рекомендованные параметры. Для проведения подробного анализа данных рекомендуется рассматривать угловые окна азимута или возвышения для 50, 75 и 90% мощности, угловые интервалы азимута или возвышения для пороговых уровней на 9, 12 и 15 дБ ниже максимального значения и интервалы корреляции для 50 и 90% уровней корреляции. Кроме того, следует отметить, что влияние шума и побочных сигналов в системе (от РЧ до обработки данных) может быть весьма значительным. Поэтому важно точно определить пороговый уровень шумов и/или побочных сигналов для системы и обеспечить для этого уровня необходимый запас надежности. Рекомендуется устанавливать запас надежности, равный 3 дБ, а для того чтобы обеспечить целостность результатов, рекомендуется минимизировать используемое в качестве критерия приемлемости отношение "пиковый уровень сигнала/уровень побочного сигнала" до, например, 15 дБ (без запаса надежности в 3 дБ), что позволит ограничить число включаемых в статистические данные профилей угла азимута или возвышения. На рисунке 10 приведен пример эффекта установления магнитуды минимального отношения пикового значения к L_{th} (ΔL). Предполагается, что на этом рисунке профиль мощности угла азимута или возвышения представляет собой распределение Лапласа (двойное экспоненциальное распределение) с разбросом угла азимута или возвышения в 14° ; разброс угла азимута или возвышения и угловой интервал азимута или возвышения рассчитываются как функции от отношения пикового значения мощности к L_{th} . На данном рисунке показано, что эти параметры претерпевают существенные изменения даже в случае использования в основном идентичных величин. Поэтому следует указать величину, используемую при статистической оценке в качестве ΔL .

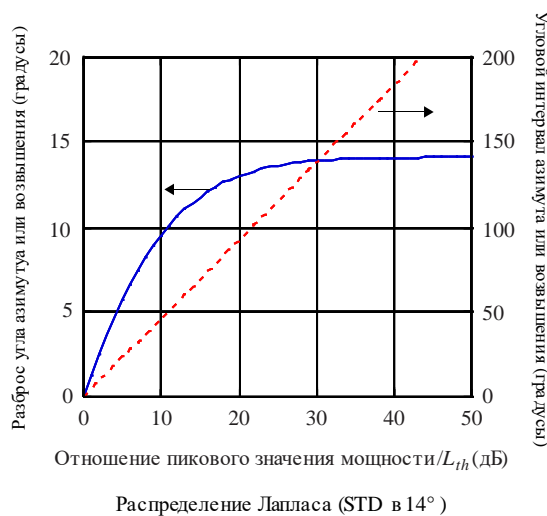


Рис. 1.12. Эффект минимального отношения пикового значения мощности к L_{th} (ΔL)

1.1.4. ПАРАМЕТРЫ ПРОФИЛЯ ЗАДЕРЖКИ ПО УГЛУ МОЩНОСТИ

Профиль задержки по углу мощности получается в результате направленных измерений, позволяющих определить принимаемую мощность в зависимости от угла прихода и задержки, как показано на рис. 1.13.

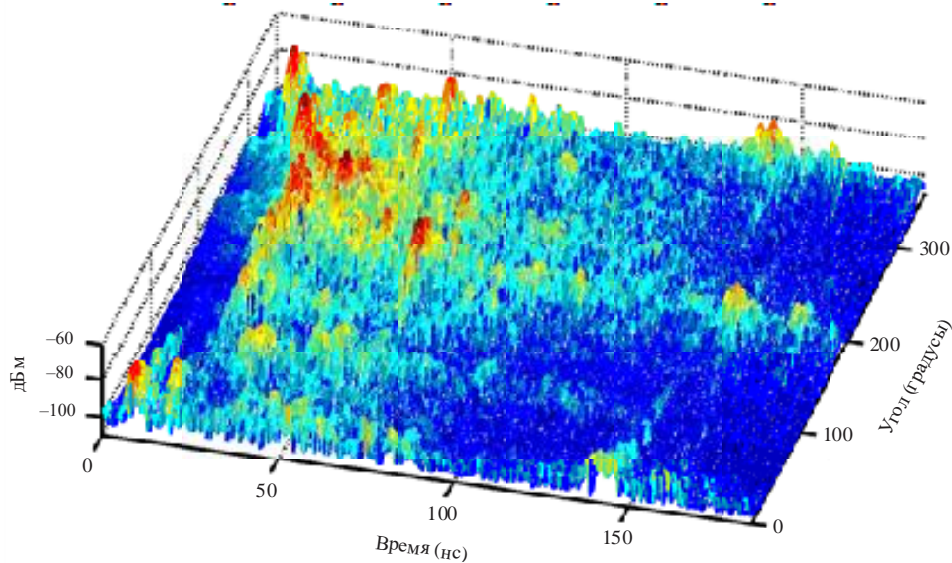


Рис. 1.13. Профили задержки по углу мощности

Всенаправленный профиль задержки мощности как функцию времени задержки можно получить из суммы профилей задержки мощности по всем углам. Суммарная всенаправленная мощность определяется по результатам направленных измерений уравнением:

$$p_m = \int_{\theta_{0v}}^{\theta_{3v}} \int_{\theta_{0h}}^{\theta_{3h}} \int_{t_0}^{t_3} p(t, \theta_h, \theta_v) dt d\theta_h d\theta_v - G_A, \quad (1.21)$$

где: $p(t)$ – плотность мощности импульсной характеристики в линейных единицах мощности, полученная с всенаправленными антеннами; $p(t, \theta_h, \theta_v)$ – направленная плотность мощности импульсной характеристики в линейных единицах мощности; t – задержка относительно времени синхронизации; t_0 – момент, когда $p(t)$ или $p(t, \theta_h, \theta_v)$ впервые превышает уровень отсечки; t_3 – момент, когда $p(t)$ или $p(t, \theta_h, \theta_v)$ превышает уровень отсечки в последний раз; θ_{0h} – угол азимута, когда $p(t, \theta_h, \theta_v)$ превышает уровень отсечки для первого угла; θ_{3h} – угол азимута, когда $p(t, \theta_h, \theta_v)$ превышает уровень отсечки для последнего угла; θ_{0v} – угол возвышения, когда $p(t, \theta_h, \theta_v)$ превышает уровень отсечки для первого угла; θ_{3v} – угол возвышения, когда $p(t, \theta_h, \theta_v)$ превышает уровень отсечки для последнего угла; G_A – коэффициент усиления антенны при пошаговом увеличении угла, меньшем 3-децибеловой ширины луча, которое приводит к перекрытию антенного луча и к увеличению принимаемой мощности G_A .

Для каждого угла прихода можно вычислить параметры временной задержки, как указано в разделе 1.1.2.

1.1.5. ПАРАМЕТРЫ ОТКЛОНЕНИЙ ПРИНЯТОГО СИГНАЛА

Определение отклонений принятого сигнала как функция времени и частоты. Отклонения принимаемого сигнала по времени и в частоте могут быть получены из периодического перемещения по соответствующей полосе частот в коротком интервале времени или из преобразования Фурье краткосрочных импульсных характеристик. Полученные в результате мелкомасштабные, изменяющиеся во времени частотные характеристики $H(f, t)$, изображенные на рис. 1.14, могут быть использованы для формирования ковариационной функции канала $R_H(f, f'; t, t')$ согласно уравнению:

$$R_H(f, f'; t, t') = E \{ H(f, t) H^*(f', t') \}, \quad (1.22)$$

где E – математическое ожидание.

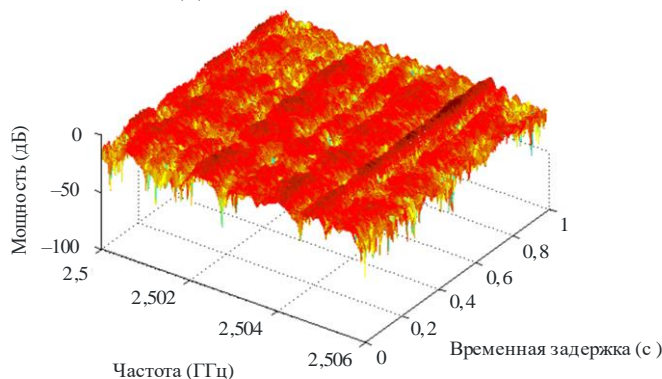


Рис. 1.14. Мелкомасштабная функция частоты, изменяющаяся во времени

При допущении в широком смысле стационарного некоррелированного рассеяния (WSSUS) ковариационная функция в уравнении (1.22) превращается в функцию разности в частоте, Δf , и разности во времени, Δt , $R_H(\Delta f, \Delta t)$.

Степень корреляции выражается нормализованной функцией разнесенной частоты и разнесенного времени, заданной уравнением:

$$\rho(\Delta f, \Delta t) = \frac{R_H(\Delta f, \Delta t)}{\sqrt{E[|H(f, t)|^2] E[|H(f + \Delta f, t + \Delta t)|^2]}}, \quad (1.23)$$

Параметры, связанные с ковариационной функцией в уравнении (1.22) и $H(f, t)$, определены далее.

Определения статистических параметров. Когерентная полоса пропускания или частотная корреляция. Для каналов WSSUS, которые содержат доминантный компонент многолучевости, когерентная (корреляционная) полоса пропускания может быть получена из уравнения:

$$R_H(\Delta f) = R_H(\Delta f, \Delta t) \Big|_{\Delta t=0}. \quad (1.24)$$

Для профилей задержки мощности, демонстрирующих существенную структуру многолучевости, когерентная полоса пропускания может быть рассчитана при помощи преобразования Фурье, $S(f)$, для плотности мощности импульсной характеристики, $p(\tau)$, согласно уравнению:

$$C(f) = \int_0^{\tau_c} p(\tau) \exp(-j2\pi f\tau) d\tau. \quad (1.25)$$

Полоса корреляции, B_x определяется как полоса частот, для которой $|R_H(\Delta f)|$ или $|C(f)|$ равно $x\%$ от $R_H(\Delta f = 0)$ или $C(f = 0)$.

Когерентное время или временная корреляция. Для WSSUS когерентное время рассчитывается на основе временной корреляции канала согласно уравнению:

$$R_H(\Delta t) = R_H(\Delta\omega, \Delta t) \Big|_{\Delta\omega=0}. \quad (1.26)$$

Когерентное время, T_x , определяется как временной интервал, для которого $|R_H(\Delta t)|$ равно $x\%$ от $|R_H(\Delta t = 0)|$.

Частота пересечений уровня и средняя продолжительность замирания. Частота пересечений уровня (LCR) и средняя продолжительность замирания (AFD) могут быть получены из отклонений мощности принимаемого сигнала на отдельно взятой частоте как функция времени или расстояния либо из амплитуды функции частоты, переменной во времени, на отдельно взятой частоте, измеренной как функция времени или расстояния. Для заданного временного интервала LCR – это количество пересечений определенного уровня принимаемым сигналом, а AFD – продолжительность периода, в течение которого он находится ниже заданного уровня. Например, для уровня $-12,5$ дБ на рис. 1.15 изображены LCR и AFD, двойные стрелки обозначают время, в течение которого сигнал находится ниже уровня, а вертикальные стрелки отображают количество пересечений заданного уровня в положительном направлении.

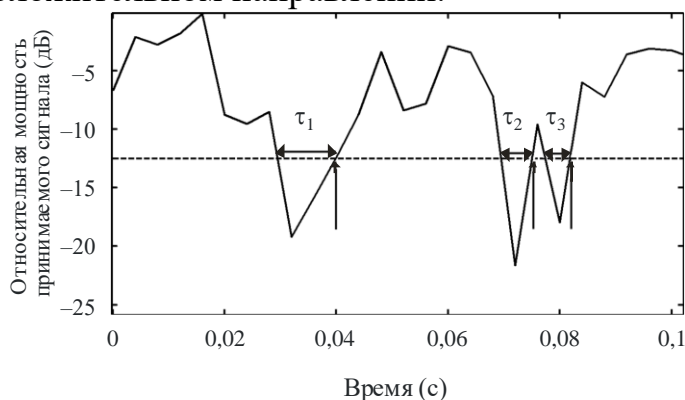


Рис. 1.15. Мощность принимаемого сигнала как функция времени

Частота пересечений уровня и средняя ширина полосы замирания. Частота пересечений уровня (LCR) и средняя ширина полосы замирания (AFBW) могут быть получены из отклонений мощности принимаемого сигнала как функция частоты либо из амплитуды функции частоты, переменной во времени, в отдельно взятый момент времени, как показано на рис. 1.15,

на котором ось времени заменена осью частоты. Для заданной полосы пропускания LCF – это количество пересечений определенного уровня принимаемым сигналом, а AFBW – средний частотный диапазон, который опускается ниже заданного порогового уровня. На рис. 1.16 изображены два параметра, рассчитанных для пороговых уровней от –15 до 0 дБ.

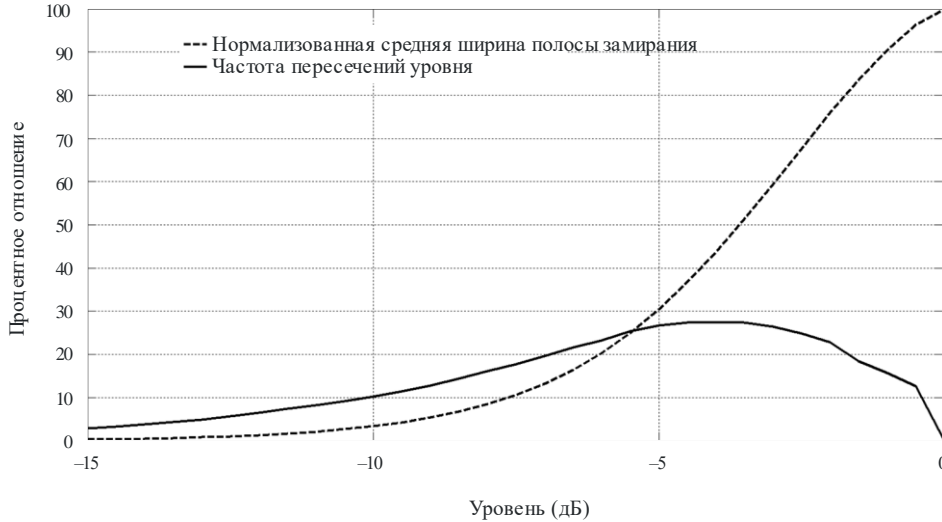


Рис. 1.16. Нормализованная средняя ширина полосы замирания и частота пересечений уровня

Рекомендованные параметры. Ширина полосы корреляции – это полоса частот, в которой автокорреляционная функция передаточной функции находится выше заданного порога; типичные пороговые значения составляют 0,5 и 0,9. LCR рассчитывается, как правило, для числа пересечений уровня в секунду, а LCF – это количество пересечений уровня на МГц.

1.1.6. ПРОСТРАНСТВЕННАЯ КОРРЕЛЯЦИЯ

Далее приведены некоторые результаты вычислений коэффициентов корреляции из углового профиля мощности и примеры воздействия коэффициентов корреляции на пропускную способность при большом количестве входов и выходов (MIMO).

