

Аннотация

В данной работе проводится анализ алгоритмов адаптивной компенсации нелинейных паразитных искажений, возникающих в приеме-передающих устройствах телекоммуникаций. Нелинейные искажения образуются в передатчиках базовых станций, а также мобильных устройств при прохождении сигнала через нелинейные цепи передающего тракта, усилитель мощности и дуплексер.

Вследствие компактности расположения компонент в RF-чипсете мобильного устройства, паразитная помеха распространяется из передатчика в приёмную часть уменьшая чувствительность приёмника.

Нелинейные искажения значительно влияют на качество радиочастотных сигналов, создавая помехи в канале связи. Такие нелинейные искажения приводят к увеличению битовой ошибки (Bit Error Rate - BER) на приемнике, генерируют внутриполосные и внеполосные помеховые сигналы и ухудшают качество передачи сигналов соседних полос. Для предотвращения искажений в современных базовых станциях и сотовых устройствах широко используются методы цифровой предискажения DPD, основанные на методе адаптивной компенсации нелинейных искажений.

В работе рассматриваются задачи адаптивной нелинейной компенсации искажений в задаче цифрового предискажения сигнала передатчика, а адаптивной компенсации паразитных помех в приёмном тракте мобильного терминала системы связи стандарта 4G/5G. Приводится математическое описание модели нелинейной помехи, а также используемых в исследовании алгоритмов.

Проводится исследование методов адаптивной компенсации нелинейных искажений, а также различных моделей нелинейных искажений радиосигналов.

Исследование алгоритмов адаптации, а также моделей нелинейных искажений, проводится на основе сигналов, сформированных на физической платформе. Сигнал помехи формируется относительно различных случаев выходной мощности усилителя.

Адаптивный корректор строится на основе классических нелинейных моделей, качественно описывающей процессы формирования нелинейных искажений, а также на основе нейросетевого подхода, позволяющего увеличить обобщающую способность адаптивной нелинейной модели. Особое внимание уделяется вопросам ускорения сходимости алгоритмов адаптации параметров нелинейной модели, а также методам сокращения ресурсов при аппаратной реализации модуля адаптивной коррекции.

Содержание

Аннотация	i
Введение	v
I.. Методы адаптивной компенсации нелинейных искажений сигнала в приеме-передающих устройствах систем связи	1
1. Постановка задачи адаптивной компенсации нелинейных искажений в устройствах связи	1
1.1. Проблема возникновения паразитных помех в приёмном тракте приема-передающего устройства	1
1.2. Схема компенсации паразитных нелинейных помех в приёмном тракте мобильного терминала	3
1.3. Проблема возникновения нелинейных искажений сигнала передающего тракта	4
1.4. Схема компенсации нелинейных искажений в передатчике приема-передающего устройства	5
1.5. Критерий компенсации нелинейных искажений в устройствах связи	6
2. Методы первого порядка для адаптации моделей нелинейных искажений	7
2.1. Метод градиентного спуска	8
2.2. Модификации метода градиентного спуска	10
3. Методы второго порядка для адаптации моделей нелинейных искажений	11
3.1. Метод Ньютона	11
3.2. Модификации метода Ньютона	13
3.3. Смешанный метод Ньютона	14
3.4. Метод LS	16
4. Методы адаптации моделей нелинейных искажений не требующие явного вычисления матрицы Гессе	18
4.1. Метод сопряженных градиентов	18
4.2. Метод DCD	18
4.3. Квазиньютоновские методы	21
4.4. Метод Barzilai-Borwein	23
4.5. Метод DFP	24
4.6. Метод BFGS	25

5.	Математические модели линейных и нелинейных искажений в системах связи	25
5.1.	Линейная модель искажений сигнала в системах связи	26
5.2.	Полиномиальная нелинейных искажений в системах связи	30
5.3.	Модель Вольтерра нелинейных искажений в системах связи	33
5.4.	Модель Гаммерштейна нелинейных искажений в системах связи	33
5.5.	Модель Винера нелинейных искажений в системах связи	35
5.6.	Модель Винера-Гаммерштейна нелинейных искажений в системах связи	35
5.7.	Нейросетевые структуры аппроксимации нелинейных искажений приемо-передающего тракта	35

II..Исследование эффективности методов адаптивной компенсации нелинейных помех в приёмном тракте мобильного терминала 36

1.	Структура тестовой платформы	36
2.	Формирование тестового набора данных	38
3.	Исследование алгоритмов компенсации стационарной паразитной помехи на приёмнике	40
3.1.	Аппроксимация стационарных нелинейных искажений на приёмнике устройства связи на основе классических и нейросетевых структур	40
3.2.	Исследование опорных уровней компенсации стационарной помехи методами второго порядка	40
4.	Исследование моделей цифрового предискажения сигнала передающего устройства в условиях динамического изменения выходной мощности нелинейного усилителя	45
4.1.	Аппроксимация динамической нелинейности на основе классических и нейросетевых структур	45
4.2.	Исследование опорных уровней компенсации динамических нелинейных искажений	45

III.Практическая реализация нелинейных корректоров в системах связи 46

1.	Исследование компенсации нелинейных искажений методами не требующими явного вычисления матрицы Гессе	46
2.	Сокращение ресурсов при реализации моделей нелинейных искажений за счет канонического тензорного разложения	55
3.	Сокращение ресурсов при реализации моделей нелинейных искажений за счет канонического тензорного разложения	55
4.	Сокращение ресурсов при реализации моделей нелинейных искажений путем оптимизации гиперпараметров модели	55

4.1.	Метод Orthogonal Least Squares	55
4.2.	Метод Proximal Policy Optimization	55
5.	Методы экономии ресурсов при реализации адаптивных фильтров . . .	55
Заключение		61

Введение

По мере совершенствования телекоммуникационных технологий возникают задачи повышения скорости передачи информации. Так например разработка систем стандарта 5G предполагает решение задач связанных с увеличением скорости передачи информации по сравнению с системами стандарта 4G. Увеличение скорости передачи требует повышения порядков цифровой модуляции. А это в свою очередь требует улучшения характеристик чувствительности приёмника.

Кроме того, важнейшим направлением развития телекоммуникационных технологий является унификация производства аппаратуры. Так основная тенденция это упрощение аналоговых трактов аппаратуры с компенсацией возникающих искажений адаптивными цифровыми корректорами [dig_front_end].

Современным направлением решения перечисленных задач является использование адаптивных систем нелинейной коррекции, которые позволяют существенно повысить коэффициент полезного действия передающих трактов системы при сохранении характеристик линейности системы в соответствии с принятыми международными стандартами связи [3gpp_36_804]. С точки зрения пользователя мобильного терминала это означает увеличение времени автономной работы телефона.

В данной работе рассматриваются алгоритмы подавления помех в приёмном тракте мобильного устройства, возникающих в передающем тракте путём прохождения через нелинейные элементы, такие как дуплексеры и усилители мощности [dpd_thesis]. Ввиду интегрального исполнения аналоговых компонент, их характеристики имеют существенно нелинейный вид. Кроме того, ввиду компактного расположения компонент в RF-чипсете мобильного терминала, нелинейные компоненты, возникающие в передающем тракте, попадают в приёмный тракт мобильного телефона, проходя по различным путям распространения, уменьшая чувствительность приёмника, а соответственно и пропускную способность нисходящего канала связи.

Адаптивная компенсация паразитных нелинейных помех позволяет улучшить пропускную способность нисходящего канала связи, повысить КПД усилителя передатчика, а также ослабить требования, предъявляемые к аналоговым дуплексерам.

В работе приводятся результаты симуляции работы модуля адаптивной коррекции для сигналов помехи различной ширины полосы, а также для различных каналов распространения помехи.

Глава I

Методы адаптивной компенсации нелинейных искажений сигнала в приеме-передающих устройствах систем связи

1. Постановка задачи адаптивной компенсации нелинейных искажений в устройствах связи

1.1 Проблема возникновения паразитных помех в приёмном тракте приема-передающего устройства

В данном разделе формулируется задача адаптивной компенсации нелинейных паразитных помех, возникающих на приёмнике приема-передающего устройства. Приводится схема адаптивной компенсации данного рода помех. А также описывается модель формирования паразитных помех на основе качественного представления физических процессов.

В качестве примера возникновения паразитных помех в приёмном тракте рассмотрим приёмно-передающее устройство, состоящее из двух каналов передатчика и приёмника (2T2R), как это показано на рис. 1. При этом передатчик и приёмник разнесены по частоте.

В процессе прохождения сигнала на несущей частоте f_0 через нелинейные цепи передатчика, генерируются компоненты на кратных частотах f_0 , $2f_0$, $3f_0$ и выше. Гармоники порядка выше 3-его, как правило, имеют существенно меньшую мощность по сравнению с гармониками 1-го, 2-го и 3-его порядка.

Среди элементов передающего тракта, представленных на рис. 1, наибольший вклад в формирование нелинейных компонент спектра вносит аналоговый усилитель мощности. В связи с этим будем далее считать, что именно усилитель мощности является источником нелинейности. Согласно стандарту LTE, используемому в системах связи четвертого поколения 4G, пара передатчик-приёмник может работать на

таких частотах, при которых первая, вторая или третья гармоника сигнала передатчика попадает в полосу приёмного тракта того же устройства [3gpp_36_104]:

- LTE Band 2 (UL 1920–1980 MHz) – LTE Band 2 (DL 1930–1990 MHz)
- LTE Band 8 (UL 880-915 MHz) - LTE Band 3 (DL 1805-1880 MHz)
- LTE Band 17 (UL 704-716 MHz) - LTE Band 4 (DL 2110-2155 MHz)

Кроме того, за счёт малых размеров RF-чипсетов, компоненты внутри чипсета расположены компактно. Вследствие технологических ограничений производства существует сложность обеспечения изоляции отдельных компонентов RF-чипсета. В связи с этим появляются различные пути распространения помехи в приёмнике, как это показано на рис. 1.

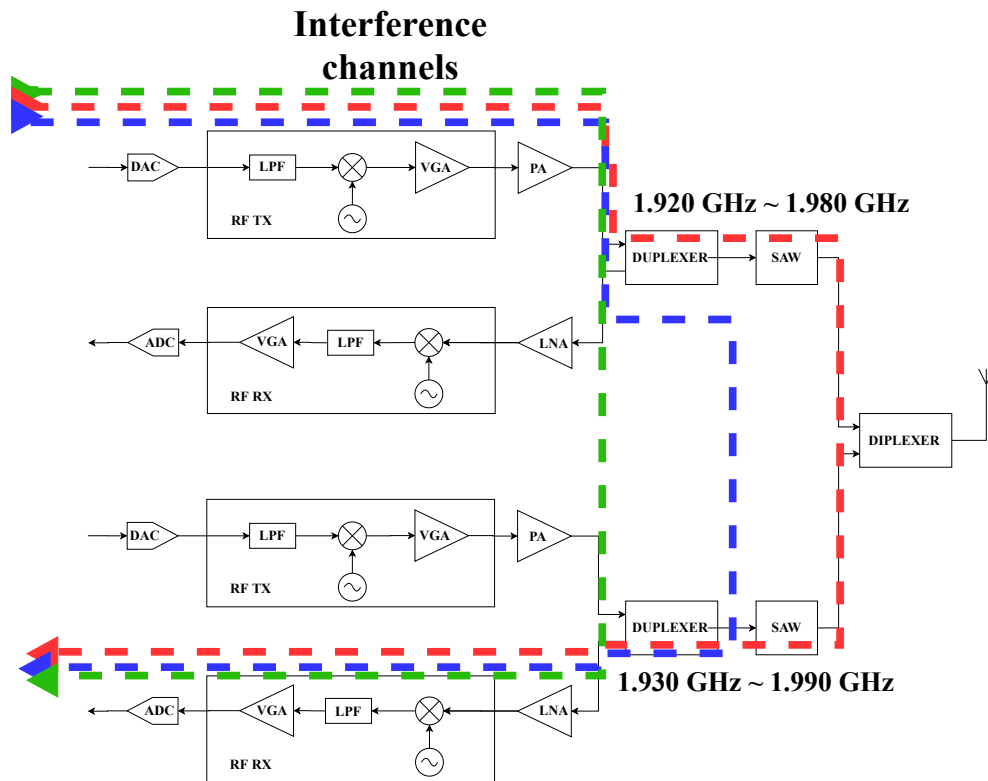


Рис. 1. Каналы распространения 1-ой гармоники нелинейных помех в приёмно-передающем устройстве 2T2R

Пример спектра сигнала передатчика на несущей частоте $f_0 = 1.93$ ГГц и помехи, возникающей в приёмнике на несущей частоте передатчика $f_1 = 1.96$ ГГц изображен на рис. 2. Задача модуля компенсации заключается в том, чтобы понизить уровень этой помехи для обеспечения требуемого уровня чувствительности приёмника.

Таким образом, одной из задач данной работы является разработка и исследование модуля адаптивной нелинейной коррекции, который позволит компенсировать

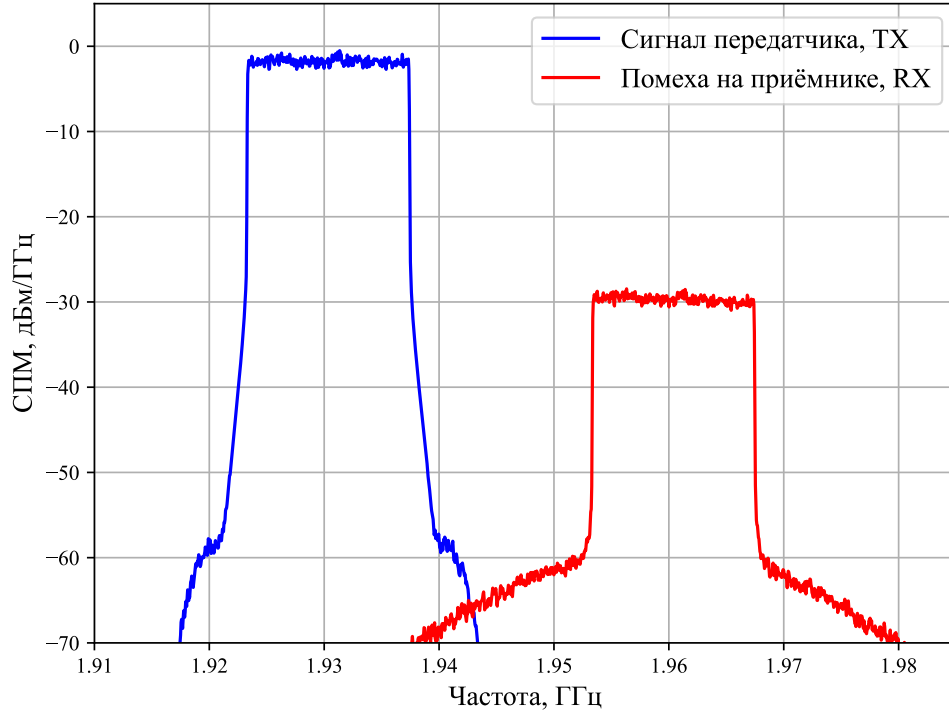


Рис. 2. Спектральная плотность мощности сигнала передатчика на частоте $f_0 = 1.93$ ГГц и паразитной помехи на частоте $f_1 = 1.96$ ГГц

паразитные помехи в приёмнике на частотах, кратных несущей частоте передатчика. Кроме того, такой подход позволит снизить требования к блокам фильтрации аналогового сигнала: дуплексерам и полосовым фильтрам.

1.2 Схема компенсации паразитных нелинейных помех в приёмном тракте мобильного терминала

Как отмечалось ранее, будем считать, что основным источником нелинейных помех в приёмном тракте является аналоговый усилитель мощности в передающем тракте мобильного терминала [dig_front_end].

Предлагается идентифицировать помехи, попадающих в приёмник путем адаптации нелинейной модели по критерию минимума среднего квадрата ошибки [adapt_filt]. Такая схема компенсации нелинейных помех изображена на рис. 3.

Исходный цифровой сигнал x на несущей частоте f_0 проходит через блок нелинейной модели $f(\cdot)$, в результате чего на её выходе формируется цифровой сигнал $y = f(x)$. Сигнал проходит по передающему тракту: через ЦАП, RF-модуль, аналоговый усилитель мощности, затем вследствие наличие различных путей распространения, паразитные гармоники исходного сигнала на частотах $f_0, 2f_0, 3f_0$ попадают на приёмную часть.

Проходя по приёмному тракту помеха приобретает вид цифрового сигнала, обозначенного как d . Помимо помехи в приёмную часть терминала поступает полезный сигнал r . Таким образом, на выходе АЦП – сумма $r + d$.

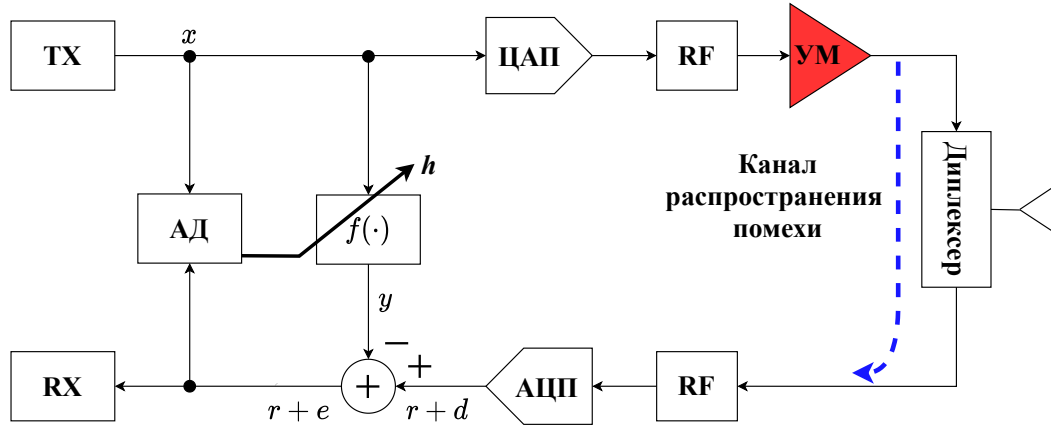


Рис. 3. Схема компенсации нелинейных помех в приёмнике приёмно-передающего устройства

Блок адаптации АД производит идентификацию, подстраивая коэффициенты z нелинейной модели $f(\cdot)$ посредством измерения отклонения $e = d - y$ принятой приёмным трактом помехи от выхода нелинейной модели. В общем случае можно считать, что коэффициент корреляции отправляемого на передатчике сигнала x и полезного принятого сигнала r равен нулю. Поэтому в результате работы модуля компенсации результирующий сигнал на приёмнике будет стремиться к полезному принятому сигналу $r + e \rightarrow r$. Задачу идентификации паразитной помехи путём адаптации нелинейной модели по критерию минимума среднего квадрата сформулируем следующим образом:

$$\begin{cases} J = e^* e \rightarrow \min_z, \\ e = d - f(x, z) \end{cases} \quad (1.2.1)$$

1.3 Проблема возникновения нелинейных искажений сигнала передающего тракта

Ввиду нелинейности характеристики передающего тракта формируются нелинейные искажения, которые значительно влияют на качество радиочастотных сигналов, создавая помехи в канале связи. Такие нелинейные искажения приводят к увеличению битовой ошибки на приемнике, генерируют внутриполосные и внеполосные помеховые сигналы и ухудшают качество передачи сигналов соседних полос.

Как было отмечено ранее, наибольший вклад в искажения сигнала в передающем тракте вносят нелинейные усилители мощности. В результате прохождения через УМ формируются компоненты спектра внутри полосы передатчика, а также внеполосные искажения. На рис. 4 изображена спектральная плотность мощности сигнала передатчика до и после прохождения через УМ.