Вычисление коэффициентов пространственной корреляции. Для расчета пространственной корреляции использовалось определение, приведенное в уравнении (1.19):

$$R_h(d) = \frac{\int_{\theta_{0h}}^{\theta_{3h}} p(\theta_h) \exp\left(-\frac{j2\pi d \sin\theta_h}{\lambda}\right) d\theta_h}{\int_{\theta_{0h}}^{\theta_{3h}} p(\theta_h) d\theta_h}, \quad R_v(d) = \frac{\int_{\theta_{0v}}^{\theta_{3v}} p(\theta_v) \exp\left(-\frac{j2\pi d \sin\theta_v}{\lambda}\right) d\theta_v}{\int_{\theta_{0v}}^{\theta_{3v}} p(\theta_v) d\theta_v}.$$

Далее содержится краткое описание полученного результата и характера воздействия на корреляцию шага антенны.

На рис. 1.17 показан идеальный усеченный лапласовый спектр азимутальной мощности (PAS):

$$PAS_L(\phi) = \sum_{k=1}^{N_c} \frac{Q_{L,k}}{\sigma_{L,k} \sqrt{2}} \exp\left[-\frac{\sqrt{2}|\phi - \phi_{0,k}|}{\sigma_{L,k}}\right] \left\{ \varepsilon\left[\phi - (\phi_{0,k} - \Delta\phi_k)\right] - \varepsilon\left[\phi - (\phi_{0,k} + \Delta\phi_k)\right] \right\} \quad (1.27)$$

где $\varepsilon(\phi)$ – это единичная ступенчатая функция и N_c – число кластеров, $\phi_{0,k}$ – средний угол падения k -го кластера, $\sigma_{L,k}$ – угловая скорость. PAS рассчитывается по $[\phi_0 - \Delta\phi, \phi_0 + \Delta\phi]$.

Допустим, что условие нормирования мощности имеет вид:

$$\sum_{k=1}^{N_c} Q_{L,k} \left[1 - \exp\left(-\frac{\sqrt{2}\Delta\phi_k}{\sigma_{L,k}}\right) \right] = 1. \quad (1.28)$$

Тогда коэффициент корреляции по огибающей рассчитывается:

$$\rho_c(D) = \left| R_{XX}(D) + jR_{XY}(D) \right|^2. \quad (1.29)$$

где: $D=2\pi d/\lambda$; d – шаг антенны; λ – длина волны; а функции перекрестной корреляции $R_{XX}(D)$ и $R_{XY}(D)$ определяются на основе вышеприведенного уравнения (1.20): $|R_h(d_c)|/|R_h(0)| = x/100$, $|R_v(d_c)|/|R_v(0)| = x/100$.

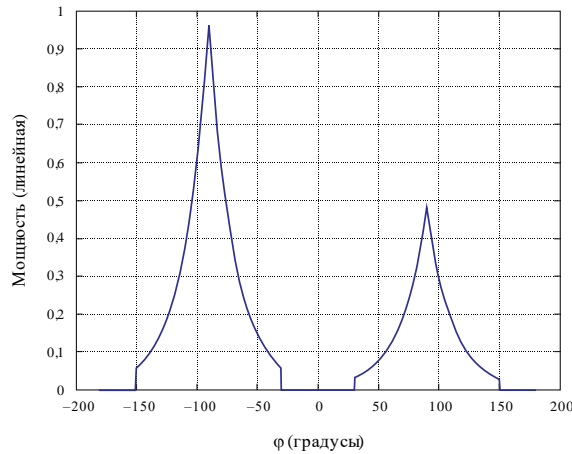


Рис. 1.17. Идеальный усеченный лапласовый спектр азимутальной мощности (PAS)

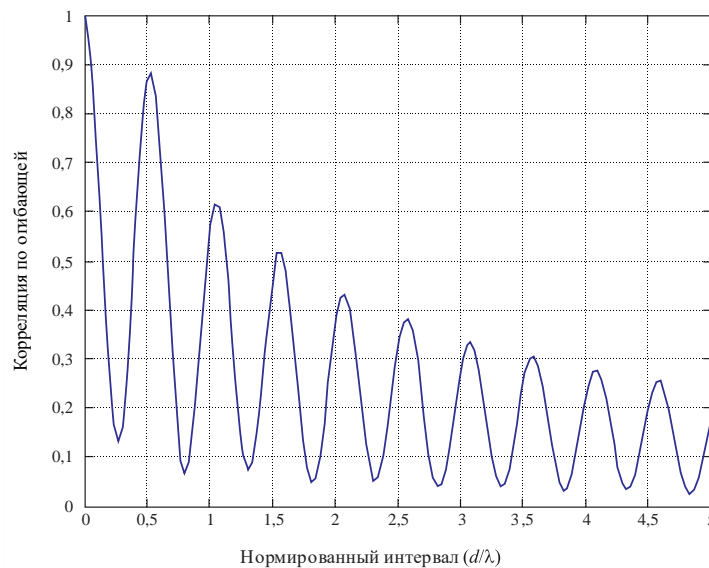


Рис. 1.18. Результирующая пространственная корреляция

На рис. 1.17 показан нормированный лапласовый PAS для двух кластеров. $AS = 30^\circ$, $\phi_0 \in [-90^\circ, +90^\circ]$. Кроме того, мощность кластера $+90^\circ$ составляет половину мощности кластера -90° . На рис. 1.18 показана результирующая пространственная корреляция: коэффициент корреляции по огибающей как функция от величины нормированного интервала ($= d/\lambda$) для двух кластеров, приведенных на рис. 1.17.

Воздействие коэффициентов корреляции на пропускную способность ММО. В случае рэлеевских замирающих каналов эргодическая пропускная способность ММО в отсутствие информации о каналах передатчика рассчитывается следующим образом:

$$\begin{aligned} C &= \log_2 \det \left(\mathbf{I}_{n_R} + \frac{p}{n_T \sigma^2} \mathbf{R}_R^{1/2} \mathbf{H}_W \mathbf{R}_T \mathbf{H}_W^H (\mathbf{R}_R^{1/2})^H \right) = \\ &= \log_2 \det \left(\mathbf{I}_{n_R} + \frac{p}{n_T \sigma^2} \mathbf{H}_W \mathbf{R}_T \mathbf{H}_W^H \mathbf{R}_R^H \right), \end{aligned} \quad (1.30)$$

где: n_R и n_T – количество антенн приемника и передатчика соответственно; p – средняя мощность принимаемого сигнала на одну антенну; σ^2 – мощность шума на каждой приемной антенне; \mathbf{I}_{n_R} – единичная матрица $n_R \times n_R$; $(\cdot)^H$ и $\det(\cdot)$ – эрмитова и определяющая операция соответственно; \mathbf{H}_W – матрица, элементы которой представляют собой независимые идентично распределенные комплексные случайные гауссовы переменные с нулевой средней и удельной дисперсией; $(\cdot)^{1/2}$ – среднеквадратичное эрмитово значение матрицы. Матрицы \mathbf{R}_R и \mathbf{R}_T определяют значение пространственной корреляции между приемниками и передатчиками соответственно, где матрица канала \mathbf{H} рассчитывается по формуле $\mathbf{H} = \mathbf{R}_R^{1/2} \mathbf{H}_W \mathbf{R}_T^{1/2}$, $\mathbf{R}_R^{1/2}$ и $\mathbf{R}_T^{1/2}$ являются позитивными определенными эрмитовыми матрицами, которые в конечном итоге считаются нормированными таким образом, что $[\mathbf{R}_R]_{j,j}$ для $j = 1, K, n_R$ и $[\mathbf{R}_T]_{i,i}$ для $i = 1, K, n_T$.

Если допустить, что \mathbf{R}_R и \mathbf{R}_T имеют полный ранг и $n_R = n_T = n$, тогда при высоком S/N (p/σ^2) приблизительная величина пропускной способности может быть рассчитана следующим образом:

$$C \approx \log_2 \det \left(\frac{p}{n_T \sigma^2} \mathbf{H}_W \mathbf{H}_W^H \right) + \log_2 \det(\mathbf{R}_R) + \log_2 \det(\mathbf{R}_T). \quad (1.31)$$

Если собственные значения \mathbf{R}_R обозначить как λ_i , $i = 1, K, n$, то в этом случае $\sum_{i=1}^n \lambda_i = n$. Из среднеарифметического и среднегеометрического неравенства:

$$\prod_{i=1}^n \lambda_i \leq 1. \quad (1.32)$$

Поскольку $\det(\mathbf{R}_R) = \prod_{i=1}^n \lambda_i$, $\log_2 \det(\mathbf{R}_R) \leq 0$, и равен нулю только в том случае, если все собственные значения \mathbf{R}_R являются одинаковыми, то есть $\mathbf{R}_R = \mathbf{I}_n$. Таким образом, эта корреляция определяет пропускную способность ММО, а потери в эргодической пропускной способности при высоких S/N рассчитываются по формуле $(\log_2 \det(\mathbf{R}_R) + \log_2 \det(\mathbf{R}_T))$, бит/с/Гц.

На рис. 1.19 проиллюстрировано воздействие величины пространственной корреляции на эргодическую пропускную способность канала ММО при $n_R=n_T=2$.

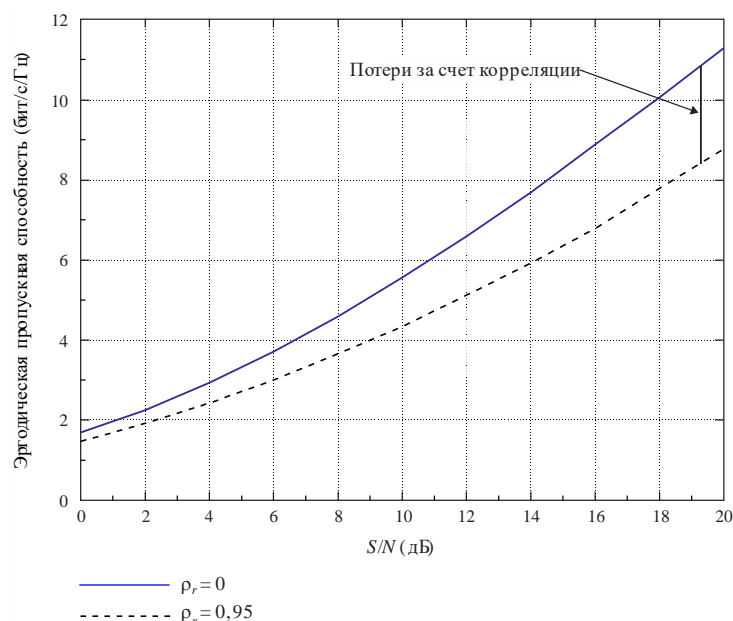


Рис. 1.19. Эргодическая пропускная способность при низкой и высокой корреляции приема

Предполагается, что на рисунке $\mathbf{R}_T = \mathbf{I}_2$. Матрица корреляции приемника выбрана в соответствии с формулой:

$$\mathbf{R}_R = \begin{bmatrix} 1 & \rho_R \\ \rho_R^* & 1 \end{bmatrix}, \quad (1.33)$$

где ρ_R означает пространственную корреляцию между антеннами приемника.

1.1.7. РАЗРЕШЕНИЕ КОМПОНЕНТОВ МНОГОЛУЧЕВОСТИ

Разрешение компонентов многолучевости в измеряемых данных зависит от полосы пропускания сигнала, используемого при измерениях. Неразрешенные компоненты многолучевости вызывают отклонения сигналов во времени или пространстве, связанные с перемещениями передатчика или приемника либо с изменениями условий распространения, как показано на рис.1.20. Эти отклонения могут быть смоделированы при помощи функций

плотности вероятности, например, функций Рэля или Райса, как указано в Рекомендации МСЭ-R P.1057 [2].

Формирование широкополосного канала. Переменная во времени импульсная характеристика может использоваться для моделирования канала в качестве линии задержки с отводами, как показано на рис. 1.20, где каждый отвод имеет задержку $\Delta\tau$, что соответствует разрешению временной задержки многолучевости и коэффициенту отвода, который представляет колебания во времени неразрешенной группы компонентов многолучевости в данном интервале временной задержки.

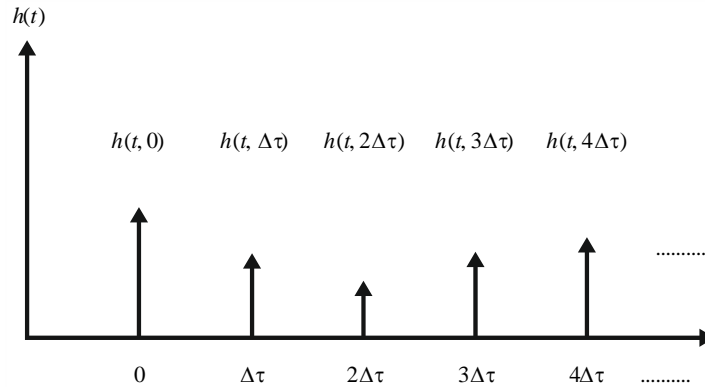


Рис. 1.20. Примеры компонентов многолучевости, используемых для формирования канала

Для имитации системы достаточно заменить многочисленные рассеянные трассы, которые могут существовать в реальном канале, всего несколькими компонентами многолучевости $m=n+1$ в модели, как показано на рис. 1.21. Таким образом получаем характеристику канала $h(t)$ в уравнении:

$$h(t) = \sum_{i=0}^n h_i \delta(t - i\Delta\tau). \quad (1.34)$$

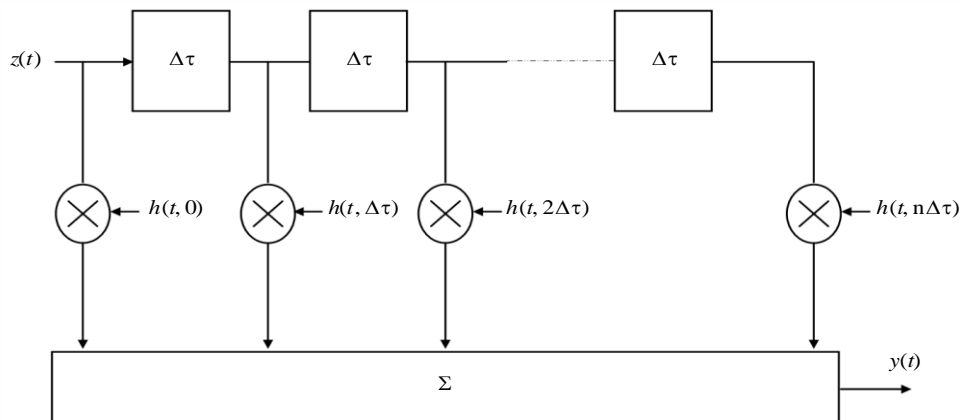


Рис. 1.21. Модель многолучевости на основе линии задержки с отводами

Для временных задержек, которые имеют доминирующий компонент или компонент прямой видимости (LOS), модель канала для каждой группы компонентов многолучевости задается функцией плотности вероятности

Райса. Если коэффициент Райса, K , равен нулю, то может использоваться модель Рэлея. Обобщенная модель канала приведена в уравнении:

$$h(t) = \sum_{i=0}^n \left\{ \sqrt{\frac{K_i p_i}{K_i + 1}} e^{j(2\pi f_{o,i} t + \phi_{o,i})} + \sqrt{\frac{p_i}{K_i + 1}} g_i(t) \right\} \delta(t - i\Delta\tau). \quad (1.35)$$

где: K_i – K -фактор Райса для i -го компонента, определяемый как отношение мощности доминирующего или LOS-компонента к рассеянному компоненту; при $K_i = 0$ получаем рэлеевское распределение; p_i – усредненная мощность i -го компонента в $h(t)$ и равная $p_i = E[|h_i(t)|^2]$; $p_i = E[|h_i(t)|^2]_{f_{o,i}}$ – доплеровская частота доминирующего или LOS-компонента i -го компонента в $h(t)$ равна $F_{D_{max,i}} \cos \theta_{o,i}$, $f_{D_{max,i}} \cos \theta_{o,i}$ где $f_{D_{max,i}}$ – максимальное доплеровское смещение, а $\theta_{o,i}$ – азимутальный угол прихода; $\phi_{o,i}$ – начальная фаза LOS-компонента i -го компонента в $h(t)$; $g_i(t)$ – удельная мощность, нулевое среднее значение, комплексный сигнал, представляющий компоненты диффузного рассеяния. Для большого количества рассеянных сигналов $g_i(t)$ может обрабатываться как комплексный гауссов случайный процесс с единичной дисперсией, прошедшей через i -й доплеровский фильтр.