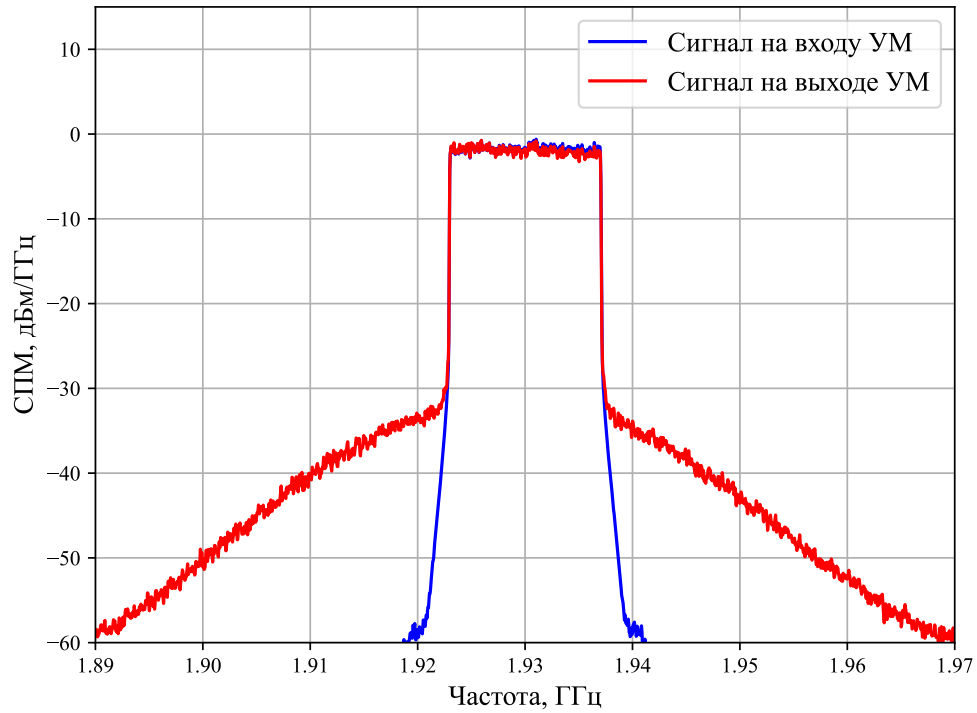


Рис. 4. Спектральная плотность мощности сигнала передатчика на входе и выходе усилителя мощности на частоте $f_0 = 1.93$ ГГц

1.4 Схема компенсации нелинейных искажений в передатчике приемо-передающего устройства

Для предотвращения искажений в современных базовых станциях и сотовых устройствах широко используются методы цифровой предискажения DPD [1]-[3].

Устройство DPD представлено блоком с обратной нелинейной характеристикой, изменяющим входной сигнал УМ так, чтобы минимизировать нелинейных искажений на выходе УМ (рис. 5).

Таким образом, DPD предискажает входной сигнал УМ таким образом, чтобы

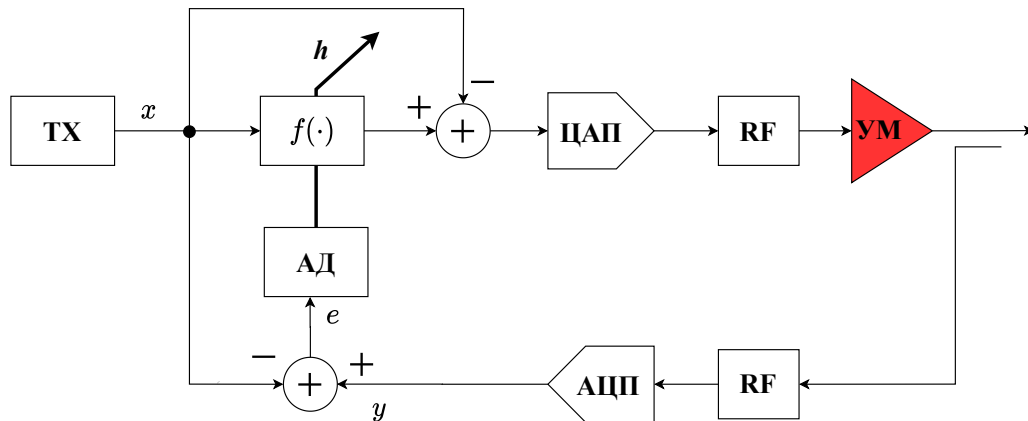


Рис. 5. Структура системы цифрового предискажения

на выходе сформировался линейно искаженный сигнал. Такой подход позволяет работать в режимах высокой нелинейности УМ и, в результате, повысить КПД УМ. Задача оптимизации цифрового предискажения может быть представлена математическим выражением [4], описывающим минимизацию отклонения входа DPD от выхода УМ:

$$\|g(x - f(x, z)) - x\|_2^2 \rightarrow \min_z, \quad (1.4.1)$$

где x – входной сигнала УМ, $g(\cdot)$ – нелинейная характеристика УМ, $f(\cdot)$ – модель цифрового предискажения с адаптивными параметрами z . Предположим, что нелинейную характеристику УМ $g(x)$ можно аппроксимировать линейной функцией в окрестности рабочей точки:

$$g(x - f(x, z)) \approx g(x) - g'_x(x)f(x, z), \quad (1.4.2)$$

где $g'_x(x)$ – производная нелинейной функции УМ по входу. Поскольку DPD должен работать в таком режиме УМ, чтобы уровень нелинейных искажений оставался значительно ниже уровня передаваемого сигнала, то $g'_x(x) \approx I$ – будет представлять собой единичную матрицу. Подставляя (1.4.2) в (1.4.1), получаем выражение:

$$\|f(x, z) - e\|_2^2 \rightarrow \min_z, \quad (1.4.3)$$

где $e = g(x) - x$ – вектор ошибки.

1.5 Критерий компенсации нелинейных искажений в устройствах связи

Существует несколько способов обработки сигнала [adapt_filt_haykin]. Симуляция алгоритмов как правило проводится на сохраненных блоках данных. В этом случае имеется мощный вычислитель (компьютер) и возможность вести математическую обработку матриц и других объектов алгоритмов. Это блочный режим работы.

При аппаратной реализации блочный режим стараются заменить стохастическим в реальном времени [adapt_filt_haykin], чтобы минимизировать задержки и сократить ресурсы. В этом случае векторы состояния, корреляционные матрицы, шаг алгоритма, а также другие объекты, обновляются каждый отсчет.

При рассмотрении методов компенсации паразитных помех будем считать, что на вход блока адаптации поступает вектор комплексных отсчетов передатчика $\mathbf{x} \in \mathbb{C}^{N \times 1}$ длины N , вектор отсчетов с приёмника $\mathbf{d} \in \mathbb{C}^{N \times 1}$ также длины N .

Задача блока адаптации заключается в том, чтобы так преобразовать исходный вектор \mathbf{x} , чтобы как можно лучше приблизить к вектору отсчетов на приёмнике \mathbf{d} путём минимизации нормы вектора ошибки $\mathbf{e} = \mathbf{d} - f(\mathbf{x}, \mathbf{z}) \in \mathbb{C}^{N \times 1}$, где $f(\cdot)$ – оператор преобразования исходного вектора в вектор отсчетов на выходе модели помехи. Оператор $f(\cdot)$ определяется коэффициентами $\mathbf{z} \in \mathbb{C}^{K \times 1}$, K – число адаптивных

параметров модели.

Рассмотрим требования к целевой функции (метрике) алгоритмов компенсации. Метрика величины отклонения выхода модели помехи от вектора отсчетов самой помехи должна быть вещественнозначной и неотрицательной для реализации процедуры поиска минимума данной метрики. Кроме того, желательно задать целевую функцию квадратичной и выпуклой для применения эффективных методов адаптации [**convex_opt**].

Такой вещественной, неотрицательной и квадратичной метрикой является средний квадрат ошибки (MSE - Mean Square Error). Физический смысл среднего квадрата ошибки – энергия отклонения выхода нелинейной модели от помехи, измеренной на приёмнике. При блочной обработке сигнала MSE имеет вид:

$$J = \|e\|_2^2 = e^H e. \quad (1.5.1)$$

Помимо MSE на практике применяется метрика - нормированный средний квадрат ошибки (NMSE - Normalized Mean Square Error):

$$J = \frac{\|e\|_2^2}{\|d\|_2^2} = \frac{e^H e}{d^H d}. \quad (1.5.2)$$

В данном случае энергия ошибки нормируется к энергии сигнала приёмника для того чтобы оценить уровень ошибки независимо от динамического диапазона сигнала на входе приемника.

Реальные сигналы, используемые в системах связи [**sklyar_dig_telecom**] являются мощностными, то есть обладают бесконечной энергией на бесконечном промежутке времени, однако в уравнениях (1.5.1), (1.5.2) фигурирует энергия ошибки и сигнала помехи, рассчитанная на конечной длине блока.

2. Методы первого порядка для адаптации моделей нелинейных искажений

Методы первого порядка [**polyak_optimiz**] занимают центральное место в задачах адаптации нелинейных моделей, применяемых для компенсации искажений в трактах телекоммуникационных устройств. Их ключевое преимущество заключается в низкой вычислительной сложности и хорошей масштабируемости, что делает их особенно привлекательными для аппаратной реализации в системах с ограниченными ресурсами.

В отличие от более сложных второпорядковых алгоритмов, методы первого порядка — такие как стохастический градиентный спуск и его модификации — опираются лишь на локальную информацию о градиенте функции ошибки, обеспечивая устойчивую и быструю адаптацию даже при неполном знании статистики сигнала.

ла. Благодаря этому они хорошо подходят для работы в реальном времени и могут эффективно функционировать при изменяющихся условиях канала или параметров нелинейности.