1.2. ЛАБОРАТОРНАЯ РАБОТА. ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ РАДИОКАНАЛА С ЗАМИРАНИЯМИ

Цель работы: построить имитационную модель для оценки помехоустойчивости систем передачи сигналов в радиоканале с АБГШ и замираниями

Скрипт 1.3.1. Скрипт для имитационного моделирования радиоканала с замираниями

```
% bpskev.m
% Evaluation program of fading counter based BPSK transmission scheme
% This program is one of example simulations that include fading
% As for the explanation, you can check Chapter 3.

%***** Preparation part *****

% Time resolution
% In this case, 0.5us is used as an example
tstp = 0.5*1.0e-6;

% Symbol rate
% In this case we assume that each sample time is equal to 1/(symbol rate).
% In this case 200 kbps is considered.
sr = 1/tstp ;

% Arrival time for each multipath normalized by tstp
% In this simulation four-path Rayleigh fading are considered
itau = [0, 2, 3, 4];

% Mean power for each multipath normalized by direct wave.
% In this simulation four-path Rayleigh fading are considered.
% This means that the second path is -10dB less than the first direct path.
dlvl = [0 ,10 ,20 ,25];

% Number of waves to generate fading for each multipath.
% In this simulation four-path Rayleigh fading are considered.
% In normal case, more than six waves are needed to generate Rayleigh fading
n0=[6,7,6,7];

% Initial Phase of delayed wave
% In this simulation four-path Rayleigh fading are considered.
th1=[0.0,0.0,0.0,0.0];

% Number of fading counter to skip (50us/0.5us)
% In this case we assume to skip 50 us
itnd0=100*2;

% Initial value of fading counter
% In this simulation four-path Rayleigh fading are considered.
% Therefore four fading counter are needed.

itnd1=[1000,2000, 3000, 4000];

% Number of directwave + Number of delayed wave
% In this simulation four-path Rayleigh fading are considered
```

```

nowl=4;

% Maximum Doppler frequency [Hz]
% You can insert your favorite value
fd=200;

% Number of data to simulate one loop
% In this case 100 data are assumed to consider
nd = 100;

% You can decide two mode to simulate fading by changing the variable flat
% flat      : flat fading or not
% (1->flat (only amplitude is fluctuated),
% 0->nomal(phase and amplitude are fluctutated)
flat =1;

%***** START CALCULATION *****

nloop = 1000; % Number of simulation loop
noe = 0; % Initial number of errors
nod = 0; % Initial number of transmitted data

for iii=1:nloop

%***** Data generation *****

    data=rand(1,nd)>0.5; % rand: built in function

%***** BPSK modulation *****

    data1=data.*2-1; % Change data from 1 or 0 notation to +1 or -1 nota-
tion

%***** Fading channel *****

    % Generated data are fed into a fading simulator
    % In the case of BPSK, only Ich data are fed into fading counter
    [data6,data7]=sefade(data1,zeros(1,length(data1)),itau,dlv1,th1,n0,...
        itnd1,nowl,length(data1),tstp,fd,flat);

    % Updata fading counter
    itnd1 = itnd1+ itnd0;

%***** BPSK Demodulation *****

    demodata=data6 > 0;

%***** Bit Error Rate (BER) *****

    % count number of instantaneous errors
    noe2=sum(abs(data-demodata)); % sum: built in function

    % count number of instantaneous transmitted data
    nod2=length(data); % length: built in function

```

```

fprintf('%d\t%e\n',iii,noe2/nod2);

noe=noe+noe2;
nod=nod+nod2;

end % for iii=1:nloop

%***** Output result *****
%ber = noe/nod;
fprintf('%d\t%d\t%e\n',noe,nod,noe/nod);

```

Скрипт 1.3.2. Функция для имитационного моделирования задержек в радиоканале с замираниями

```

% delay.m
% Gives delay to input signal

function [iout,qout] = delay( idata, qdata , nsamp , idel )

%***** variables *****
% idata  input Ich data
% qdata  input Qch data
% iout   output Ich data
% qout   output Qch data
% nsamp  Number of samples to be simulated
% idel   Number of samples to be delayed
%*****

iout=zeros(1,nsamp);
qout=zeros(1,nsamp);

if idel ~= 0
    iout(1:idel) = zeros(1,idel);
    qout(1:idel) = zeros(1,idel);
end

iout(idel+1:nsamp) = idata(1:nsamp-idel);
qout(idel+1:nsamp) = qdata(1:nsamp-idel);

```

Скрипт 1.3.3. Функция для имитационного моделирования рэлеевских замираний в радиоканале

```

% fade.m
% Generate Rayleigh fading

function [iout,qout,ramp,rcos,rsin]=fade(idata,qdata,nsamp,tstp,...
    fd,no,counter,flat)

%***** variables *****
% idata  : input Ich data
% qdata  : input Qch data
% iout   : output Ich data
% qout   : output Qch data
% ramp   : Amplitude contaminated by fading
% rcos   : Cosine value contaminated by fading
% rsin   : Cosine value contaminated by fading
% nsamp  : Number of samples to be simulated
% tstp   : Minimum time resolution
% fd     : maximum doppler frequency

```

```

% no      : number of waves in order to generate fading
% counter : fading counter
% flat    : flat fading or not
% (1->flat (only amplitude is fluctuated),
% 0->nomal(phase and amplitude are fluctuated)
%*****

if fd ~= 0.0
    ac0 = sqrt(1.0 ./ (2.0.*(no + 1))); % power normalized constant(ich)
    as0 = sqrt(1.0 ./ (2.0.*no));      % power normalized constant(qch)
    ic0 = counter;                    % fading counter

    pai = 3.14159265;
    wm = 2.0.*pai.*fd;
    n = 4.*no + 2;
    ts = tstp;
    wmts = wm.*ts;
    paino = pai./no;

    xc=zeros(1,nsamp);
    xs=zeros(1,nsamp);
    ic=[1:nsamp]+ic0;

    for nn = 1: no
        cwn = cos( cos(2.0.*pai.*nn./n).*ic.*wmts );
        xc = xc + cos(paino.*nn).*cwn;
        xs = xs + sin(paino.*nn).*cwn;
    end

    cwmt = sqrt(2.0).*cos(ic.*wmts);
    xc = (2.0.*xc + cwmt).*ac0;
    xs = 2.0.*xs.*as0;

    ramp=sqrt(xc.^2+xs.^2);
    rcos=xc./ramp;
    rsin=xs./ramp;

    if flat ==1
        iout = sqrt(xc.^2+xs.^2).*idata(1:nsamp); % output signal(ich)
        qout = sqrt(xc.^2+xs.^2).*qdata(1:nsamp); % output signal(qch)
    else
        iout = xc.*idata(1:nsamp) - xs.*qdata(1:nsamp); % output signal(ich)
        qout = xs.*idata(1:nsamp) + xc.*qdata(1:nsamp); % output signal(qch)
    end

else
    iout=idata;
    qout=qdata;
end
end

```

Скрипт 1.3.4. Функция для имитационного моделирования рэлеевских замираний в радиоканале

```

% sefade.m
% This function generates frequency selecting fading

function[iout,qout,ramp,rcos,rsin]=sefade(idata,qdata,itau,dlv1,th,n0,...
    itn,n1,nsamp,tstp,fd,flat)

```

```

%***** variables *****
% idata  input Ich data
% qdata  input Qch data
% iout   output Ich data
% qout   output Qch data
% ramp   : Amplitude contaminated by fading
% rcos   : Cosine value contaminated by fading
% rsin   : Cosine value contaminated by fading
% itau   : Delay time for each multipath fading
% dlvl   : Attenuation level for each multipath fading
% th     : Initialized phase for each multipath fading
% n0     : Number of waves in order to generate each multipath fading
% itn    : Fading counter for each multipath fading
% n1     : Number of summation for direct and delayed waves
% nsamp  : Total number of symbols
% tstp   : Minimum time resolution
% fd     : Maximum doppler frequency
% flat   flat fading or not
% (1->flat (only amplitude is fluctuated),
% 0->nomal(phase and amplitude are fluctuated)
%*****

iout = zeros(1,nsamp);
qout = zeros(1,nsamp);

total_attn = sum(10 .^( -1.0 .* dlvl ./ 10.0));

for k = 1 : n1

    atts = 10.^( -0.05 .* dlvl(k));

    if dlvl(k) >= 40.0
        atts = 0.0;
    end

    theta = th(k) .* pi ./ 180.0;

    [itmp,qtmp] = delay ( idata , qdata , nsamp , itau(k));
    [itmp3,qtmp3,ramp,rcos,rsin] = fade (itmp,qtmp,nsamp,tstp,...
        fd,n0(k),itn(k),flat);

    iout = iout + atts .* itmp3 ./ sqrt(total_attn);
    qout = qout + atts .* qtmp3 ./ sqrt(total_attn);

end

```


2. МЕЖСИМВОЛЬНАЯ ИНТЕРФЕРЕНЦИЯ И ФОРМИРУЮЩИЕ ФИЛЬТРЫ

2.1. ЛЕКЦИЯ. МЕЖСИМВОЛЬНАЯ ИНТЕРФЕРЕНЦИЯ И ФОРМИРУЮЩИЕ ФИЛЬТРЫ НАЙКВИСТА

Цель занятия: изучить влияние межсимвольной интерференции на передачу и прием сигналов.

2.2. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ. РАЗРАБОТКА ФОРМИРУЮЩЕГО ФИЛЬТРА НАЙКВИСТА

Цель занятия: изучить особенности компенсации влияния межсимвольной интерференции средствами формирующего фильтра Найквиста.

2.3. ЛАБОРАТОРНАЯ РАБОТА. ОЦЕНКА ФУНКЦИОНИ- РОВАНИЯ ФОРМИРУЮЩЕГО ФИЛЬТРА НАЙКВИСТА

Цель занятия: изучить особенности моделей распространения радиоволн (РРВ) в системах мобильной связи (СМС).

Скрипт 2.3.2. Функция для формирования коэффициентов формирующего фильтра Найквиста

```
% Program 3-3
% hrollfcoef.m
% Generate coefficients of Nyquist filter

function [xh] = hrollfcoef(irfn,ipoint,sr,alfs,ncc)

%***** variables *****
% irfn    : Number of symbols to use filtering
% ipoint  : Number of samples in one symbol
% sr      : symbol rate
% alfs    : rolloff coeficiense
% ncc     : 1 -- transmitting filter  0 -- receiving filter
% *****

xi=zeros(1,irfn*ipoint+1);
xq=zeros(1,irfn*ipoint+1);

point = ipoint;
tr = sr ;
tstp = 1.0 ./ tr ./ ipoint;
n = ipoint .* irfn;
mid = ( n ./ 2 ) + 1;
sub1 = 4.0 .* alfs .* tr;           % 4*alpha*R_s

for i = 1 : n

    icon = i - mid;
    ym = icon;

    if icon == 0.0
```

```

    xt = (1.0-alfs+4.0.*alfs./pi).* tr; % h(0)
else
    sub2 =16.0.*alfs.*alfs.*ym.*ym./ipoint./ipoint;
    if sub2 ~= 1.0
        x1=sin(pi*(1.0-alfs)/ipoint*ym)./pi./(1.0-sub2)./ym./tstp;
        x2=cos(pi*(1.0+alfs)/ipoint*ym)./pi.*sub1./(1.0-sub2);
        xt = x1 + x2; % h(t) plot(1:length(xh)),xh)
    else % (4alphaRst)^2 = 1plot(1:length(xh)),xh)
        xt = alfs.*tr.*((1.0-2.0/pi).*cos(pi/4.0/alfs)+...
            (1.0+2.0./pi).*sin(pi/4.0/alfs))./sqrt(2.0);
    end % if sub2 ~= 1.0
end % if icon == 0.0

if ncc == 0 % in the case of receiver
    xh( i ) = xt ./ ipoint ./ tr; % normalization
elseif ncc == 1 % in the case of transmitter
    xh( i ) = xt ./ tr; % normalization
else
    error('ncc error');
end % if ncc == 0

end % for i = 1 : n

```

Скрипт 2.3.3. Функция передискретизации для оценки функционирования формирующего фильтра Найквиста

```

% Program 3-4
% oversamp.m
% Insert zero data to input data
function [out] = oversamp( indata, nsymb , sample)
%***** variables *****
% indata : input sequence
% nsymb : Number of symbols
% sample : Number of oversample
% *****
out=zeros(1,nsymb*sample);
out(1:sample:1+sample*(nsymb-1))=indata;

```

3. ЦИФРОВЫЕ СИГНАЛЫ ФАЗОВОЙ МАНИПУЛЯЦИИ

3.1. ЛЕКЦИЯ. ЦИФРОВЫЕ СИГНАЛЫ ФАЗОВОЙ МАНИПУЛЯЦИИ

Цель занятия: изучить цифровые сигналы фазовой манипуляции BPSK, QPSK, OQPSK, QAM.

3.2. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ. РАЗРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ ДЛЯ ОЦЕНКИ ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ BPSK, QPSK, DBPSK, DQPSK, OQPSK, QAM В КАНАЛЕ С АБГШ И ЗАМИРАНИЯМИ

Цель занятия: разработать имитационную модель оценки помехоустойчивости сигналов BPSK, QPSK, OQPSK, QAM в канале с АБГШ и замираниями.