Кроме того, градиентные методы допускают реализацию в фиксированной точке, что позволяет использовать простые арифметические блоки и снижает энергопотребление. Такая особенность делает их оптимальным выбором для встроенных и энергоэффективных систем, где важно обеспечить баланс между скоростью сходимости и аппаратной сложностью. В контексте нелинейной компенсации это обеспечивает возможность динамического обновления коэффициентов модели с минимальными затратами ресурсов при сохранении высокой точности восстановления сигнала.

2.1 Метод градиентного спуска

Градиентный спуск для случая нелинейной модели с вещественными параметрами представляет собой следующий итеративный алгоритм:

$$\mathbf{z}_{k+1} = \mathbf{z}_k - \mu (D_{\mathbf{z}_k} J(\mathbf{z}_k))^T, \mathbf{z}_k \in \mathbb{R}^{K \times 1}, \quad (2.1.1)$$

$$\left\| D_{\mathbf{z}^*} J(\mathbf{z}_k) - D_{\mathbf{z}^*} J(\mathbf{z}_{k+1}) \right\|_2 < L \left\| D_{\mathbf{z}^*} J(\mathbf{z}_k) - D_{\mathbf{z}^*} J(\mathbf{z}_{k+1}) \right\|_2, \quad \mu \leq L, \quad (2.1.2)$$

где $D_{\mathbf{z}} J \in \mathbb{R}^{1 \times K}$ – оператор вычисления производной функции J по адаптивным параметрам \mathbf{z} . Условие Липшица `[polyak_optimiz]` задаёт ограничение на скорость поиска минимума целевой функции. Для ускорения алгоритма достаточно увеличить шаг μ , однако при слишком больших μ – алгоритм расходится. Критическое значение, при котором наступает стагнация алгоритма – $\mu_{\text{кр}} = \frac{L}{2}$.

Ввиду того, что в устройствах связи работают с комплекснозначными сигналами, параметры моделей нелинейной обработки сигналов также выбирают комплекснозначными. Направление возрастания вещественной функции комплексного переменного определяется производной вещественной функции по комплексно-сопряженным параметрам `[wirt_deriv_book]`. В связи с этим, метод градиентного спуска для моделей с комплексными параметрами строится следующим образом:

$$\mathbf{z}_{k+1} = \mathbf{z}_k - \mu (D_{\mathbf{z}_k^*} J(\mathbf{z}_k))^T, \mathbf{z}_k \in \mathbb{C}^{K \times 1}, \quad (2.1.3)$$

где $D_{\mathbf{z}_k^*} J \in \mathbb{C}^{1 \times K}$ – оператор вычисления производной вещественной функции J по адаптивным комплексно-сопряженным параметрам \mathbf{z}^* .

Получим разностное уравнение метода градиентного спуска оптимизации вещественной скалярной функции потерь, зависящей от комплексных векторов параметров. Для этого разложим функцию потерь J в окрестности нуля в ряд Тейлора до членов первого порядка малости:

$$dJ = D_{\mathbf{z}} J d\mathbf{z} + D_{\mathbf{z}^*} J d\mathbf{z}^* \in \mathbb{R}, \quad (2.1.4)$$

где $D_z J \in \mathbb{C}^{1 \times K}$, $D_{z^*} J \in \mathbb{C}^{1 \times K}$, $dz \in \mathbb{C}^{K \times 1}$, $dz^* \in \mathbb{C}^{K \times 1}$. Разложим выход нелинейной модели в Ряд Тейлора в окрестности нуля до членов первого порядка малости [wirt_deriv_book]:

$$d\mathbf{y} = D_z \mathbf{y} dz + D_{z^*} \mathbf{y} dz^* \in \mathbb{C}^{N \times 1}, \quad (2.1.5)$$

где $D_z \mathbf{y}$, $D_{z^*} \mathbf{y} \in \mathbb{C}^{N \times K}$ – матрицы Якоби, производные выхода модели по прямым и комплексно-сопряженным параметрам модели.

Распишем дифференциал среднего квадрата ошибки, введенного в (1.5.1) и подставим разложение выхода нелинейной модели \mathbf{y} до членов первого порядка малости (2.1.5):

$$\begin{aligned} dJ &= d(e^H e) = de^H e + e^H de = -e^T d\mathbf{y}^* - e^H d\mathbf{y} = \\ &= -e^T D_z \mathbf{y}^* dz - e^T D_{z^*} \mathbf{y}^* dz^* - e^H D_z \mathbf{y} dz - e^H D_{z^*} \mathbf{y} dz^* = \\ &= (-e^T D_z \mathbf{y}^* - e^H D_z \mathbf{y}) dz + (-e^T D_{z^*} \mathbf{y}^* - e^H D_{z^*} \mathbf{y}) dz^*. \end{aligned} \quad (2.1.6)$$

Сравнивая выражения перед дифференциалом \mathbf{z}^* в (2.1.4) и (2.1.6), получаем выражение производной функции потерь по комплексно-сопряженным параметрам \mathbf{z}^* :

$$(D_{z^*} J)^T = (-e^T D_{z^*} \mathbf{y}^* - e^H D_{z^*} \mathbf{y})^T = -(D_z \mathbf{y})^H e - (D_{z^*} \mathbf{y})^T e^* \in \mathbb{C}^{K \times 1}. \quad (2.1.7)$$

Таким образом, подставляя (2.1.7) в (2.1.3) получаем разностное уравнение градиентного спуска для функции потерь MSE с вектором ошибки общего вида, то есть зависящим от прямых и сопряженных параметров $\mathbf{e} = \mathbf{e}(\mathbf{x}, \mathbf{z}, \mathbf{z}^*)$:

$$\mathbf{z}_{k+1} = \mathbf{z}_k + \mu(D_z \mathbf{y})^H \mathbf{e} + \mu(D_{z^*} \mathbf{y})^T \mathbf{e}^* \in \mathbb{C}^{K \times 1} \quad (2.1.8)$$

Пусть вектор отклонения выхода модели от значений помехи $\mathbf{e}(\mathbf{x}, \mathbf{z}) \in \mathbb{C}^{N \times 1}$ — голоморфная функция относительно \mathbf{z} , то есть не зависит от вектора комплексно-сопряженных параметров \mathbf{z}^* . В этом случае Якобиан выхода модели от сопряженных параметров равен нулю:

$$D_{z^*} \mathbf{y} = \mathbf{0}. \quad (2.1.9)$$

Тогда из (2.1.8) разностное уравнение метода градиентного спуска может быть представлено как:

$$\mathbf{z}_{k+1} = \mathbf{z}_k + \mu(D_z \mathbf{y})^H \mathbf{e}. \quad (2.1.10)$$

В нелинейной адаптивной обработке сигналов условие (2.1.9) используется при построении нелинейных моделей, поскольку в этом случае снижается вычислительная сложность пересчета параметров адаптивных моделей, снижается требования по памяти, энергопотребление, а также занимаемая площадь на кристалле.

2.2 Модификации метода градиентного спуска

Классический метод градиентного спуска (2.1.3) является эффективным в случае с гладкими выпуклыми функциями потерь, однако в случае сильно искривлённого ландшафта или шумных градиентов он может сходиться медленно. Для ускорения сходимости и сглаживания колебаний используется метод тяжёлого шара (momentum), предложенный Поляком [polyak1964]. Основная идея заключается во введении вектора скорости, аккумулирующего информацию о предыдущих шагах:

$$\begin{cases} \mathbf{m}_{k+1} = \beta \mathbf{m}_k + \mu (D_{\mathbf{z}^*} J(\mathbf{z}_k))^T, \\ \mathbf{z}_{k+1} = \mathbf{z}_k - \mathbf{m}_{k+1}, \end{cases} \quad (2.2.1)$$

где $0 \leq \beta < 1$ — коэффициент инерции. Такой подход позволяет снизить осцилляции и ускорить движение вдоль пологих направлений, однако градиент вычисляется в текущей точке, что может приводить к избыточным колебаниям.

Для устранения этого недостатка была предложена модификация Нестерова, известная как *ускоренный градиент Нестерова* [nesterov1983method] (Nesterov accelerated gradient, NAG). В этом подходе градиент вычисляется в «прогнозируемом» положении, что стабилизирует алгоритм адаптации:

$$\begin{cases} \mathbf{m}_{k+1} = \beta \mathbf{m}_k + \mu (D_{\mathbf{z}^*} J(\mathbf{z}_k - \beta \mathbf{v}_k))^T, \\ \mathbf{z}_{k+1} = \mathbf{z}_k - \mathbf{m}_{k+1}. \end{cases} \quad (2.2.2)$$

Таким образом, данный метод корректирует направление движения до обновления параметров, что обеспечивает более устойчивую и быструю сходимость для выпуклых задач.

Дальнейшее развитие концепции инерционного градиентного спуска привело к созданию алгоритма *Adam* (Adaptive Moment Estimation) [kingma2014adam], который сочетает механизмы накопления импульса градиента и адаптивной нормализации шага по каждой координате параметров. В отличие от классических схем с фиксированной скоростью обучения, Adam автоматически масштабирует величину шага в зависимости от первой и второй статистических моментов градиента, что обеспечивает баланс между скоростью сходимости и устойчивостью обновлений. Такая адаптивность позволяет алгоритму поддерживать высокую скорость обучения на ранних этапах и эффективно выходить из окрестностей локальных минимумов, что делает его одним из наиболее универсальных и применимых методов стохастической оптимизации:

$$\begin{cases} \mathbf{m}_{k+1} = \beta_1 \mathbf{m}_k + (1 - \beta_1) (D_{\mathbf{z}^*} J(\mathbf{z}_k))^T, \\ \mathbf{v}_{k+1} = \beta_2 \mathbf{v}_k + (1 - \beta_2) |(D_{\mathbf{z}^*} J(\mathbf{z}_k))^T|^2, \\ \hat{\mathbf{m}}_{k+1} = \frac{\mathbf{m}_{k+1}}{1 - \beta_1^{k+1}}, \quad \hat{\mathbf{v}}_{k+1} = \frac{\mathbf{v}_{k+1}}{1 - \beta_2^{k+1}}, \\ \mathbf{z}_{k+1} = \mathbf{z}_k - \mu \frac{\hat{\mathbf{m}}_{k+1}}{\sqrt{\hat{\mathbf{v}}_{k+1}} + \varepsilon}, \end{cases} \quad (2.2.3)$$

где $\beta_1 \approx 0.9$, $\beta_2 \approx 0.999$, $\varepsilon \approx 10^{-8}$. Adam объединяет преимущества момента (инерция) и адаптивных методов (нормализация шага по координатам), что обеспечивает стабильную сходимость даже при шумных или нестационарных градиентах.

Эволюция этих подходов отражает постепенное улучшение базового метода: от простого градиентного шага (2.1.3) к инерционному ускорению (2.2.1), затем к прогнозирующему ускорению Нестерова (2.2.2), и, наконец, к адаптивно-инерционной схеме (2.2.3).

3. Методы второго порядка для адаптации моделей нелинейных искажений

3.1 Метод Ньютона

Несмотря на широкое распространение и простоту реализации методов первого порядка, их эффективность существенно снижается в задачах с жёсткими (плохо обусловленными) функциями, где направление антиградиента может существенно отличаться от направления наискорейшего убывания. Для ускорения сходимости и более точного учёта локальной геометрии целевой функции применяются методы второго порядка, использующие информацию не только о градиенте, но и о кривизне поверхности уровня.

Классическим представителем данного класса является *метод Ньютона*, основанный на аппроксимации функции её локальным квадратичным разложением в окрестности текущей точки. Основная идея заключается в том, что, если функция $J(\mathbf{z}, \mathbf{z}^*)$ близка к квадратичной, то минимум аппроксимации можно найти аналитически, что обеспечивает сверхлинейную сходимость. В отличие от методов первого порядка, метод Ньютона использует матрицу Гессе (матрицу вторых производных) для корректировки направления поиска, что позволяет автоматически масштабировать шаги в зависимости от локальной кривизны поверхности.

Таким образом, метод Ньютона интерпретируется как переход от *направлений убывания* к *направлениям оптимального спуска*, обеспечивающим более точное приближение минимума.

Для формального описания метода Ньютона введем следующие обозначения. Производная второго порядка скалярной вещественной функции J по вектору па-

раметров описывается следующим образом:

$$D_z(D_z J)^T = H_{z,z} J \in \mathbb{C}^{K \times K} - \text{матрица Гессе.} \quad (3.1.1)$$

Для удобства записи алгоритма метода Ньютона введем вектор удвоенной длины, включающий вектор прямых и сопряженных параметров:

$$\tilde{z}^T = \begin{pmatrix} z & z^* \end{pmatrix}^T \quad (3.1.2)$$

Используя введенные выше обозначения, дифференциал второго порядка функции J может быть описан следующим образом `[complex_deriv]`:

$$d^2 J = \begin{pmatrix} dz^T & d(z^*)^T \end{pmatrix} \begin{pmatrix} H_{z,z} J & H_{z^*,z} J \\ H_{z,z^*} J & H_{z^*,z^*} J \end{pmatrix} \begin{pmatrix} dz \\ dz^* \end{pmatrix} = \tilde{z}^T H_{\tilde{z},\tilde{z}} J \tilde{z} \quad (3.1.3)$$

Таким образом, разностное уравнение метода Ньютона может быть представлен следующим образом:

$$\tilde{z}_{k+1} = \tilde{z}_k - \mu (H_{\tilde{z},\tilde{z}} J)^{-1} (D_{\tilde{z}^*} J)^T \quad (3.1.4)$$

Получим разностное уравнение метода Ньютона для функции потерь MSE (1.5.1). Градиент функции потерь по вектору параметров удвоенной длины может быть получен из выражения (2.1.7):

$$(D_{\tilde{z}^*} J)^T = \begin{pmatrix} D_z J & D_{z^*} J \end{pmatrix}^T = \begin{pmatrix} (D_{z^*} J)^* & D_{z^*} J \end{pmatrix}^T, \quad (3.1.5)$$

где использовано свойство $(D_{z^*} J)^* = D_z J^* = D_z J$ ввиду вещественности функции потерь $J \in \mathbb{R}$.

Заметим также, что ввиду определения матрицы Гессе (3.1.1), достаточно найти две из четырех составляющих матрицы (3.1.3):

$$H_{z,z^*} J = D_{z^*} (D_z J)^T = (D_z (D_{z^*} J)^T)^* = (H_{z^*,z} J)^*, \quad (3.1.6)$$

$$H_{z,z} J = D_z (D_z J)^T = (D_{z^*} (D_{z^*} J)^T)^* = (H_{z^*,z^*} J)^* \quad (3.1.7)$$

Найдем теперь матрицы Гессе для MSE из выражения (2.1.7):

$$d(D_{z^*} J)^T = -d \left[(D_z \mathbf{y})^H \mathbf{e} \right] - d \left[(D_{z^*} \mathbf{y})^T \mathbf{e}^* \right]. \quad (3.1.8)$$

Найдем первое слагаемое выражения (3.1.8):

$$\begin{aligned} d \left[(D_z \mathbf{y})^H \mathbf{e} \right] &= d \left[\mathbf{I}_K (D_z \mathbf{y})^H \mathbf{e} \right] = d \left[(\mathbf{e}^T \otimes \mathbf{I}_K) \text{vec}(D_z \mathbf{y})^H \right] = \\ &= (-d \mathbf{y}^T \otimes \mathbf{I}_K) \text{vec}(D_z \mathbf{y})^H + (\mathbf{e}^T \otimes \mathbf{I}_K) d \text{vec}(D_z \mathbf{y})^H = \\ &= -(D_z \mathbf{y})^H d \mathbf{y} + (\mathbf{e}^T \otimes \mathbf{I}_K) d \text{vec}(D_z \mathbf{y})^H, \end{aligned} \quad (3.1.9)$$

где $\mathbf{I}_K \in \mathbb{R}^{K \times K}$ – единичная матрица, \otimes – произведение Кронекера, $\text{vec}(\cdot)$ – оператор векторизации матрицы, вытягивает матрицу в единый столбец. Отметим, что в выкладках (3.1.9) использовано свойство $\text{vec}(ABC) = (C^T \otimes A)\text{vec}(B)$ [**wirt_deriv_book**].

В выражении (3.1.9) разложим вектор в ряд Тейлора до слагаемых первого порядка малости:

$$d\text{vec}(D_z \mathbf{y})^H = D_z \text{vec}(D_z \mathbf{y})^H dz + D_{z^*} \text{vec}(D_{z^*} \mathbf{y})^H dz^*. \quad (3.1.10)$$

Подставим (2.1.5), (3.1.10) в (3.1.9):

$$\begin{aligned} d \left[(D_z \mathbf{y})^H \mathbf{e} \right] &= -(D_z \mathbf{y})^H (D_z \mathbf{y}) dz - (D_z \mathbf{y})^H (D_{z^*} \mathbf{y}) dz^* + \\ &+ (\mathbf{e}^T \otimes \mathbf{I}_K) D_z \text{vec}(D_z \mathbf{y})^H dz + (\mathbf{e}^T \otimes \mathbf{I}_K) D_{z^*} \text{vec}(D_z \mathbf{y})^H dz^*. \end{aligned} \quad (3.1.11)$$

По аналогии с (3.1.11) получаем выражение для второго слагаемого дифференциала градиента функции потерь MSE:

$$\begin{aligned} d \left[(D_{z^*} \mathbf{y})^T \mathbf{e}^* \right] &= -(D_{z^*} \mathbf{y})^T (D_{z^*} \mathbf{y}^*) dz^* - (D_{z^*} \mathbf{y})^T (D_z \mathbf{y}^*) dz + \\ &+ (\mathbf{e}^H \otimes \mathbf{I}_K) D_z \text{vec}(D_{z^*} \mathbf{y})^T dz + (\mathbf{e}^H \otimes \mathbf{I}_K) D_{z^*} \text{vec}(D_{z^*} \mathbf{y})^T dz^*. \end{aligned} \quad (3.1.12)$$

Подставим выражения (3.1.11), (3.1.12) в (3.1.8) и приравняем выражения при дифференциалах dz , dz^* с выражениями при дифференциалах в (2.1.4), получим искомые Гесссианы:

$$\begin{aligned} H_{z^*,z} J &= D_z (D_{z^*} J)^T = (D_z \mathbf{y})^H (D_z \mathbf{y}) - (\mathbf{e}^T \otimes \mathbf{I}_K) D_z \text{vec}(D_z \mathbf{y})^H + \\ &+ (D_{z^*} \mathbf{y})^T (D_z \mathbf{y}^*) - (\mathbf{e}^H \otimes \mathbf{I}_K) D_z \text{vec}(D_{z^*} \mathbf{y})^T, \end{aligned} \quad (3.1.13)$$

$$\begin{aligned} H_{z^*,z^*} J &= D_{z^*} (D_{z^*} J)^T = (D_z \mathbf{y})^H (D_{z^*} \mathbf{y}) - (\mathbf{e}^T \otimes \mathbf{I}_K) D_{z^*} \text{vec}(D_z \mathbf{y})^H + \\ &+ (D_{z^*} \mathbf{y})^T (D_{z^*} \mathbf{y}^*) - (\mathbf{e}^H \otimes \mathbf{I}_K) D_{z^*} \text{vec}(D_{z^*} \mathbf{y})^T. \end{aligned} \quad (3.1.14)$$

Таким образом, выражения (3.1.13), (3.1.14), (3.1.6), (3.1.7) определяют полную матрицу вторых производных, а выражения (3.1.5), (2.1.7) полный градиент функции потерь MSE в разностном уравнении (3.1.4).