Скрипт 3.2.1. Функция для модуляции сигналов QPSK

```
% qpskmod.m
%
% Function to perform QPSK modulation

function [iout,qout]=qpskmod(paradata,para,nd,ml)

%***** variables *****
% paradata : input data (para-by-nd matrix)
% iout :output I ch data
% qout :output Q ch data
% para    : Number of paralell channels
% nd     : Number of data
% ml     : Number of modulation levels
% (QPSK ->2  16QAM -> 4)
% *****

m2=ml./2;

paradata2=paradata.*2-1;
count2=0;

for jj=1:nd

    isi = zeros(para,1);
    isq = zeros(para,1);

    for ii = 1 : m2
        isi = isi + 2.^( m2 - ii ) .* paradata2((1:para),ii+count2);
        isq = isq + 2.^( m2 - ii ) .* paradata2((1:para),m2+ii+count2);
    end

    iout((1:para),jj)=isi;
    qout((1:para),jj)=isq;
```

```

count2=count2+m1;

end

```

Скрипт 3.2.2. Функция для демодуляции сигналов QPSK

```

% qpskdemod.m
%
% Function to perform QPSK demodulation

function [demodata]=qpskdemod(idata,qdata,para,nd,m1)

%***** variables *****
% idata :input Ich data
% qdata :input Qch data
% demodata: demodulated data (para-by-nd matrix)
% para   : Number of paralell channels
% nd    : Number of data
% m1   : Number of modulation levels
% (QPSK ->2  16QAM -> 4)
% *****

demodata=zeros(para,m1*nd);
demodata((1:para),(1:m1:m1*nd-1))=idata((1:para),(1:nd))>=0;
demodata((1:para),(2:m1:m1*nd))=qdata((1:para),(1:nd))>=0;

```

Скрипт 3.2.3. Функция для модуляции сигналов QAM

```

% qammod.m
%
% This function is used for Gray coding of 16QAM modulation

function [iout,qout]=qammod(paradata,para,nd,m1)

%***** variables *****
% paradata : input data (para-by-nd matrix)
% iout    :output Ich data
% qout    :output Qch data
% para   : Number of paralell channels
% nd    : Number of data
% m1   : Number of modulation levels
% (QPSK ->2  16QAM -> 4)
% *****

% The constellation power

k=sqrt(10);
iv=[-3 -1 3 1];

m2=m1/2;
count2=0;

for ii=1 : nd

    isi = zeros(para,1);
    isq = zeros(para,1);

```

```

for jj=1 : m1
    if jj = m2
        isi = isi +2.^( m2- jj ).*paradata((1:para),count2+jj);
    else
        isq = isq +2.^( m1- jj ).*paradata((1:para),count2+jj);
    end
end

iout((1:para),ii) = iv(isi+1)./k;
qout((1:para),ii) =iv(isq+1)./k;

count2=count2+m1;

end

```

Скрипт 3.2.4. Функция для демодуляции сигналов QAM

```

% Program 3-24
% qamdemod.m
%
% Function to decode 16QAM modulation

function [demodata]=qamdemod(idata,qdata,para,nd,m1)

%***** variables *****
% idata :input Ich data
% qdata :input Qch data
% demodata: demodulated data (para-by-nd matrix)
% para   : Number of paralell channels
% nd    : Number of data
% m1    : Number of modulation levels
% (QPSK ->2  16QAM -> 4)
% *****

k=sqrt(10);
idata=idata.*k;
qdata=qdata.*k;
demodata=zeros(para,m1*nd);

m2=m1/2;
count2=0;

for ii = 1:nd

    a=1;
    b=1;
    i_lngth=0;
    q_lngth=0;

    for jj= 1:m2

        if jj ~= 1

            if demodata((1:para),jj-1+count2)==1
                a=-a;
            end

```

```

        if demodata((1:para),m2+jj-1+count2)==1
            b=-b;
        end

        i_lngth=i_lngth+i_plrty.*2.^(m2-jj+1);
        q_lngth=q_lngth+q_plrty.*2.^(m2-jj+1);
    end

    if idata((1:para),ii) >= i_lngth
        demodata((1:para),jj+count2)=a>=0;
        i_plrty=1;
    else
        demodata((1:para),jj+count2)=a=0;
        i_plrty=-1;
    end

    if qdata((1:para),ii) >= q_lngth
        demodata((1:para),m2+jj+count2)=b>=0;
        q_plrty=1;
    else
        demodata((1:para),m2+jj+count2)=b=0;
        q_plrty=-1;
    end

end % for jj= 1:m2

count2=count2+m1;

end % for ii = 1:nd

```

3.3. ЛАБОРАТОРНАЯ РАБОТА. ОЦЕНКА ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ BPSK, QPSK, DBPSK, DQPSK, OQPSK, QAM В КАНАЛЕ С ЗАМИРАНИЯМИ

Цель занятия: оценить помехоустойчивость сигналов BPSK, QPSK, OQPSK, QAM в канале с АБГШ и замираниями.

3.3.1. ОЦЕНКА ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ BPSK

Скрипт 3.3.1. Скрипт для оценки помехоустойчивости сигналов BPSK в канале с АБГШ

```

% Program 3-1
% bpsk.m
% Simulation program to realize BPSK transmission system

%***** Preparation part *****
sr=256000.0; % Symbol rate
ml=1; % Number of modulation levels
br=sr.*ml; % Bit rate (=symbol rate in this case)
nd = 1000; % Number of symbols that simulates in each loop
ebn0=3; % Eb/N0
IPOINT=8; % Number of oversamples

%***** Filter initialization *****
irfn=21; % Number of filter taps

```

```

alfs=0.5;      % Rolloff factor
[xh] = hrollfcoef(irfn,IPOINT,sr,alfs,1);   %Transmitter filter coefficients
[xh2] = hrollfcoef(irfn,IPOINT,sr,alfs,0);  %Receiver filter coefficients

%***** START CALCULATION *****
nloop=100;    % Number of simulation loops
noe = 0;      % Number of error data
nod = 0;      % Number of transmitted data

for iii=1:nloop

%***** Data generation *****
    data=rand(1,nd)>0.5; % rand: built in function

%***** BPSK Modulation *****
    data1=data.*2-1;
    [data2] = oversamp( data1, nd , IPOINT) ;
    data3 = conv(data2,xh); % conv: built in function

%***** Attenuation Calculation *****
    spow=sum(data3.*data3)/nd;
    attn=0.5*spow*sr/br*10.^(-ebn0/10);
    attn=sqrt(attn);

%***** Fading channel *****
    % Generated data are fed into a fading simulator
    % In the case of BPSK, only Ich data are fed into fading counter
    % [ifade,qfade]=sefade(data3,zeros(1,length(data3)),itau,dlvl,th1,n0,...
    %                                     itnd1,nowl,length(data3),tstp,fd,flat);
    % Update fading counter
    %itnd1 = itnd1+ itnd0;

%***** Add White Gaussian Noise (AWGN) *****
    inoise=randn(1,length(data3)).*attn; % randn: built in function
    data4=data3+inoise;
    data5=conv(data4,xh2); % conv: built in function

    sampl=irfn*IPOINT+1;
    data6 = data5(sampl:8:8*nd+sampl-1);

%***** BPSK Demodulation *****
    demodata=data6 > 0;

%***** Bit Error Rate (BER) *****
    noe2=sum(abs(data-demodata)); % sum: built in function
    nod2=length(data); % length: built in function
    noe=noe+noe2;
    nod=nod+nod2;

    fprintf('%d\t%e\n',iii,noe2/nod2);
end % for iii=1:nloop

%***** Output result *****
ber = noe/nod;
fprintf('%d\t%d\t%d\t%e\n',ebn0,noe,nod,noe/nod);
fid = fopen('BERbpsk.dat','a');

```

```

fprintf(fid, '%d\t%e\t%f\t%f\t\n', ebn0, noe/nod, noe, nod);
fclose(fid);
%***** end of file *****

```

Скрипт 3.3.2. Скрипт для оценки помехоустойчивости сигналов BPSK в канале с замираниями

```

% Program 3-2 bpsk_fading.m
% Simulation program to realize BPSK transmission system
% (under one path fading)

%***** Preparation part *****
sr=256000.0; % Symbol rate
ml=1; % Number of modulation levels
br=sr.*ml; % Bit rate (=symbol rate in this case)
nd = 100; % Number of symbols that simulates in each loop
ebn0=5; % Eb/N0
IPOINT=8; % Number of oversamples

%***** Filter initialization *****
irfn=21; % Number of filter taps
alfs=0.5; % Rolloff factor
[xh] = hrollfcoef(irfn, IPOINT, sr, alfs, 1); %Transmitter filter coefficients
[xh2] = hrollfcoef(irfn, IPOINT, sr, alfs, 0); %Receiver filter coefficients

%***** Fading initialization *****
% Time resolution
tstp=1/sr/IPOINT;

% Arrival time for each multipath normalized by tstp
% If you would like to simulate under one path fading model
itau = [0];

% Mean power for each multipath normalized by direct wave.
dlvl = [0];

% Number of waves to generate fading for each multipath.
% In normal case, more than six waves are needed to generate Rayleigh fading
n0=[6];

% Initial Phase of delayed wave
th1=[0.0];

% Number of fading counter to skip
itnd0=nd*IPOINT*100;

% Initial value of fading counter
% In this simulation one-path Rayleigh fading are considered.
% Therefore one fading counter are needed.
itnd1=[1000];

% Number of directwave + Number of delayed wave
% In this simulation one-path Rayleigh fading are considered
nowl=1;

% Maximum Doppler frequency [Hz]
fd=160;

```



```

% flat      : flat fading or not
% (1->flat (only amplitude is fluctuated),
% 0->nomal(phase and amplitude are fluctutated)
flat =1;

%***** START CALCULATION *****
nloop=1000; % Number of simulation loops
noe = 0;    % Number of error data
nod = 0;    % Number of transmitted data

for iii=1:nloop

%***** Data generation *****
    data=rand(1,nd)>0.5; % rand: built in function

%***** BPSK Modulation *****
    data1=data.*2-1;
    [data2] = oversamp( data1, nd , IPOINT) ;
    data3 = conv(data2,xh); % conv: built in function

%***** Attenuation Calculation *****
    spow=sum(data3.*data3)/nd;
    attn=0.5*spow*sr/br*10.^(-ebn0/10);
    attn=sqrt(attn);

%***** Fading channel *****
    % Generated data are fed into a fading simulator
    % In the case of BPSK, only Ich data are fed into fading counter
    [ifade,qfade]=sefade(data3,zeros(1,length(data3)),itau,dlvl,th1,n0,...
        itnd1,nowl,length(data3),tstp,fd,flat);
    % Updata fading counter
    itnd1 = itnd1+ itnd0;

%***** Add White Gaussian Noise (AWGN) *****
    inoise=randn(1,length(ifade)).*attn; % randn: built in function
    data4=ifade+inoise;
    data5=conv(data4,xh2); % conv: built in function

    sampl=irfn*IPOINT+1;
    data6 = data5(sampl:8:8*nd+sampl-1);

%***** BPSK Demodulation *****
    demodata=data6 > 0;

%***** Bit Error Rate (BER) *****
    % count number of instantaneous errors
    noe2=sum(abs(data-demodata)); % sum: built in function

    % count number of instantaneous transmitted data
    nod2=length(data); % length: built in function

    noe=noe+noe2;
    nod=nod+nod2;

    %fprintf('%d\t%e\n',iii,noe2/nod2);

```

```

end % for iii=1:nloop

%***** Output result *****
ber = noe/nod;
fprintf('%d\t%d\t%d\t%e\n', ebn0, noe, nod, noe/nod);
%fid = fopen('BERbpskfad.dat', 'a');
%fprintf(fid, '%d\t%e\t%f\t%f\t\n', ebn0, noe/nod, noe, nod);
%fclose(fid);

```

3.3.2. ОЦЕНКА ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ QPSK

Скрипт 3.3.3. Скрипт для оценки помехоустойчивости сигналов QPSK в канале с АБГШ

```

% Program 3-5
% qpsk.m
% Simulation program to realize QPSK transmission system

%***** Preparation part *****
sr=256000.0; % Symbol rate
ml=2; % ml: Number of modulation levels
(BPSK:ml=1, QPSK:ml=2, 16QAM:ml=4)
br=sr .* ml; % Bit rate
nd = 1000; % Number of symbols that simulates in each loop
ebn0=3; % Eb/N0
IPOINT=8; % Number of oversamples

%***** Filter initialization *****
irfn=21; % Number of taps
alfs=0.5; % Rolloff factor
[xh] = hrollfcoef(irfn, IPOINT, sr, alfs, 1); %Transmitter filter coefficients
[xh2] = hrollfcoef(irfn, IPOINT, sr, alfs, 0); %Receiver filter coefficients

%***** START CALCULATION *****
nloop=100; % Number of simulation loops
noe = 0; % Number of error data
nod = 0; % Number of transmitted data

for iii=1:nloop
%***** Data generation *****
data1=rand(1,nd*ml)>0.5; % rand: built in function

%***** QPSK Modulation *****
[ich, qch]=qpskmod(data1, 1, nd, ml);
[ich1, qch1]= compoversamp(ich, qch, length(ich), IPOINT);
[ich2, qch2]= compconv(ich1, qch1, xh);

%***** Attenuation Calculation *****
spow=sum(ich2.*ich2+qch2.*qch2)/nd; % sum: built in function
attn=0.5*spow*sr/br*10.^(-ebn0/10);
attn=sqrt(attn); % sqrt: built in function

%***** Add White Gaussian Noise (AWGN) *****
[ich3, qch3]= comb(ich2, qch2, attn); % add white gaussian noise
[ich4, qch4]= compconv(ich3, qch3, xh2);

```

```

syncpoint=irfn*IPOINT+1;
ich5=ich4(syncpoint:IPOINT:length(ich4));
qch5=qch4(syncpoint:IPOINT:length(qch4));

%***** QPSK Demodulation
%*****
[demodata]=qpskdemod(ich5,qch5,1,nd,ml);

%***** Bit Error Rate (BER)
%*****
noe2=sum(abs(data1-demodata)); % sum: built in function
nod2=length(data1); % length: built in function
noe=noe+noe2;
nod=nod+nod2;
end % for iii=1:nloop

%***** Output result *****
ber = noe/nod;
fprintf('%d\t%d\t%d\t%e\n',ebn0,noe,nod,noe/nod);

```

Скрипт 3.3.4. Скрипт для оценки помехоустойчивости сигналов QPSK в канале с замираниями

```

% Program 3-6
% qpsk_fading.m
% Simulation program to realize QPSK transmission system
% (under one path fading)

%***** Preparation part *****

sr=256000.0; % Symbol rate
ml=2; % ml: Number of modulation levels
(BPSK:ml=1,QPSK:ml=2,16QAM:ml=4)
br=sr .* ml; % Bit rate
nd = 100; % Number of symbols that simulates in each loop
ebn0=10; % Eb/N0
IPOINT=8; % Number of oversamples

%***** Filter initialization *****

irfn=21; % Number of taps
alfs=0.5; % Rolloff factor
[xh] = hrollfcoef(irfn,IPOINT,sr,alfs,1); %Transmitter filter coefficients
[xh2] = hrollfcoef(irfn,IPOINT,sr,alfs,0); %Receiver filter coefficients

%***** Fading initialization *****
% If you use fading function "sefade", you can initialize all of parameters.
% Otherwise you can comment out the following initialization.

% Time resolution
tstp=1/sr/IPOINT;

% Arrival time for each multipath normalized by tstp
% If you would like to simulate under one path fading model,
% you have only to set direct wave.
itau = [0];

```

```

% Mean power for each multipath normalized by direct wave.
% If you would like to simulate under one path fading model,
% you have only to set direct wave.
dlvl = [0];

% Number of waves to generate fading for each multipath.
% In normal case, more than six waves are needed to generate Rayleigh fading
n0=[6];

% Initial Phase of delayed wave
% In this simulation four-path Rayleigh fading are considered.
th1=[0.0];

% Number of fading counter to skip
itnd0=nd*IPOINT*100;

% Initial value of fading counter
% In this simulation one-path Rayleigh fading are considered.
% Therefore one fading counter are needed.
itnd1=[1000];

% Number of directwave + Number of delayed wave
% In this simulation one-path Rayleigh fading are considered
now1=1;

% Maximum Doppler frequency [Hz]
fd=160;

% You can decide two mode to simulate fading by changing the variable flat
% flat      : flat fading or not
% 1->flat (only amplitude is fluctuated)
% 0->nomal(phase and amplitude are fluctuated)
flat =1;

%***** START CALCULATION *****
nloop=1000; % Number of simulation loops
noe = 0;    % Number of error data
nod = 0;    % Number of transmitted data

for iii=1:nloop
%***** Data generation *****
    data1=rand(1,nd*ml)>0.5; % rand: built in function

%***** QPSK Modulation *****
    [ich,qch]=qpskmod(data1,1,nd,ml);
    [ich1,qch1]= compoversamp(ich,qch,length(ich),IPOINT);
    [ich2,qch2]= compconv(ich1,qch1,xh);

%***** Attenuation Calculation *****
    spow=sum(ich2.*ich2+qch2.*qch2)/nd; % sum: built in function
    attn=0.5*spow*sr/br*10.^(-ebn0/10);
    attn=sqrt(attn); % sqrt: built in function

%***** Fading channel *****

    % Generated data are fed into a fading simulator
    [ifade,qfade]=sefade(ich2,qch2,itau,dlvl,th1,n0,...

```

```

itnd1,now1,length(ich2),tstp,fd,flat);

% Update fading counter
itnd1 = itnd1+ itnd0;

%***** Add White Gaussian Noise (AWGN) *****
[ich3,qch3]= comb(ifade,qfade,attn);% add white gaussian noise
[ich4,qch4]= compconv(ich3,qch3,xh2);

syncpoint=irfn*IPOINT+1;
ich5=ich4(syncpoint:IPOINT:length(ich4));
qch5=qch4(syncpoint:IPOINT:length(qch4));

%***** QPSK Demodulation *****
[demodata]=qpskdemod(ich5,qch5,1,nd,ml);

%***** Bit Error Rate (BER) *****
noe2=sum(abs(data1-demodata)); % sum: built in function
nod2=length(data1); % length: built in function
noe=noe+noe2;
nod=nod+nod2;
end % for iii=1:nloop
%***** Output result *****
ber = noe/nod;
fprintf('%d\t%d\t%d\t%e\n',ebn0,noe,nod,noe/nod);
%fid = fopen('BERqpskfad.dat','a');
%fprintf(fid,'%d\t%e\t%f\t%f\t\n',ebn0,noe/nod,noe,nod);
%fclose(fid);

```

3.3.3. ОЦЕНКА ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ OQPSK

Скрипт 3.3.5. Скрипт для оценки помехоустойчивости сигналов OQPSK в канале с АБГШ

```

% oqpsk.m
%
% Simulation program to realize OQPSK transmission system
%***** Preparation part *****

sr=256000.0; % Symbol rate
ml=2; % ml:Number of modulation levels (BPSK:ml=1, QPSK:ml=2,
16QAM:ml=4)
br=sr.*ml; % bit rate
nd = 1000; % Number of symbols that simulates in each loop
ebn0=3; % Eb/N0
IPOINT=8; % Number of oversamples

%***** Filter initialization *****

irfn=21; % Number of taps
alfs=0.5; % Rolloff factor
[xh] = hrollfcoef(irfn,IPOINT,sr,alfs,1); %Transmitter filter coefficients
[xh2] = hrollfcoef(irfn,IPOINT,sr,alfs,0); %Receiver filter coefficients

%***** START CALCULATION *****
*****

nloop=100; % Number of simulation loops

```

```

noe = 0;      % Number of error data
nod = 0;      % Number of transmitted data

for iii=1:nloop

%***** Data generation
%*****

    data1=rand(1,nd*ml)>0.5; % rand: built in function

%***** OQPSK Modulation
%*****

    [ich,qch]=qpskmod(data1,1,nd,ml);
    [ich1,qch1]=compoversamp(ich,qch,length(ich),IPOINT);
    ich21=[ich1 zeros(1,IPOINT/2)];
    qch21=[zeros(1,IPOINT/2) qch1];
    [ich2, qch2]=compconv(ich21,qch21,xh);

%***** Attenuation Calculation
%*****

    spow=sum(ich2.*ich2+qch2.*qch2)/nd; % sum: built in function
    attn=0.5*spow*sr/br*10.^(-ebn0/10);
    attn=sqrt(attn); % sqrt: built in function

%***** Fading channel *****

    % Generated data are fed into a fading simulator
    % [ifade,qfade]=se-
fade(ich2,qch2,itau,dlv1,th1,n0,itnd1,now1,length(ich1),tstp,fd,flat);

    % Update fading counter
    %itnd1 = itnd1+ itnd0;

%***** Add White Gaussian Noise (AWGN)
%*****

    [ich3,qch3]= comb(ich2,qch2,attn);% add white gaussian noise
    [ich4,qch4]= compconv(ich3,qch3,xh2);

    syncpoint=irfn*IPOINT+1;
    ich5=ich4(syncpoint:IPOINT:length(ich4));
    qch5=qch4(syncpoint+IPOINT/2:IPOINT:length(qch4));

%***** OQPSK Demodulation
%*****

    [demodata]=qpskdemod(ich5,qch5,1,nd,ml);

%***** Bit Error Rate (BER)
%*****

    noe2=sum(abs(data1-demodata)); % sum: built in function
    nod2=length(data1); % length: built in function

```

```

noe=noe+noe2;
nod=nod+nod2;

fprintf('%d\t%e\n',iii,noe2/nod2); % fprintf: built in function

end % for iii=1:nloop

%***** Output result *****

ber = noe/nod;
fprintf('%d\t%d\t%d\t%e\n',ebn0,noe,nod,noe/nod); % fprintf: built in func-
tion
fid = fopen('BERoqpsk.dat','a');
fprintf(fid,'%d\t%e\t%f\t%f\t\n',ebn0,noe/nod,noe,nod); % fprintf: built in
function
fclose(fid);

```

Скрипт 3.3.6. Скрипт для оценки помехоустойчивости сигналов OQPSK в канале с замираниями

```

% Program 3-12
% oqpsk_fading.m
%
% Simulation program to realize OQPSK transmission system
% (under one path fading)
%***** Preparation part *****

sr=256000.0; % Symbol rate
ml=2; % ml:Number of modulation levels (BPSK:ml=1, QPSK:ml=2,
16QAM:ml=4)
br=sr .* ml; % Bit rate
nd = 1000; % Number of symbols that simulates in each loop
ebn0=10; % Eb/N0
IPOINT=8; % Number of oversamples

%***** Filter initialization *****

irfn=21; % Number of taps
alfs=0.5; % Rolloff factor
[xh] = hrollfcoef(irfn,IPOINT,sr,alfs,1); %Transmitter filter coefficients
[xh2] = hrollfcoef(irfn,IPOINT,sr,alfs,0); %Receiver filter coefficients

%***** Fading initialization *****
% If you use fading function "sefade", you can initialize all of parameters.
% Otherwise you can comment out the following initialization.
% The detailed explanation of all of variables are mentioned in Program 2-8.

% Time resolution

tstp=1/sr/IPOINT;

% Arrival time for each multipath normalized by tstp
% If you would like to simulate under one path fading model, you have only
to set
% direct wave.

itau = [0];

```

```

% Mean power for each multipath normalized by direct wave.
% If you would like to simulate under one path fading model, you have only
to set
% direct wave.
dlv1 = [0];

% Number of waves to generate fading for each multipath.
% In normal case, more than six waves are needed to generate Rayleigh fading
n0=[6];

% Initial Phase of delayed wave
% In this simulation four-path Rayleigh fading are considered.
th1=[0.0];

% Number of fading counter to skip
itnd0=nd*IPOINT;

% Initial value of fading counter
% In this simulation one-path Rayleigh fading are considered.
% Therefore one fading counter are needed.

itnd1=[1000];

% Number of directwave + Number of delayed wave
% In this simulation one-path Rayleigh fading are considered
now1=1;

% Maximum Doppler frequency [Hz]
% You can insert your favorite value
fd=160;

% You can decide two mode to simulate fading by changing the variable flat
% flat      : flat fading or not
% (1->flat (only amplitude is fluctuated),0->nomal(phase and amplitude are
fluctutated)
flat =1;

%***** START CALCULATION
%*****

nloop=200; % Number of simulation loops

noe = 0;    % Number of error data
nod = 0;    % Number of transmitted data

for iii=1:nloop

%***** Data generation
%*****

        data1=rand(1,nd*ml)>0.5; % rand: built in function

%***** OQPSK Modulation
%*****

```



```

[ich,qch]=qpskmod(data1,1,nd,ml);
[ich1,qch1]=compooversamp(ich,qch,length(ich),IPOINT);
ich21=[ich1 zeros(1,IPOINT/2)];
qch21=[zeros(1,IPOINT/2) qch1];
[ich2,qch2]=compcconv(ich21,qch21,xh);

%***** Attenuation Calculation
*****

spow=sum(ich2.*ich2+qch2.*qch2)/nd; % sum: built in function
attn=0.5*spow*sr/br*10.^(-ebn0/10);
attn=sqrt(attn); % sqrt: built in function

%***** Fading channel *****

% Generated data are fed into a fading simulator
[ifade,qfade]=se-
fade(ich2,qch2,itau,dlv1,th1,n0,itnd1,now1,length(ich1),tstp,fd,flat);

% Update fading counter
itnd1 = itnd1+ itnd0;

%***** Add White Gaussian Noise (AWGN)
*****

[ich3,qch3]= comb(ifade,qfade,attn);% add white gaussian noise
[ich4,qch4]= compcconv(ich3,qch3,xh2);

syncpoint=irfn*IPOINT+1;
ich5=ich4(syncpoint:IPOINT:length(ich4));
qch5=qch4(syncpoint+IPOINT/2:IPOINT:length(qch4));

%***** OQPSK Demodulation
*****

[demodata]=qpskdemod(ich5,qch5,1,nd,ml);

%***** Bit Error Rate (BER)
*****

noe2=sum(abs(data1-demodata)); % sum: built in function
nod2=length(data1); % length: built in function
noe=noe+noe2;
nod=nod+nod2;

fprintf('%d\t%e\n',iii,noe2/nod2); % fprintf: built in function

end % for iii=1:nloop

%***** Output result *****

ber = noe/nod;
fprintf('%d\t%d\t%d\t%e\n',ebn0,noe,nod,noe/nod); % fprintf: built in func-
tion
fid = fopen('BERoqpskfad.dat','a');

```

```
fprintf(fid, '%d\t%e\t%f\t%f\t\n', ebn0, noe/nod, noe, nod); % fprintf: built in
function
fclose(fid);
```

3.3.4. ОЦЕНКА ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ QAM

Скрипт 3.3.7. Скрипт для оценки помехоустойчивости сигналов QAM в канале с АБГШ

```
% qam16
% Simulation program to realize 16QAM transmission system

%***** preparation part *****

sr=256000.0; % Symbol rate
ml=4; % ml:Number of modulation levels (BPSK:ml=1, QPSK:ml=2,
16QAM:ml=4)
br=sr .* ml; % Bit rate
nd = 1000; % Number of symbols that simulates in each loop
ebn0=6; % Eb/N0
IPOINT=8; % Number of oversamples

%***** Filter initialization *****

irfn=21; % Number of taps
alfs=0.5; % Rolloff factor
[xh] = hrollfcoef(irfn, IPOINT, sr, alfs, 1); %Transmitter filter coefficients
[xh2] = hrollfcoef(irfn, IPOINT, sr, alfs, 0); %Receiver filter coefficients

%***** START CALCULATION *****
*****

nloop=100; % Number of simulation loops

noe = 0; % Number of error data
nod = 0; % Number of transmitted data

for iii=1:nloop

%***** Data generation *****
*****

data1=rand(1, nd*ml)>0.5;

%***** 16QAM Modulation *****
*****

[ich, qch]=qammod(data1, 1, nd, ml);
[ich1, qch1]= compoversamp(ich, qch, length(ich), IPOINT);
[ich2, qch2]= compconv(ich1, qch1, xh);

%***** Attenuation Calculation *****
*****

spow=sum(ich2.*ich2+qch2.*qch2)/nd;
attn=0.5*spow*sr/br*10.^(-ebn0/10);
attn=sqrt(attn);
```

```

%***** Fading channel *****

% Generated data are fed into a fading simulator
% [ifade,qfade]=se-
fade(ich2,qch2,itau,dlvl,th1,n0,itnd1,now1,length(ich2),tstp,fd,flat);

% Update fading counter
% itnd1 = itnd1+ itnd0;

%***** Add White Gaussian Noise (AWGN)
*****

[ich3,qch3]= comb(ich2,qch2,attn);% add white gaussian noise
[ich4,qch4]= compconv(ich3,qch3,xh2);

saml=irfn*IPOINT+1;
ich5 = ich4(saml:IPOINT:length(ich4));
qch5 = qch4(saml:IPOINT:length(ich4));

%***** 16QAM Demodulation
*****

[demodata]=qamdemod(ich5,qch5,1,nd,ml);

%***** Bit Error Rate (BER) *****

noe2=sum(abs(data1-demodata));
nod2=length(data1);
noe=noe+noe2;
nod=nod+nod2;

fprintf('%d\t%e\n',iii,noe2/nod2);
end % for iii=1:nloop

%***** Output result *****

ber = noe/nod;
fprintf('%d\t%d\t%d\t%e\n',ebn0,noe,nod,noe/nod);
fid = fopen('BERqam.dat','a');
fprintf(fid,'%d\t%e\t%f\t%f\t\n',ebn0,noe/nod,noe,nod);
fclose(fid);

```

Скрипт 3.3.8. Скрипт для оценки помехоустойчивости сигналов QAM в канале с замираниями

```

% qam16_fading
% Simulation program to realize 16QAM transmission system
% (under one path fading)

%***** preparation part *****

sr=256000.0; % Symbol rate
ml=4; % ml: Number of modulation levels (BPSK:ml=1, QPSK:ml=2,
16QAM:ml=4)
br=sr .* ml; % Bit rate
nd = 100; % Number of symbols that simulates in each loop