3.2 Модификации метода Ньютона

Как отмечалось ранее, метод Ньютона сходится квадратично вблизи точки оптимума. При старте алгоритма с удалённой от оптимума начальной точки итеративная процедура может расходиться [**polyak_optimiz**]. Для обеспечения глобальной сходимости широко применяются модификации метода Ньютона.

Алгоритм Левенберга-Марквардта [**polyak_optimiz**] является одной из модификаций метода Ньютона. Метод заключается в том, чтобы внести добавку к гесссиану

(3.1.4) для улучшения численной устойчивости алгоритма при обращении матрицы:

$$\mathbf{z}_{k+1} = \mathbf{z}_k - (\mathbf{H}_J(\mathbf{z}_k) + \lambda_k \mathbf{I}_K)^{-1} D_{\mathbf{z}} J(\mathbf{z}_k), \quad (3.2.1)$$

где λ_k – действительное положительное число. Добавка $\lambda_k \mathbf{I}_K$ – регуляризация, которая изменяет направление шага алгоритма в сторону антиградиента тем больше, чем больше норма $\|\lambda_k \mathbf{I}_K\|$:

$$\lim_{\lambda_k \rightarrow \infty} (\mathbf{H}_J(\mathbf{z}_k) + \lambda_k \mathbf{I}_K)^{-1} = \mu \mathbf{I}_K, \quad \mu \rightarrow 0.$$

Число λ_k подбирается каждую итерацию при помощи одномерной оптимизации. Такие алгоритмы подбора λ_k вычислительно проще обращения матрицы.

Кроме того, практическое применение находит демпфированный метод Ньютона [nesterov], представленный алгоритмом 1. Идея модификации заключается в том, чтобы на каждой итерации уменьшать шаг алгоритма до тех пор, пока значение целевой функции не станет меньше, чем при исходном значении шага.

На вход демпфированного метода Ньютона поступают векторы сигнала передатчика \mathbf{x} и принимаемого \mathbf{d} сигналов, начальные значения параметров модели \mathbf{z} , исходное значение шага μ и желаемое число итераций P . Функция `model(·)` выдает вектор сигналов на выходе адаптивной модели на основе текущих параметров и входа модели.

Далее по методу Ньютона вычисляется вектор шага коэффициентов, после чего определяются новые значения коэффициентов с текущим значением шага μ .

На основе новых параметров получаем выход модели и новое отклонение. Если значение критерия MSE стало меньше после обновления коэффициентов, то шаг μ удваивается. Такое увеличение шага будет происходить пока он меньше единицы. Параметры запоминаются, алгоритм переходит в новую итерацию.

Если значение критерия MSE ухудшилось, алгоритм будет уменьшать значение шага μ до тех пор, пока значение критерия не станет лучше исходного. После такой процедуры алгоритм запомнит значения параметров и запустится новая итерация.

Ввиду того, что в задачах нелинейной обработки сигналов задача вообще говоря не является квадратичной, методу Ньютона понадобится больше одной итерации для поиска точки оптимума.

3.3 Смешанный метод Ньютона

Смешанный метод Ньютона — метод второго порядка, предназначенный для обучения моделей с комплексными параметрами путём минимизации (??). Для корректной работы алгоритма вектор ошибки \mathbf{e} (??) должен быть голоморфным относительно параметров модели \mathbf{z} , то есть должен не зависеть от сопряжённых параметров \mathbf{z}^* : $\mathbf{e} = \mathbf{e}(\mathbf{x}, \mathbf{z})$. В этом случае MNM обладает свойством отталкивания от седловых точек [mixed_newton].

Algorithm 1: Damped Newton method

Data: x, d, z, P, μ
Result: Obtain z

```

1 for  $i \in 0, P-1$  do
2    $y_{prev} = \text{model}(z, x)$ ;
3    $e_{prev} = d - y_{prev}$ ;
4    $\Delta z = H_J^{-1} \nabla_J(z)$ ;
5    $z_{tmp} = z + \mu \Delta z$ ;
6    $y_{curr} = \text{model}(z_{tmp}, x)$ ;
7    $e_{curr} = d - y_{curr}$ ;
8   if  $MSE(e_{prev}) - MSE(e_{curr}) \geq 0$  then
9      $\mu = \mu * 2$ ;
10    if  $\mu > 1$  then
11       $\mu = 1$ ;
12     $z = z_{tmp}$ ,
13  else
14    flag = True;
15    while flag do
16       $\mu = \mu / 1.5$ ;
17       $z_{tmp1} = z + \mu \Delta z$ ;
18       $y_{div} = \text{model}(z_{tmp1}, x)$ ;
19       $e_{div} = d - y_{div}$ ;
20      if  $MSE(e_{prev}) - MSE(e_{div}) \geq 0$  then
21        flag = False;
22         $z = z_{div}$ 

```

Поскольку алгоритм MNM использует только смешанный гессиан $H_{z^*,z} J$ для обучения параметров модели `[mixed_newton]`, то разностное уравнения смешанного метода Ньютона может быть представлено следующим выражением:

$$z_{k+1} = z_k - \mu (H_{z^*,z} J)^{-1} (D_{z^*} J)^T, \quad (3.3.1)$$

где J — MSE, определенная выражением (1.5.1).

В то же время, согласно обозначению (3.1.1) и выражению для градиента (??), смешанный гессиан можно рассчитать следующим способом:

$$H_{z^*,z} J = D_z (D_{z^*} J)^T = -D_z ((D_z y)^H e). \quad (3.3.2)$$

Поскольку вектор выхода модели y является голоморфным относительно параметров модели, то смешанный гессиан может быть представлен как:

$$H_{z^*,z} J = (D_z y)^H D_z y. \quad (3.3.3)$$

Таким образом, учитывая выражения (??), (3.3.3) и (3.3.1), алгоритм смешанного

метода Ньютона можно представить следующим разностным уравнением:

$$\mathbf{z}_{k+1} = \mathbf{z}_k - \mu((D_{\mathbf{z}} \mathbf{y})^H D_{\mathbf{z}} \mathbf{y})^{-1} (D_{\mathbf{z}} \mathbf{y})^H \mathbf{e}. \quad (3.3.4)$$

Отдельно отметим, что сложность хранения гессиана оценивается как $o(M^2)$ [**matrix_comp**], если $M \times M$ – размерность гессиана. Обращение гессиана может быть сведено к решению системы из M линейных уравнений. Вычислительная сложность такой операции может быть оценена как $o(M^3)$ [**matrix_comput**].

Кроме того, отметим, что разностное уравнение (??) метода Ньютона применительно к модели Гаммерштейна также представляет собой итеративную LS-оценку.

3.4 Метод LS

Метод LS применим только к однослойным моделям, когда MSE задаётся квадратичной функцией, минимизация которой сводится к решению линейной системы [**convex_opt**]. Поэтому LS будем применять к адаптации КИХ-фильтра и модели нелинейности, заданной блоком PLA.

Рассмотрим адаптацию линейного КИХ-фильтра порядка M с коэффициентами \mathbf{w} . Оптимальные коэффициенты с точки зрения среднего квадрата ошибки MSE задаются выражаются уравнением Винера-Хопфа [**adapt_filt_haykin**]:

$$\mathbf{w}_{opt} = \mathbf{R}_{xx}^{-1} \mathbf{r}_{dx}, \quad (3.4.1)$$

где \mathbf{R}_{xx} – корреляционная матрица сигнала (5.1.12), а \mathbf{r}_{dx} – вектор взаимной корреляции, который определяется:

$$\mathbf{r}_{dx} = \mathbb{E}(\mathbf{u}_n^* d_n), \quad (3.4.2)$$

где \mathbf{u}_n – вектор состояния фильтра (5.1.3).

LS (Least Squares)-оценка является решением уравнения Винера-Хопфа в блочном режиме, то есть при ограниченной длине выборки.

Определим задачу поиска оптимальных коэффициентов линейного КИХ-фильтра как задачу поиска решения системы линейных уравнений:

$$\begin{cases} \mathbf{d} = \mathbf{U} \mathbf{w}_{opt}, \\ \mathbf{e} = \mathbf{d} - \mathbf{U} \mathbf{w}, \\ \mathbf{w}_{opt} = \arg \min_{\mathbf{w}} (\mathbf{e}^H \mathbf{e}) \end{cases} \quad (3.4.3)$$

$J(\mathbf{w}) = \mathbf{e}^H \mathbf{e}$ – вещественнозначная функция комплексного вектора \mathbf{w} . Градиент такой функции равен [**wirt_deriv_tech_rep**]:

$$\nabla_J(\mathbf{w}) \equiv \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}^H} = \left(\frac{\partial J(\mathbf{w})}{\partial w_0^*} \quad \frac{\partial J(\mathbf{w})}{\partial w_1^*} \quad \dots \quad \frac{\partial J(\mathbf{w})}{\partial w_{M-1}^*} \right)^T, \quad (3.4.4)$$

$$\nabla_J^H(\mathbf{w}) \equiv \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = \left(\frac{\partial J(\mathbf{w})}{\partial w_0} \quad \frac{\partial J(\mathbf{w})}{\partial w_1} \quad \dots \quad \frac{\partial J(\mathbf{w})}{\partial w_{M-1}} \right), \quad (3.4.5)$$

где под выражением $\frac{\partial J(\mathbf{w})}{\partial w_i^*} \forall i = 0, M-1$ понимается производная Виртингера функции комплексных переменных [wirt_deriv_book].