```

```

ebn0=15;      % Eb/N0
IPOINT=8;    % Number of oversamples

%***** Filter initialization *****

irfn=21;      % Number of taps
alfs=0.5;    % Rolloff factor
[xh] = hrollfcoef(irfn,IPOINT,sr,alfs,1); %Transmitter filter coefficients
[xh2] = hrollfcoef(irfn,IPOINT,sr,alfs,0); %Receiver filter coefficients

%***** Fading initialization *****
% If you use fading function "sefade", you can initialize all of parameters.
% Otherwise you can comment out the following initialization.
% The detailed explanation of all of variables are mentioned in Program 2-8.

% Time resolution

tstp=1/sr/IPOINT;

% Arrival time for each multipath normalized by tstp
% If you would like to simulate under one path fading model, you have only
to set
% direct wave.

itau = [0];

% Mean power for each multipath normalized by direct wave.
% If you would like to simulate under one path fading model, you have only
to set
% direct wave.
dlvl = [0];

% Number of waves to generate fading for each multipath.
% In normal case, more than six waves are needed to generate Rayleigh fading
n0=[6];

% Initial Phase of delayed wave
% In this simulation four-path Rayleigh fading are considered.
th1=[0.0];

% Number of fading counter to skip
itnd0=nd*IPOINT*100;

% Initial value of fading counter
% In this simulation one-path Rayleigh fading are considered.
% Therefore one fading counter are needed.

itnd1=[1000];

% Number of directwave + Number of delayed wave
% In this simulation one-path Rayleigh fading are considered
now1=1;

% Maximum Doppler frequency [Hz]
% You can insert your favorite value
fd=160;

```

```

% You can decide two mode to simulate fading by changing the variable flat
% flat      : flat fading or not
% (1->flat (only amplitude is fluctuated),0->nomal(phase and amplitude are
fluctutated)
flat =1;

%***** START CALCULATION
*****

nloop=1000; % Number of simulation loops

noe = 0;    % Number of error data
nod = 0;    % Number of transmitted data

for iii=1:nloop

%***** Data generation
*****

    data1=rand(1,nd*ml)>0.5;

%***** 16QAM Modulation
*****

    [ich,qch]=qammod(data1,1,nd,ml);
    [ich1,qch1]= compoversamp(ich,qch,length(ich),IPOINT);
    [ich2,qch2]= compconv(ich1,qch1,xh);

%***** Attenuation Calculation
*****

    spow=sum(ich2.*ich2+qch2.*qch2)/nd;
    attn=0.5*spow*sr/br*10.^(-ebn0/10);
    attn=sqrt(attn);

%***** Fading channel *****

    % Generated data are fed into a fading simulator
    [ifade,qfade,ramp]=se-
fade(ich2,qch2,itau,dlvl,th1,n0,itnd1,now1,length(ich2),tstp,fd,flat);

    % Updata fading counter
    itnd1 = itnd1+ itnd0;

%***** Add White Gaussian Noise (AWGN)
*****

    [ich3,qch3]= comb(ifade,qfade,attn);% add white gaussian noise

%***** Compensate the fluctuation of fading by
ramp*****

    ich3=ich3./ramp(1:length(ramp));
    qch3=qch3./ramp(1:length(ramp));

```

```

[ich4,qch4]= compconv(ich3,qch3,xh2);

saml=irfn*IPOINT+1;
ich5 = ich4(saml:IPOINT:length(ich4));
qch5 = qch4(saml:IPOINT:length(ich4));

%***** 16QAM Demodulation *****
%*****

[demodata]=qamdmod(ich5,qch5,1,nd,m1);

%***** Bit Error Rate (BER) *****

noe2=sum(abs(data1-demodata));
nod2=length(data1);
noe=noe+noe2;
nod=nod+nod2;

fprintf('%d\t%e\n',iii,noe2/nod2);
end % for iii=1:nloop

%***** Output result *****

ber = noe/nod;
fprintf('%d\t%d\t%d\t%e\n',ebn0,noe,nod,noe/nod);
fid = fopen('BERqamfad.dat','a');
fprintf(fid,'%d\t%e\t%f\t%f\t\n',ebn0,noe/nod,noe,nod);
fclose(fid);

```

4. ЦИФРОВЫЕ СИГНАЛЫ ЧАСТОТНОЙ МАНИПУЛЯЦИИ

4.1. ЛЕКЦИЯ. ЦИФРОВЫЕ СИГНАЛЫ ЧАСТОТНОЙ МАНИПУЛЯЦИИ

Цель занятия: изучить особенности передачи и приема цифровых сигналов частотной манипуляции FMSK, GMSK.

4.2. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ. РАЗРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ ДЛЯ ОЦЕНКИ ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ FMSK, GMSK В КАНАЛЕ С АБГШ И ЗАМИРАНИЯМИ

Цель занятия: разработать имитационную модель для оценки помехоустойчивости сигналов FMSK, GMSK в канале с АБГШ и замираниями.

Скрипт 4.2.1. Функция для реализации фильтра Гаусса

```
% gaussf.m
%
% Function to form Gaussaian filter

function [xh] = gaussf(B,irfn,ipoint,sr,ncc)

%*****
%   irfn      : Number of symbols to use filtering
%   ipoint    : Number of samples in one symbol
%   sr       : symbol rate
%   B        : filter coeficiense
%   ncc;     : 1 -- transmitting filter  0 -- receiving filter
%*****

point = ipoint;

tr = sr ;
n = ipoint .* irfn;
mid = ( n ./ 2 ) + 1;
fo=B/sqrt(2*log(2));

for i = 1 : n

    icon = i - mid;
    ym = icon;

    xt=1/2*(erf(-sqrt(2/log(2))*pi*B*(ym/ipoint-
1/2)/tr)+erf(sqrt(2/log(2))*pi*B*(ym/ipoint+1/2)/tr));

    if ncc == 0                                % in the case of receiver
        xh( i ) = xt ;
    elseif ncc == 1                            % in the case of transmitter
        xh( i ) = xt;
    else
```

```

    error('ncc error');
end
end

```

4.3. ЛАБОРАТОРНАЯ РАБОТА. ОЦЕНКА ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ FMSK, GMSK В КАНАЛЕ С ЗАМИРАНИЯМИ

Цель занятия: оценить помехоустойчивость сигналов FMSK, GMSK в канале с АБГШ и замираниями средствами имитационного моделирования.

4.3.1. ОЦЕНКА ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ FMSK

Скрипт 4.3.1. Скрипт для оценки помехоустойчивости сигналов FMSK в канале с АБГШ

```

% msk2.m
%
% Simulation program to realize MSK transmission system
%***** Preparation part *****

sr=256000.0; % Symbol rate
ml=1;       % ml:Number of modulation levels
br=sr.*ml;  % Bit rate
nd = 1000;  % Number of symbols that simulates in each loop
ebn0=5;     % Eb/N0
IPOINT=8;   % Number of oversamples

%***** START CALCULATION
%*****

nloop=100;  % Number of simulation loops

noe = 0;    % Number of error data
nod = 0;    % Number of transmitted data

for iii=1:nloop

%***** Data generation
%*****

    data1=rand(1,nd*ml)>0.5; % rand: built in function

%***** MSK Modulation *****

    data11=2*data1-1;
    data2=oversamp2(data11,length(data11),IPOINT);

    th=zeros(1,length(data2)+1);
    ich2=zeros(1,length(data2)+1);
    qch2=zeros(1,length(data2)+1);

    for ii=2:length(data2)+1
        th(1,ii)=th(1,ii-1)+pi/2*data2(1,ii-1)./IPOINT;
    end
end

```



```

    ich2=cos(th);
    qch2=sin(th);

%***** Attenuation Calculation *****
%*****

    spow=sum(ich2.*ich2+qch2.*qch2)/(nd*IPOINT); % sum: built in function
    attn=0.5*spow*sr/br*10.^(-ebn0/10);
    attn=sqrt(attn); % sqrt: built in function

%***** Fading channel *****
%[ifade,qfade]=se-
fade2(data2,qdata1,itau,dlvl1,th1,n0,itnd1,now1,length(data2),fftlen2,fstp,f
d,flat);

%***** Add White Gaussian Noise (AWGN) *****
%*****

    [ich3,qch3]= comb(ich2,qch2,attn);% add white gaussian noise

    syncpoint = 1;
    ich5=ich3(syncpoint:IPOINT:length(ich3));
    qch5=qch3(syncpoint:IPOINT:length(qch3));

%***** MSK Demodulation *****
%*****

    demodata2(1,1)=-1;

    for k=3:2:nd*ml+1
        demodata2(1,k)=ich5(1,k)*qch5(1,k-1)*cos(pi*(k))>0;
    end

    for n=2:2:nd*ml+1
        demodata2(1,n)=ich5(1,n-1)*qch5(1,n)*cos(pi*(n))>0;
    end

    [demodata]=demodata2(1,2:nd*ml+1);

%***** Bit Error Rate (BER) *****
%*****

    noe2=sum(abs(data1-demodata)); % sum: built in function
    nod2=length(data1); % length: built in function
    noe=noe+noe2;
    nod=nod+nod2;

    fprintf('%d\t%e\n',iii,noe2/nod2); % fprintf: built in function

end % for iii=1:nloop

%***** Data file *****
%*****

ber = noe/nod;

```

```

fprintf('%d\t%d\t%d\t%e\n', ebn0, noe, nod, noe/nod); % fprintf: built in function
fid = fopen('BERmsk2.dat', 'a');
fprintf(fid, '%d\t%e\t%f\t%f\t\n', ebn0, noe/nod, noe, nod); % fprintf: built in function
fclose(fid);

```

Скрипт 4.3.2. Скрипт для оценки помехоустойчивости сигналов FMSK в канале с замираниями

```

% msk2_fading.m
%
% Simulation program to realize MSK transmission system
% (under one path fading)

%***** Preparation part *****

sr=256000.0; % Symbol rate
ml=1; % ml: Number of modulation levels
br=sr.*ml; % Bit rate
nd = 100; % Number of symbols that simulates in each loop
ebn0=15; % Eb/N0
IPOINT=8; % Number of oversamples

%***** Fading initialization *****
% If you use fading function "sefade", you can initialize all of parameters.
% Otherwise you can comment out the following initialization.
% The detailed explanation of all of variables are mentioned in Program 2-8.

% Time resolution

tstp=1/sr/IPOINT;

% Arrival time for each multipath normalized by tstp
% If you would like to simulate under one path fading model, you have only
to set
% direct wave.

itau = [0];

% Mean power for each multipath normalized by direct wave.
% If you would like to simulate under one path fading model, you have only
to set
% direct wave.
dlvl = [0];

% Number of waves to generate fading for each multipath.
% In normal case, more than six waves are needed to generate Rayleigh fading
n0=[6];

% Initial Phase of delayed wave
% In this simulation four-path Rayleigh fading are considered.
th1=[0.0];

% Number of fading counter to skip
itnd0=nd*IPOINT*100;

% Initial value of fading counter

```

```

% In this simulation one-path Rayleigh fading are considered.
% Therefore one fading counter are needed.

itnd1=[3000];

% Number of directwave + Number of delayed wave
% In this simulation one-path Rayleigh fading are considered
now1=1;

% Maximum Doppler frequency [Hz]
% You can insert your favorite value
fd=320;

% You can decide two mode to simulate fading by changing the variable flat
% flat      : flat fading or not
% (1->flat (only amplitude is fluctuated),0->nomal(phase and amplitude are
fluctutated)
flat =1;

%***** START CALCULATION
%*****

nloop=1000; % Number of simulation loops

noe = 0; % Number of error data
nod = 0; % Number of transmitted data

for iii=1:nloop

%***** Data generation
%*****

    data1=rand(1,nd*ml)>0.5; % rand: built in function

%***** MSK Modulation *****

    data11=2*data1-1;
    data2=oversamp2(data11,length(data11),IPOINT);

    th=zeros(1,length(data2)+1);
    ich2=zeros(1,length(data2)+1);
    qch2=zeros(1,length(data2)+1);

    for ii=2:length(data2)+1
        th(1,ii)=th(1,ii-1)+pi/2*data2(1,ii-1)./IPOINT;
    end

    ich2=cos(th);
    qch2=sin(th);

%***** Attenuation Calculation
%*****

    spow=sum(ich2.*ich2+qch2.*qch2)/(nd*IPOINT); % sum: built in function
    attn=0.5*spow*sr/br*10.^(-ebn0/10);
    attn=sqrt(attn); % sqrt: built in function

```

```

%***** Fading channel *****

% Generated data are fed into a fading simulator
[ifade,qfade]=se-
fade(ich2,qch2,itau,dlvl,th1,n0,itnd1,now1,length(ich2),tstp,fd,flat);

% Updata fading counter
itnd1 = itnd1+ itnd0;

%***** Add White Gaussian Noise (AWGN)
*****
[ich3,qch3]= comb(ifade,qfade,attn);% add white gaussian noise

syncpoint = 1;
ich5=ich3(syncpoint:IPOINT:length(ich3));
qch5=qch3(syncpoint:IPOINT:length(qch3));

%***** MSK Demodulation *****

demodata2(1,1)=-1;

for k=3:2:nd*ml+1
    demodata2(1,k)=ich5(1,k)*qch5(1,k-1)*cos(pi*(k))>0;
end

for n=2:2:nd*ml+1
    demodata2(1,n)=ich5(1,n-1)*qch5(1,n)*cos(pi*(n))>0;
end

[demodata]=demodata2(1,2:nd*ml+1);

%***** Bit Error Rate (BER)
*****

noe2=sum(abs(data1-demodata)); % sum: built in function
nod2=length(data1); % length: built in function
noe=noe+noe2;
nod=nod+nod2;

fprintf('%d\t%e\n',iii,noe2/nod2); % fprintf: built in function

end % for iii=1:nloop

%***** Data file
*****

ber = noe/nod;
fprintf('%d\t%d\t%d\t%e\n',ebn0,noe,nod,noe/nod); % fprintf: built in func-
tion
fid = fopen('BERmsk2fad.dat','a');
fprintf(fid,'%d\t%e\t%f\t%f\t\n',ebn0,noe/nod,noe,nod); % fprintf: built in
function
fclose(fid);

```

4.3.2. ОЦЕНКА ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ GMSK

Скрипт 4.3.3. Скрипт для оценки помехоустойчивости сигналов GMSK в канале с АБГШ

```
% gmsk.m
%
% Simulation program to realize GMSK transmission system
%
%***** Preparation part *****

sr=256000.0; % Symbol rate
ml=1;       % ml: Number of modulation levels
br=sr.*ml;  % Bit rate
nd = 1000;  % Number of symbols that simulates in each loop
ebn0=5;     % Eb/N0
IPOINT=8;   % Number of oversamples

%***** Filter initialization *****

irfn=21;    % Number of taps
B=0.25*sr;
B2=0.6*sr;
[xh] = gaussf(B,irfn,IPOINT,sr,1); %Transmitter filter coefficients
[xh2] =gaussf(B2,irfn,IPOINT,sr,0); %Receiver filter coefficients

%***** START CALCULATION *****
*****

nloop=100;  % Number of simulation loops

noe = 0;    % Number of error data
nod = 0;    % Number of transmitted data

for iii=1:nloop

%***** Data generation *****
*****

    data1=rand(1,nd.*ml)>0.5; % rand: built in function

%***** GMSK Modulation *****
*****

    data11=2*data1-1;
    data2=oversamp(data11,length(data11),IPOINT);
    data3=conv(data2,xh); % NEW for GMSK

    th=zeros(1,length(data3)+1);
    ich2=zeros(1,length(data3)+1);
    qch2=zeros(1,length(data3)+1);

    for ii=2:length(data3)+1
        th(1,ii)=th(1,ii-1)+pi/2*data3(1,ii-1)./IPOINT;
    end

    ich2=cos(th);
```

```

qch2=sin(th);

%***** Attenuation Calculation *****

spow=sum(ich2.*ich2+qch2.*qch2)/nd; % sum: built in function
attn=0.5*spow*sr/br*10.^(-ebn0/10);
attn=sqrt(attn); % sqrt: built in function

%***** Fading channel *****

%[ifade,qfade]=se-
fade2(data2,qdata1,itau,dlvl1,th1,n0,itnd1,now1,length(data2),fftlen2,fstp,fd,flat);

%***** Add White Gaussian Noise (AWGN) *****

[ich3,qch3]= comb(ich2,qch2,attn);% add white gaussian noise

[ich4,qch4] = compconv(ich3,qch3,xh2);

syncpoint =irfn*IPOINT-IPOINT/2+1;
ich5=ich4(syncpoint:IPOINT:length(ich4));
qch5=qch4(syncpoint:IPOINT:length(qch4));

%***** GMSK Demodulation *****

demodata2(1,1)=-1;

for k=3:2:nd*ml+1
    demodata2(1,k)=ich5(1,k)*qch5(1,k-1)*cos(pi*(k))>0;
end

for n=2:2:nd*ml+1
    demodata2(1,n)=ich5(1,n-1)*qch5(1,n)*cos(pi*(n))>0;
end

[demodata]=demodata2(1,2:nd*ml+1);

%***** Bit Error Rate (BER) *****

noe2=sum(abs(data1-demodata)); % sum: built in function
nod2=length(data1); % length: built in function
noe=noe+noe2;
nod=nod+nod2;

fprintf('%d\t%e\n',iii,noe2/nod2); % fprintf: built in function
end % for iii=1:nloop

%***** Output result *****

ber = noe/nod;