Ввиду того, что средний квадрат ошибки для однослойной модели, в данном случае КИХ-фильтра, является квадратичной функцией, то глобальным минимум метрики MSE может быть найден следующий образом:

$$\left. \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}^H} \right|_{\mathbf{w}=\mathbf{w}_{opt}} = \mathbf{0} \quad (3.4.6)$$

Найдём экстремум функции MSE для уравнения (3.4.3):

$$\nabla_J(\mathbf{w}) = \frac{\partial}{\partial \mathbf{w}^H} (\mathbf{d} - \mathbf{U}\mathbf{w})^H (\mathbf{d} - \mathbf{U}\mathbf{w}) = \quad (3.4.7)$$

$$= \frac{\partial}{\partial \mathbf{w}^H} (\mathbf{d}^H - \mathbf{w}^H \mathbf{U}^H) (\mathbf{d} - \mathbf{U}\mathbf{w}) = \quad (3.4.8)$$

$$= -\mathbf{U}^H (\mathbf{d} - \mathbf{U}\mathbf{w}) = \mathbf{U}^H \mathbf{U}\mathbf{w} - \mathbf{U}^H \mathbf{d} = \mathbf{0} \quad (3.4.9)$$

$$\hat{\mathbf{w}}_{opt} = (\mathbf{U}^H \mathbf{U})^{-1} \mathbf{U}^H \mathbf{d}, \quad (3.4.10)$$

где $\hat{\mathbf{w}}_{opt}$ – оценка оптимальных коэффициентов. Отметим, что вектор:

$$\check{\mathbf{r}}_{dx} = \mathbf{U}^H \mathbf{d} \quad (3.4.11)$$

представляет собой смещенную оценку вектора взаимной корреляции. В то же время, как отмечалось ранее (5.1.20), матрица:

$$\check{\mathbf{R}}_{xx} = \mathbf{U}^H \mathbf{U} \quad (3.4.12)$$

представляет собой смещенную оценку матрицы корреляции.

Таким образом, оценки оптимальных коэффициентов КИХ-фильтра и блока PLA имеют вид:

$$\hat{\mathbf{w}}_{opt} = \check{\mathbf{R}}_{xx}^{-1} \check{\mathbf{r}}_{dx}, \quad (3.4.13)$$

$$\hat{\mathbf{h}}_{opt} = \check{\mathbf{R}}_{vv}^{-1} \check{\mathbf{r}}_{dv}. \quad (3.4.14)$$

Уравнения (3.4.13), (3.4.14) отражают связь между LS-оценкой и уравнением Винера-Хопфа для поиска оптимальных коэффициентов линейного КИХ-фильтра (3.4.1).

Отметим, что метод LS подразумевает накопление корреляционной матрицы, её обращение и выполнение операции матрично-векторного умножения. Такой метод поиска оптимальных коэффициентов трудный с точки зрения вычислительной сложности, однако применяется для оценки оптимальных коэффициентов моделей ввиду сходимости к вектору оптимальных коэффициентов за одну итерацию за счёт квадратичности целевой функции.

Метод LS неприменим к модели Гаммерштейна, однако существует итеративная модификация метода LS. Эта модификация по сути представляет собой метод Ньютона [nesterov], который будет описан далее.

4. Методы адаптации моделей нелинейных искажений не требующие явного вычисления матрицы Гессе

Методы второго порядка подразумевают необходимость накапливать матрицу вторых производных для каждого нового блока данных, хранить в памяти, а также обращаться, обеспечивая при этом численную устойчивость процедуры инверсии матрицы [matrix_comput]. Такие методы являются вычислительно сложными, кроме того достаточно трудными с точки зрения реализации в целочисленной арифметике. В связи с этим в теории нелинейной обработки сигналов предлагается использовать методы применяющие приближенное вычисление обратной матрицы вторых производных либо приближенное вычисление произведения обратной матрицы и градиента функции потерь.

4.1 Метод сопряженных градиентов

4.2 Метод DCD

Метод покоординатного спуска (англ. DCD – Dichotomous Coordinate Descent) является методом поиска решения систем линейных уравнений с квадратной матрицей и не относится к методам первого порядка [dcd_alg_thesis]. В случае КИХ-фильтра и блоков PLA такие СЛАУ будут иметь вид (4.2.3), (4.2.4):

$$\mathbf{d} = \mathbf{U} \mathbf{w}_{opt}, \quad (4.2.1)$$

$$\mathbf{U}^H \mathbf{d} = \mathbf{U}^H \mathbf{U} \mathbf{w}_{opt}, \quad (4.2.2)$$

$$\check{\mathbf{r}}_{xd} = \check{\mathbf{R}}_{xx} \mathbf{w}_{opt}, \quad (4.2.3)$$

$$\check{\mathbf{r}}_{vd} = \check{\mathbf{R}}_{vv} \mathbf{h}_{opt}. \quad (4.2.4)$$

Ввиду эквивалентности (4.2.3) и (4.2.4) с точки зрения решения системы линейных уравнений введём следующие обозначения:

$$\begin{array}{ll}
 \text{КИХ-фильтр:} & \text{блок PLA:} \\
 \check{r}_{xd} \equiv b & \check{r}_{vd} \equiv b \\
 \check{R}_{xx} \equiv R & \check{R}_{vv} \equiv R \\
 w \equiv h & h \equiv h \\
 \check{r}_{xd} - \check{R}_{xx}w \equiv r & \check{r}_{vd} - \check{R}_{vv}h \equiv r \\
 M \equiv P & L \equiv P
 \end{array} \tag{4.2.5}$$

Идея алгоритма заключается в том, чтобы упростить процедуру обращения матрицы. Для этого на каждой итерации предлагается выделять наибольший по модулю элемент в векторе ошибки \mathbf{r} , обнулять его и пересчитывать вектор коэффициентов, двигаясь. Таким образом, по направлению убывания среднего квадрата ошибки вдоль соответствующей координаты вектора коэффициентов. Метод DCD для вещественнозначных векторов и матриц (4.2.5) описан в алгоритме 2 [dcd_alg_thesis].

Algorithm 2: Real-valued leading DCD algorithm

Data: R, r, K_1, K_2
Result: Obtain $\Delta h, r$

- 1 Initialization:
- 2 $\Delta h = \mathbf{0}, \alpha = H, m = 1$
- 3 **for** $i \in \overline{0, K_1}$ **do**
- 4 $k = \arg \max_{p=\overline{0, P-1}} \{|r_p|\}$, go to line 8;
- 5 $m = m + 1, \alpha = \alpha/2$;
- 6 **if** $m > K_2$ **then**
- 7 algorithm stops;
- 8 **if** $|r_k| \leq (\frac{\alpha}{2})R_{k,k}$ **then**
- 9 go to line 5;
- 10 $\Delta h_k = \Delta h_k + \text{sign}(r_k)\alpha$;
- 11 $\mathbf{r} = \mathbf{r} - \text{sign}(r_n)\alpha \mathbf{R}_{:,k}$

Константы K_1, K_2 задаются пользователем. K_1 определяет количество шагов в направлении уменьшения среднего квадрата ошибки, K_2 задаёт число возможных дроблений шага α на каждой итерации перед внесением изменения в вектор приращения Δh .

Выбирается индекс k наибольшей по модулю координаты невязки \mathbf{r}_k . Далее алгоритм уменьшает шаг α до тех пор, пока условие строки 8 не станет ложным. Таким образом, шаг α будет уменьшаться до тех пор, пока приращение невязки $\alpha \mathbf{R}_{:,k}$ не станет меньше модуля самой невязки. Это необходимо для устойчивости алгоритма.

Далее вносится приращение в вектор невязки \mathbf{r} . Для того, чтобы преобразование было равносильным, из вектора коэффициентов \mathbf{h} необходимо вычесть поправ-

ку $\text{sign}(r_n)\alpha$. Для этого в вектор приращения коэффициентов $\Delta\mathbf{h}$ вносится поправка $\text{sign}(r_n)\alpha$ по координате с индексом k .

Рассмотрим алгоритм DCD для комплекснозначных векторов и матриц (4.2.5). Такой алгоритм можно получить сведением системы комплекснозначных уравнений к системе вещественнозначных уравнений и воспользовавшись алгоритмом 2. Сгруппируем комплекснозначные вектора и матрицы следующим образом:

$$\begin{aligned}\mathbf{R} &= \mathbf{R}_1 + j\mathbf{R}_2, \quad \mathbf{R} \in \mathbb{C}^{P/2 \times P/2} \\ \mathbf{h} &= \mathbf{h}_1 + j\mathbf{h}_2, \quad \mathbf{h} \in \mathbb{C}^{P/2 \times 1} \\ \mathbf{b} &= \mathbf{b}_1 + j\mathbf{b}_2, \quad \mathbf{b} \in \mathbb{C}^{P/2 \times 1},\end{aligned}\tag{4.2.6}$$

где $\mathbf{R}_1, \mathbf{R}_2, \mathbf{h}_1, \mathbf{h}_2, \mathbf{b}_1, \mathbf{b}_2$ – вещественнозначные объекты. Тогда:

$$\mathbf{R}\mathbf{h} = (\mathbf{R}_1\mathbf{h}_1 - \mathbf{R}_2\mathbf{h}_2) + j(\mathbf{R}_2\mathbf{h}_1 - \mathbf{R}_1\mathbf{h}_2) = \mathbf{c}_1 + j\mathbf{c}_2,\tag{4.2.7}$$

$$\begin{pmatrix} \mathbf{R}_1\mathbf{h}_1 - \mathbf{R}_2\mathbf{h}_2 \\ \mathbf{R}_2\mathbf{h}_1 - \mathbf{R}_1\mathbf{h}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{pmatrix},\tag{4.2.8}$$

$$\begin{pmatrix} \mathbf{R}_1 & -\mathbf{R}_2 \\ \mathbf{R}_2 & \mathbf{R}_1 \end{pmatrix} \begin{pmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix}.\tag{4.2.9}$$

Из матричного уравнения (4.2.9) следует, что решение системы $P/2$ комплекснозначных уравнений с $P/2$ переменными эквивалентно решению P вещественнозначных уравнений с P переменными. Исходя из этого DCD алгоритм для комплекснозначной системы размерности $P/2 \times P/2$ может быть представлен, как алгоритм 3 [dcd_alg_thesis].

Сначала производится поиск индекса k наибольшего модуля среди вещественных и мнимых частей элементов вектора отклонения \mathbf{r} . В переменную s записывается 1, если в наибольшей по модулю является вещественная часть одной из координат \mathbf{r} , $s = j$, если мнимая. Эта переменная учитывает вещественный или мнимый характер приращения невязки и поправки на вектор коэффициентов.