```

```

fprintf('%d\t%d\t%d\t%e\n', ebn0, noe, nod, noe/nod); % fprintf: built in function
fid = fopen('BERgmsk.dat', 'a');
fprintf(fid, '%d\t%e\t%f\t%f\t\n', ebn0, noe/nod, noe, nod); % fprintf: built in function
fclose(fid);

```

Скрипт 4.3.4. Скрипт для оценки помехоустойчивости сигналов GMSK в канале с замираниями

```

% gmsk_fading.m
%
% Simulation program to realize GMSK transmission system
% (under one path fading)

%***** Preparation part *****

sr=256000.0; % Symbol rate
ml=1; % ml: Number of modulation levels
br=sr.*ml; % Bit rate
nd = 100; % Number of symbols that simulates in each loop
ebn0=15; % Eb/N0
IPOINT=8; % Number of oversamples

%***** Filter initialization *****

irfn=21; % Number of taps
B=0.25*sr;
B2=0.6*sr;
[xh] = gaussf(B, irfn, IPOINT, sr, 1); %Transmitter filter coefficients
[xh2] =gaussf(B2, irfn, IPOINT, sr, 0); %Receiver filter coefficients

%***** Fading initialization *****
% If you use fading function "sefade", you can initialize all of parameters.
% Otherwise you can comment out the following initialization.
% The detailed explanation of all of variables are mentioned in Program 2-8.

% Time resolution

tstp=1/sr/IPOINT;

% Arrival time for each multipath normalized by tstp
% If you would like to simulate under one path fading model, you have only
to set
% direct wave.

itau = [0];

% Mean power for each multipath normalized by direct wave.
% If you would like to simulate under one path fading model, you have only
to set
% direct wave.
dlvl = [0];

% Number of waves to generate fading for each multipath.
% In normal case, more than six waves are needed to generate Rayleigh fading
n0=[6];

```

```

% Initial Phase of delayed wave
% In this simulation four-path Rayleigh fading are considered.
th1=[0.0];

% Number of fading counter to skip
itnd0=nd*IPOINT*100;

% Initial value of fading counter
% In this simulation one-path Rayleigh fading are considered.
% Therefore one fading counter are needed.

itnd1=[1000];

% Number of directwave + Number of delayed wave
% In this simulation one-path Rayleigh fading are considered
now1=1;

% Maximum Doppler frequency [Hz]
% You can insert your favorite value
fd=320;

% You can decide two mode to simulate fading by changing the variable flat
% flat      : flat fading or not
% (1->flat (only amplitude is fluctuated),0->nomal(phase and amplitude are
fluctutated)
flat =1;

%***** START CALCULATION
%*****

nloop=1000; % Number of simulation loops

noe = 0;    % Number of error data
nod = 0;    % Number of transmitted data

for iii=1:nloop

%***** Data generation
%*****

    data1=rand(1,nd.*m1)>0.5; % rand: built in function

%***** GMSK Modulation
%*****

    data11=2*data1-1;
    data2=oversamp(data11,length(data11),IPOINT);
    data3=conv(data2,xh); % NEW for GMSK

    th=zeros(1,length(data3)+1);
    ich2=zeros(1,length(data3)+1);
    qch2=zeros(1,length(data3)+1);

    for ii=2:length(data3)+1
        th(1,ii)=th(1,ii-1)+pi/2*data3(1,ii-1)./IPOINT;
    end

```



```

    ich2=cos(th);
    qch2=sin(th);

%***** Attenuation Calculation *****

    spow=sum(ich2.*ich2+qch2.*qch2)/nd; % sum: built in function
    attn=0.5*spow*sr/br*10.^(-ebn0/10);
    attn=sqrt(attn); % sqrt: built in function

%***** Fading channel *****

    % Generated data are fed into a fading simulator
    [ifade,qfade]=se-
fade(ich2,qch2,itau,dlv1,th1,n0,itnd1,now1,length(ich2),tstp,fd,flat);

    % Update fading counter
    itnd1 = itnd1+ itnd0;

%***** Add White Gaussian Noise (AWGN) *****

    [ich3,qch3]= comb(ifade,qfade,attn);% add white gaussian noise

    [ich4,qch4] = compconv(ich3,qch3,xh2);

    syncpoint =irfn*IPOINT-IPOINT/2+1;
    ich5=ich4(syncpoint:IPOINT:length(ich4));
    qch5=qch4(syncpoint:IPOINT:length(qch4));

%***** GMSK Demodulation *****

    demodata2(1,1)=-1;

    for k=3:2:nd*ml+1
        demodata2(1,k)=ich5(1,k)*qch5(1,k-1)*cos(pi*(k))>0;
    end

    for n=2:2:nd*ml+1
        demodata2(1,n)=ich5(1,n-1)*qch5(1,n)*cos(pi*(n))>0;
    end

    [demodata]=demodata2(1,2:nd*ml+1);

%***** Bit Error Rate (BER) *****

    noe2=sum(abs(data1-demodata)); % sum: built in function
    nod2=length(data1); % length: built in function
    noe=noe+noe2;
    nod=nod+nod2;

    fprintf('%d\t%e\n',iii,noe2/nod2); % fprintf: built in function

end % for iii=1:nloop

```

```

%***** Output result *****

ber = noe/nod;
fprintf('%d\t%d\t%d\t%e\n', ebn0, noe, nod, noe/nod); % fprintf: built in function
fid = fopen('BERgmskfad.dat', 'a');
fprintf(fid, '%d\t%e\t%f\t%f\t\n', ebn0, noe/nod, noe, nod); % fprintf: built in function
fclose(fid);

```

Скрипт 4.3.5. Функция для реализации фильтра Гаусса

```

% gaussf.m
%
% Function to form Gaussaian filter

function [xh] = gaussf(B, irfn, ipoint, sr, ncc)

%*****
%   irfn      : Number of symbols to use filtering
%   ipoint    : Number of samples in one symbol
%   sr       : symbol rate
%   B        : filter coeficiense
%   ncc;     : 1 -- transmitting filter  0 -- receiving filter
%*****

point = ipoint;

tr = sr ;
n = ipoint .* irfn;
mid = ( n ./ 2 ) + 1;
fo=B/sqrt(2*log(2));

for i = 1 : n

    icon = i - mid;
    ym = icon;

    xt=1/2*(erf(-sqrt(2/log(2))*pi*B*(ym/ipoint-1/2)/tr)+erf(sqrt(2/log(2))*pi*B*(ym/ipoint+1/2)/tr));

    if ncc == 0 % in the case of receiver
        xh( i ) = xt ;
    elseif ncc == 1 % in the case of transmitter
        xh( i ) = xt;
    else
        error('ncc error');
    end
end
end

```

5. СИГНАЛЫ С РАСШИРЕННЫМ СПЕКТРОМ

5.1. ЛЕКЦИЯ. СИГНАЛЫ С РАСШИРЕННЫМ СПЕКТРОМ

Цель занятия: изучить особенности передачи и приема сигналов с расширенным спектром DSSS.

5.2. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ. РАЗРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ ДЛЯ ОЦЕНКИ ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ DSSS В КАНАЛЕ С АБГШ И ЗАМИРАНИЯМИ

Цель занятия: разработать имитационную модель для оценки помехоустойчивости сигналов DSSS в канале с АБГШ и замираниями.

Скрипт 5.2.1. Функция вычисления автокорреляции сигналов DSSS

```
% Program 5-1
% autocorr
% Autocorrelation function of a sequence

function [out] = autocorr(indata, tn)

% indata:   input sequence
% tn:       number of period
% out:      autocorrelation data

if nargin < 2
    tn = 1;
end

ln = length(indata);
out = zeros(1, ln*tn);

for ii = 0:ln*tn-1
    out(ii+1) = sum(indata.*shift(indata,ii,0));
end
```

Скрипт 5.2.2. Функция вычисления взаимной корреляции сигналов DSSS

```
% Program 5-2
% crosscorr.m
% Crosscorrelation function of a sequence

function [out] = crosscorr(indata1, indata2, tn)

% indata1:  input sequence1
% indata2:  input sequence2
% tn:       number of period
% out:      crosscorrelation data

if nargin < 3
    tn = 1;
end
```

```

ln = length(indata1);
out = zeros(1, ln*tn);

for ii = 0:ln*tn-1
    out(ii+1) = sum(indata1.*shift(indata2,ii,0));
end

```

Скрипт 5.2.3. Функция расширения спектра в сигнале DSSS при передаче

```

% Program 5-7
% spread.m
% Data spread function

function [iout, qout] = spread(idata, qdata, code1)

% idata:    ich data sequence
% qdata:    qch data sequence
% iout:     ich output data sequence
% qout:     qch output data sequence
% code1:    spread code sequence

switch nargin
    case {0, 1}
        error('lack of input argument');
    case 2
        code1 = qdata;
        qdata = idata;
end

[hn, vn] = size(idata);
[hc, vc] = size(code1);

if hn > hc
    error('lack of spread code sequences');
end

iout = zeros(hn, vn*vc);
qout = zeros(hn, vn*vc);

for ii = 1:hn
    iout(ii,:) = reshape(rot90(code1(ii,:),3)*idata(ii,:),1,vn*vc);
    qout(ii,:) = reshape(rot90(code1(ii,:),3)*qdata(ii,:),1,vn*vc);
end

```

Скрипт 5.2.4. Функция сужения спектра в сигнале DSSS при приеме

```

% Program 5-8
% despread.m
% Data despread function

function [iout, qout] = despread(idata, qdata, code1)

% idata:    ich data sequence
% qdata:    qch data sequence
% iout:     ich output data sequence
% qout:     qch output data sequence

```

```

% code1:    spread code sequence

switch nargin
    case {0, 1}
        error('lack of input arguments');
    case 2
        code1 = qdata;
        qdata = idata;
end

[hn, vn] = size(idata);
[hc, vc] = size(code1);

vn = fix(vn/vc); % rounds the elements of X to the nearest integers

iout = zeros(hc, vn);
qout = zeros(hc, vn);

for ii = 1:hc
    iout(ii,:) = rot90(flipud(rot90(re-
shape(idata(ii,:),vc,vn))*rot90(code1(ii,:),3)));
    qout(ii,:) = rot90(flipud(rot90(re-
shape(qdata(ii,:),vc,vn))*rot90(code1(ii,:),3)));
end

```

5.3. ЛАБОРАТОРНАЯ РАБОТА. ОЦЕНКА ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ DSSS В КАНАЛЕ С ЗАМИРАНИЯМИ

Цель занятия: оценить помехоустойчивость сигналов DSSS в канале с АБГШ и замираниями.

Скрипт 5.3.1. Скрипт для оценки помехоустойчивости сигналов DSSS в канале с АБГШ и замираниями

```

% Simulation program to realize DS-CDMA system
% dscdma.m

% Preparation part
sr = 256000.0;      % symbol rate
ml = 2;            % number of modulation levels
br = sr*ml;        % bit rate
nd = 100;          % number of symbol
ebn0 = 3;          % Eb/N0

% Filter initialization
irfn = 21;         % number of filter taps
IPOINT = 8;        % number of oversamples
alfs = 0.5;        % roll off factor
[xh] = hrollfcoef(irfn, IPOINT, sr, alfs, 1);
[xh2] = hrollfcoef(irfn, IPOINT, sr, alfs, 0);

% Spreading code initialization
user = 1;          % number of users
seq = 1;           % 1:M-sequence, 2:Gold, 3:Orthogonal Gold
stage = 3;         % number of stages
ptap1 = [1 3];    % position of taps for 1st

```

```

ptap2 = [2 3];      % position of taps for 2nd
regi1 = [1 1 1];   % initial value of register for 1st
regi2 = [1 1 1];   % initial value of register for 2nd

% Generation of spreading code
switch seq
    case 1
        code = mseq(stage, ptap1, regi1, user);      % Gold sequence
    case 2
        m1 = mseq(stage, ptap1, regi1);
        m2 = mseq(stage, ptap2, regi2);
        code = goldseq(m1, m2, user);                % Orthogonal Gold se-
quence
    case 3
        m1 = mseq(stage, ptap1, regi1);
        m2 = mseq(stage, ptap2, regi2);
        code = [goldseq(m1, m2, user), zeros(user,1)];
end
code = code*2-1;
clen = length(code);

% Fading initialization
rfade = 0;      % Rayleigh fading 0:nothing, 1:consider
itau = [0, 8]; % delay time
dlvl1 = [0.0, 40.0]; % attenuation level
n0 = [6, 7]; % number of waves to generate fading
th1 = [0.0, 0.0]; % initial phase of delayed wave
itnd1 = [3001, 4004]; % set fading counter
now1 = 2; % number of direct wave wave + delayed wave
tstp = 1/sr/IPOINT/clen; % time resolution
fd = 160; % doppler frequency [Hz]
flat = 1; % flat Rayleigh environment
itndel = nd*IPOINT*clen*30; % number of fading counter to skip

% Start calculation
nloop = 1000;
noe = 0;
nod = 0;

for ii = 1:nloop

    % Transmitter
    data = rand(user, nd*m1) >0.5;
    [ich, qch] = qpskmod(data, user, nd, m1); % QPSK modulation
    [ich1, qch1] = spread(ich, qch, code); % spreading
    [ich2, qch2] = compoversamp2(ich1, qch1, IPOINT); % oversampling
    [ich3, qch3] = compconv(ich2, qch2, xh); % filter

    if user == 1
        ich4 = ich3;
        qch4 = qch3;
    else
        ich4 = sum(ich3);
        qch4 = sum(qch3);
    end

    % Fading channel
    if rfade == 0

```

```

        ich5 = ich4;
        qch5 = qch4;
    else
        [ich5, qch5] = sefade(ich4, qch4, itau, dlvl1, th1, n0,...
            itnd1, now1, length(ich4), tstp, fd, flat);
        itnd1 = itnd1 + itndel;
    end

    % Receiver
    spow = sum(rot90(ich3.^2+qch3.^2))/nd;           % attenuation calcula-
tion
    attn = sqrt(0.5*spow*sr/br*10^(-ebn0/10));

    [ich6, qch6] = comb2(ich5, qch5, attn);         % add AWGN
    [ich7, qch7] = compconv2(ich6, qch6, xh2);     % filter

    sampl = irfn*IPOINT + 1;
    ich8 = ich7(:, sampl:IPOINT:IPOINT*nd*crlen+sampl-1);
    qch8 = qch7(:, sampl:IPOINT:IPOINT*nd*crlen+sampl-1);
    [ich9, qch9] = despread(ich8, qch8, code);     % despreading

    demodata = qpskdemod(ich9, qch9, user, nd, ml); % QPSK demodulation

    % BER
    noe2 = sum(sum(abs(data-demodata)));
    nod2 = user*nd*ml;
    noe = noe+noe2;
    nod = nod+nod2;
    % fprintf('%d\t%e\n', ii, noe2/nod2);
end

ber = noe/nod;
fprintf('%d\t%d\t%d\t%e\n', ebn0, noe, nod, noe/nod);

```

6. СИГНАЛЫ С ОРТОГОНАЛЬНЫМ ЧАСТОТНЫМ РАЗДЕЛЕНИЕМ

6.1. ЛЕКЦИЯ. СИГНАЛЫ С ОРТОГОНАЛЬНЫМ ЧАСТОТНЫМ РАЗДЕЛЕНИЕМ

Цель занятия: изучить особенности передачи и приема сигналов с ортогональным частотным разделением OFDM.

6.2. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ. РАЗРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ ДЛЯ ОЦЕНКИ ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ OFDM В КАНАЛЕ С АБГШ И ЗАМИРАНИЯМИ

Цель занятия: разработать имитационную модель для оценки помехоустойчивости сигналов OFDM в канале с АБГШ и замираниями.

Скрипт 6.2.1. Функция формирования выборок АБГШ

```
% Program 2-4
% comb.m
% Generate additive white gaussian noise

function [iout,qout] = comb (idata,qdata,attn)

%***** variables *****
% idata : input Ich data
% qdata : input Qch data
% iout  output Ich data
% qout  output Qch data
% attn : attenuation level caused by Eb/No or C/N
%*****

iout = randn(1,length(idata)).*attn;
qout = randn(1,length(qdata)).*attn;

iout = iout+idata(1:length(idata));
qout = qout+qdata(1:length(qdata));
```

Скрипт 6.2.2. Функция формирования защитного интервала в переданном сигнале OFDM

```
% Function 4-3
% giins.m
% Function to insert guard interval into transmission signal

function [iout,qout]= giins(idata,qdata,fftlen,gilen,nd);

%***** variables *****
% idata      : Input Ich data
% qdata      : Input Qch data
% iout       : Output Ich data
% qout       : Output Qch data
% fftlen     : Length of FFT (points)
% gilen      : Length of guard interval (points)
```



```

% *****

idata1=reshape(idata,fftlen,nd);
qdata1=reshape(qdata,fftlen,nd);
idata2=[idata1(fftlen-gilen+1:fftlen,:); idata1];
qdata2=[qdata1(fftlen-gilen+1:fftlen,:); qdata1];

iout=reshape(idata2,1,(fftlen+gilen)*nd);
qout=reshape(qdata2,1,(fftlen+gilen)*nd);

```

Скрипт 6.2.3. Функция исключения защитного интервала из принятого сигнала OFDM

```

% Function 4-4
% girem.m
% Function to remove guard interval from received signal

function [iout,qout]= girem(idata,qdata,fftlen2,gilen,nd);

%***** variables *****
% idata      : Input Ich data
% qdata      : Input Qch data
% iout       : Output Ich data
% qout       : Output Qch data
% fftlen2    : Length of FFT (points)
% gilen      : Length of guard interval (points)
% nd         : Number of OFDM symbols
% *****

idata2=reshape(idata,fftlen2,nd);
qdata2=reshape(qdata,fftlen2,nd);

iout=idata2(gilen+1:fftlen2,:);
qout=qdata2(gilen+1:fftlen2,:);

```

6.3. ЛАБОРАТОРНАЯ РАБОТА. ОЦЕНКА ПОМЕХОУСТОЙЧИВОСТИ СИГНАЛОВ OFDM В КАНАЛЕ С ЗАМИРАНИЯМИ

Цель занятия: выполнить оценку помехоустойчивости сигналов OFDM в канале с АБГШ и замираниями.

Скрипт 6.3.1. Скрипт для оценки помехоустойчивости сигналов OFDM в канале с АБГШ

```

% Program 4-1
% ofdm.m
% Simulation program to realize OFDM transmission system

%***** preparation part *****
para=128; % Number of parallel channel to transmit (points)
fftlen=128; % FFT length
noc=128; % Number of carrier
nd=6; % Number of information OFDM symbol for one loop
ml=2; % Modulation level : QPSK
sr=250000; % Symbol rate
br=sr.*ml; % Bit rate per carrier

```

```

gilen=32; % Length of guard interval (points)
ebn0=3; % Eb/N0

%***** main loop part *****
nloop=100; % Number of simulation loops
noe = 0; % Number of error data
nod = 0; % Number of transmitted data
eop=0; % Number of error packet
nop=0; % Number of transmitted packet

for iii=1:nloop
    %***** transmitter
    %***** Data generation *****
    seldata=rand(1,para*nd*ml)>0.5; % rand : built in function

    %***** Serial to parallel conversion
    paradata=reshape(seldata,para,nd*ml); % reshape : built in function

    %***** QPSK modulation
    [ich,qch]=qpskmod(paradata,para,nd,ml);
    kmod=1/sqrt(2); % sqrt : built in function
    ich1=ich.*kmod;
    qch1=qch.*kmod;

    %***** IFFT *****
    x=ich1+qch1.*i;
    y=ifft(x); % ifft : built in function
    ich2=real(y); % real : built in function
    qch2=imag(y); % imag : built in function

    %***** Gurad interval insertion *****
    [ich3,qch3]= giins(ich2,qch2,fftlen,gilen,nd);
    fftlen2=fftlen+gilen;

    %***** Attenuation Calculation *****
    spow=sum(ich3.^2+qch3.^2)/nd./para; % sum : built in function
    attn=0.5*spow*sr/br*10.^(-ebn0/10);
    attn=sqrt(attn);

    %***** Receiver *****
    %***** AWGN addition *****
    [ich4,qch4]=comb(ich3,qch3,attn);

    %***** Guard interval removal *****
    [ich5,qch5]= girem(ich4,qch4,fftlen2,gilen,nd);

    %***** FFT *****
    rx=ich5+qch5.*i;
    ry=fft(rx); % fft : built in function
    ich6=real(ry); % real : built in function
    qch6=imag(ry); % imag : built in function

    %***** demoduration *****
    ich7=ich6./kmod;

```

```

qch7=qch6./kmod;
[demodata]=qpskdemod(ich7,qch7,para,nd,ml);

%***** Parallel to serial conversion *****
demodata1=reshape(demodata,1,para*nd*ml);

%***** Bit Error Rate (BER) *****
% instantaneous number of error and data
noe2=sum(abs(demodata1-seldata)); % sum : built in function
nod2=length(seldata); % length : built in function

% cumulative the number of error and data in noe and nod
noe=noe+noe2;
nod=nod+nod2;

% calculating PER
if noe2~=0
    eop=eop+1;
else
    eop=eop;
end
eop;
nop=nop+1;
fprintf('%d\t%e\t%d\n',iii,noe2/nod2,eop); % fprintf : built in func-
tion
end

%***** Output result *****
per=eop/nop;
ber=noe/nod;

fprintf('%f\t%e\t%e\t%d\t\n',ebn0,ber,per,nloop);
%fid = fopen('BERofdm.dat','a');
fprintf(fid,'%f\t%e\t%e\t%d\t\n',ebn0,ber,per,nloop);
fclose(fid);

```

Скрипт 6.3.2. Скрипт для оценки помехоустойчивости сигналов OFDM в канале с замираниями

```

% Program 4-2
% ofdm_fading.m
% Simulation program to realize OFDM transmission system
% (under one path fading)

%***** preparation part *****
para=128; % Number of parallel channel to transmit (points)
fftlen=128; % FFT length
noc=128; % Number of carrier
nd=6; % Number of information OFDM symbol for one loop
ml=2; % Modulation level : QPSK
sr=250000; % Symbol rate
br=sr.*ml; % Bit rate per carrier
gilen=32; % Length of guard interval (points)
ebn0=10; % Eb/N0

%***** Fading initialization *****
% If you use fading function "sefade", you can initialize all of parameters.

```

```

% Otherwise you can comment out the following initialization.
% The detailed explanation of all of variables are mentioned in Program 2-8.

% Time resolution
tstp=1/sr/(fftlen+gilen);

% Arrival time for each multipath normalized by tstp
% If you would like to simulate under one path fading model,
% you have only to set direct wave.
itau = [0];

% Mean power for each multipath normalized by direct wave.
% If you would like to simulate under one path fading model,
% you have only to set direct wave.
dlvl = [0];

% Number of waves to generate fading for each multipath.
% In normal case, more than six waves are needed to generate Rayleigh fading
n0=[6];

% Initial Phase of delayed wave
% In this simulation four-path Rayleigh fading are considered.
th1=[0.0];

% Number of fading counter to skip
itnd0=nd*(fftlen+gilen)*10;

% Initial value of fading counter
% In this simulation one-path Rayleigh fading are considered.
% Therefore one fading counter are needed.
itnd1=[1000];

% Number of directwave + Number of delayed wave
% In this simulation one-path Rayleigh fading are considered
nowl=1;

% Maximum Doppler frequency [Hz]
fd=320;

% You can decide two mode to simulate fading by changing the variable flat
% flat      : flat fading or not
% 1->flat (only amplitude is fluctuated),
% 0->nomal(phase and amplitude are fluctuated)
flat =1;

%***** main loop part *****
nloop=500; % Number of simulation loops
noe = 0;   % Number of error data
nod = 0;   % Number of transmitted data
eop=0;     % Number of error packet
nop=0;     % Number of transmitted packet

for iii=1:nloop
    %***** transmitter
    %***** Data generation *****
    seldata=rand(1,para*nd*ml)>0.5; % rand : built in function

```

```

%***** Serial to parallel conversion
*****
paradata=reshape(seldata,para,nd*ml); % reshape : built in function

%***** QPSK modulation
*****
[ich,qch]=qpskmod(paradata,para,nd,ml);
kmod=1/sqrt(2); % sqrt : built in function
ich1=ich.*kmod;
qch1=qch.*kmod;

%***** IFFT *****
x=ich1+qch1.*i;
y=ifft(x); % ifft : built in function
ich2=real(y); % real : built in function
qch2=imag(y); % imag : built in function

%***** Guard interval insertion *****
[ich3,qch3]= giins(ich2,qch2,fftlen,gilen,nd);
fftlen2=fftlen+gilen;

%***** Attenuation Calculation *****
spow=sum(ich3.^2+qch3.^2)/nd./para; % sum : built in function
attn=0.5*spow*sr/br*10.^(-ebn0/10);
attn=sqrt(attn);

%***** Fading channel *****
% Generated data are fed into a fading simulator
[ifade,qfade]=se-
fade(ich3,qch3,itau,dlvl,th1,n0,itnd1,now1,length(ich3),tstp,fd,flat);

% Updata fading counter
itnd1 = itnd1+ itnd0;

%***** Receiver *****
%***** AWGN addition *****
[ich4,qch4]=comb(ifade,qfade,attn);

%***** Guard interval removal *****
[ich5,qch5]= girem(ich4,qch4,fftlen2,gilen,nd);

%***** FFT *****
rx=ich5+qch5.*i;
ry=fft(rx); % fft : built in function
ich6=real(ry); % real : built in function
qch6=imag(ry); % imag : built in function

%***** demoduration *****
ich7=ich6./kmod;
qch7=qch6./kmod;
[demodata]=qpskdemod(ich7,qch7,para,nd,ml);

%***** Parallel to serial conversion *****
demodata1=reshape(demodata,1,para*nd*ml);

```

```

%***** Bit Error Rate (BER)
*****
% instantaneous number of error and data
noe2=sum(abs(demodata1-seldata)); % sum : built in function
nod2=length(seldata); % length : built in function

% cumulative the number of error and data in noe and nod
noe=noe+noe2;
nod=nod+nod2;

% calculating PER
if noe2~=0
    eop=eop+1;
else
    eop=eop;
end
eop;
nop=nop+1;
fprintf('%d\t%e\t%d\n',iii,noe2/nod2,eop); % fprintf : built in func-
tion
end

%***** Output result *****
per=eop/nop;
ber=noe/nod;

fprintf('%f\t%e\t%e\t%d\t\n',ebn0,ber,per,nloop);
%fid = fopen('BERofdmfad.dat','a');
fprintf(fid,'%f\t%e\t%e\t%d\t\n',ebn0,ber,per,nloop);
fclose(fid);

```

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

2G	2 rd Generation	Второе поколение сетей сотовой связи, работающих в рамках стандарта GSM
3G	3 rd Generation	Третье поколение сетей сотовой связи, работающих в рамках стандарта UMTS
3GPP	3rd Generation Partnership Project	Партнерский проект ETSI, занимающийся стандартизацией в области систем 3G

СПИСОК ИСТОЧНИКОВ

1. Рекомендация МСЭ-R P.1407-6 (06/2017). Многолучевое распространение и параметризация его характеристик.
2. Рекомендация МСЭ-R P.1057-5 (12/2017). Распределения вероятностей, касающихся моделирования распространения радиоволн.
3. Галкин, В. А. Цифровая мобильная радиосвязь. Учебное пособие для вузов [Электронный ресурс] / В. А. Галкин. - М.: Горячая линия–Телеком, 2012. - 592 с.
4. Скляр, Бернард. Цифровая связь. Теоретические основы и практическое применение / [пер. с англ. Е. Г. Грозы и др.; под ред. А. В. Назаренко ; науч. консультант Е. В. Гусева]. - 2-е изд., испр. - М.: Вильямс, 2004. - 1104 с.
5. Бабков, В. Ю. Сотовые системы мобильной радиосвязи [Электронный ресурс]: учебное пособие/ В. Ю. Бабков, И. А. Цикин. - СПб.: БХВ-Петербург, 2013. - 432 с.
6. Narada H., Prasad R. Simulation and software radio for mobile communications. – Artech House, 2002.

Фокин Григорий Алексеевич

ФИЗИЧЕСКИЕ ОСНОВЫ СИСТЕМ МОБИЛЬНОЙ СВЯЗИ

Учебное пособие

Редактор *И. И. Щенсяк*
Компьютерная верстка

План 2020 г.

Подписано к печати
Объем 12,5 усл.-печ. л. Тираж 80 экз.

Редакционно-издательский центр СПбГУТ
191186 СПб., наб. р. Мойки, 61
Отпечатано в СПбГУТ