Затем, как и для вещественнозначного случая, шаг алгоритма уменьшается до тех пор, пока приращение вектора невязки \mathbf{r} не станет меньше модуля самой невязки. После чего производится пересчёт вектора приращения коэффициентов $\Delta\mathbf{h}$ и вектора невязки \mathbf{r} .

В результате алгоритм 3 выдаёт вектор невязки \mathbf{r} и вектор приращения коэффициентов $\Delta\mathbf{h}$, после чего производится обновление коэффициентов.

Алгоритм является методом приближенного решения системы линейных уравнений и работает интеративно. В связи с приближенным характером решения СЛАУ вблизи точки оптимума процедура DCD не гарантирует монотонной сходимости.

Блочный алгоритм DCD применяется к модели Гаммерштейна так же как и блочная реализация градиентного спуска. Выход модели представляется в виде векторно-

Algorithm 3: Complex-valued leading DCD algorithm

Data: $\mathbf{R}, \mathbf{r}, K_1, K_2$
Result: Obtain $\Delta \mathbf{h}, \mathbf{r}$
 1 Initialization:
 2 $\Delta \mathbf{h} = \mathbf{0}, \alpha = H, m = 1$
 3 **for** $i \in \overline{0, K_1}$ **do**
 4 $[k, s] = \arg \max_{p=\overline{0, (P-1)/2}} \{|Re(r_p)|, |Im(r_p)|\}$, go to line 12;
 5 $m = m + 1, \alpha = \alpha/2$;
 6 **if** $m > K_2$ **then**
 7 algorithm stops;
 8 **if** $s = 1$ **then**
 9 $r_{tmp} = Re(r_k)$
 10 **else**
 11 $r_{tmp} = Im(r_k)$
 12 **if** $|r_{tmp}| \leq (\frac{\alpha}{2})R_{k,k}$ **then**
 13 go to line 5;
 14 $\Delta h_k = \Delta h_k + \text{sign}(r_k)s\alpha$;
 15 $\mathbf{r} = \mathbf{r} - \text{sign}(r_n)s\alpha \mathbf{R}_{:,k}$

матричного произведения (??) составной матрицы и вектора (??). На вход алгоритма DCD в данном случае поступают матрица \mathbf{R} и вектор \mathbf{b} следующего вида:

$$\mathbf{R} = \mathbf{V}_{full}^H \mathbf{V}_{full} = \begin{pmatrix} \mathbf{V}_f^H \\ \mathbf{U}^H \end{pmatrix} \begin{pmatrix} \mathbf{V}_f & \mathbf{U} \end{pmatrix} = \begin{pmatrix} \mathbf{V}_f^H \mathbf{V}_f & \mathbf{V}_f^H \mathbf{U} \\ \mathbf{U}^H \mathbf{V}_f & \mathbf{U}^H \mathbf{U} \end{pmatrix}, \quad (4.2.10)$$

$$\mathbf{b} = \mathbf{V}_{full}^H \mathbf{d} = \begin{pmatrix} \mathbf{V}_f^H \\ \mathbf{U}^H \end{pmatrix} \mathbf{d}. \quad (4.2.11)$$

4.3 Квазиньютоновские методы

Как отмечалось ранее, метод Ньютона обладает локальной квадратичной сходимостью, высокой точностью решения. Кроме того он является аффинно инвариантным [nesterov], что означает, что вид разностного уравнения и скорость сходимости не меняются при аффинном преобразовании координат. Однако, для реализации метода требуется хранение гессиана в памяти, также операция обращения гессиана, которая является вычислительно сложной.

Для снижения вычислительной сложности применяют методы, которые обладают скоростью сходимости выше, чем у градиентного спуска, имея при этом накладные расходы меньше, чем у метода Ньютона. Квазиньютоновские методы объединяют в себе простоту градиентного спуска с точки зрения вычислительной сложности и высокую скорость сходимости.

Метод Ньютона раскладывает целевую функцию в окрестности точки \mathbf{w}_{n-1} (??):

$$p(\mathbf{w}_n) = J(\mathbf{w}_{n-1}) + (\mathbf{w}_n - \mathbf{w}_{n-1})^H \nabla J(\mathbf{w}_{n-1}) + \frac{1}{2}(\mathbf{w}_n - \mathbf{w}_{n-1})^H \mathbf{H}_J(\mathbf{w}_{n-1})(\mathbf{w}_n - \mathbf{w}_{n-1}). \quad (4.3.1)$$

Идея квазиньютоновских алгоритмов в том, чтобы заменить матрицу Гессе на её оценку:

$$\mathbf{H}_J \rightarrow \mathbf{B}_n, \quad \mathbf{B}_n \succ 0. \quad (4.3.2)$$

Отметим, что искомая оценка гессиана должна быть положительно определённой. Выполним замену:

$$\mathbf{w}_n - \mathbf{w}_{n-1} = \Delta \mathbf{w}_{n-1}, \quad (4.3.3)$$

тогда из (4.3.1), (4.3.2) и (4.3.3) получим:

$$p(\Delta \mathbf{w}_{n-1} + \mathbf{w}_{n-1}) = J(\mathbf{w}_{n-1}) + \Delta \mathbf{w}_{n-1}^H \nabla J(\mathbf{w}_{n-1}) + \frac{1}{2} \Delta \mathbf{w}_{n-1}^H \mathbf{B}_{n-1} \Delta \mathbf{w}_{n-1}. \quad (4.3.4)$$

По аналогии с методом Ньютона (??) квазиньютоновский метод будет иметь вид:

$$\mathbf{w}_n = \mathbf{w}_{n-1} - \mu \mathbf{B}_{n-1}^{-1} \nabla J(\mathbf{w}_{n-1}) = \mathbf{w}_{n-1} - \mu \mathbf{H}_{n-1} \nabla J(\mathbf{w}_{n-1}), \quad (4.3.5)$$

где $\mathbf{B}_n, \mathbf{H}_n$ – оценки прямого и обратного гессиана соответственно.

Требования, которые предъявляются к оценке гессиана:

- Быстрое обновление $\mathbf{B}_{n-1} \rightarrow \mathbf{B}_n$ в условиях, когда доступны только градиенты
- Быстрый поиск направления:

$$\Delta \mathbf{w}_{n-1} = \mu \mathbf{B}_{n-1}^{-1} \nabla J(\mathbf{w}_{n-1}) = \mu \mathbf{H}_{n-1} \nabla J(\mathbf{w}_{n-1}) \quad (4.3.6)$$

- Компактное хранение \mathbf{B}_n
- Сверхлинейная сходимость

Получим уравнение на поиск матрицы \mathbf{B}_n . Для этого сформулируем так называемое правило двух градиентов [fletcher]:

$$\left. \frac{\partial p(\Delta \mathbf{w}_n + \mathbf{w}_n)}{\partial \Delta \mathbf{w}_n^H} \right|_{\Delta \mathbf{w}_n = \mathbf{0}} = \nabla J(\mathbf{w}_n), \quad (4.3.7)$$

$$\left. \frac{\partial p(\Delta \mathbf{w}_n + \mathbf{w}_n)}{\partial \Delta \mathbf{w}_n^H} \right|_{\Delta \mathbf{w}_n = -\mu \Delta \mathbf{w}_{n-1}} = \nabla J(\mathbf{w}_{n-1}). \quad (4.3.8)$$

Требование (4.3.7) следует напрямую из (4.3.4). Вычислим производную в левой части

(4.3.8):

$$\left. \frac{\partial p(\Delta \mathbf{w}_n + \mathbf{w}_n)}{\partial \Delta \mathbf{w}_n^H} \right|_{\Delta \mathbf{w}_n = -\mu \Delta \mathbf{w}_{n-1}} = \nabla_J(\mathbf{w}_n) - \frac{1}{2} \mathbf{B}_n \Delta \mathbf{w}_n = \nabla_J(\mathbf{w}_n) - \mu \mathbf{B}_n \Delta \mathbf{w}_{n-1} \quad (4.3.9)$$

Требование (4.3.8) приводит к квазиньютоновскому уравнению (англ. Secant equation):

$$\nabla_J(\mathbf{w}_n) - \nabla_J(\mathbf{w}_{n-1}) = \mu \mathbf{B}_n \Delta \mathbf{w}_{n-1} \quad (4.3.10)$$

Сделаем замену и получим квазиньютоновское уравнение в следующем виде:

$$\mathbf{s}_{n-1} = \mu(\mathbf{w}_n - \mathbf{w}_{n-1}), \quad (4.3.11)$$

$$\mathbf{f}_{n-1} = \nabla_J(\mathbf{w}_n) - \nabla_J(\mathbf{w}_{n-1}), \quad (4.3.12)$$

$$\mathbf{B}_n \mathbf{s}_{n-1} = \mathbf{f}_{n-1}. \quad (4.3.13)$$

В уравнении (4.3.13) векторы $\mathbf{s}_{n-1}, \mathbf{f}_{n-1}$ определяются в процессе вычислений. Квазиньютоновское уравнение является уравнением на поиск матрицы \mathbf{B}_n .

Отметим, что ввиду требования положительной определённости \mathbf{B}_n (4.3.2), уравнение (4.3.13) не будет иметь решение в случае $\mathbf{s}_{n-1}^T \mathbf{f}_{n-1} < 0$. При выборе шага важно следить за тем, чтобы это произведение было неотрицательным.

Кроме того, пусть система (4.3.13) имеет F уравнений ($F = M + L$ для модели Гаммерштейна), тогда она будет иметь $\frac{F(F-1)}{2}$ неизвестных ввиду симметричности оценки гессиана. Это означает, что требуются дополнительные условия для обеспечения единственности решения.

Для устойчивости алгоритма соседние оценки гессиана $\mathbf{B}_{n-1}, \mathbf{B}_n$ должно быть близки друг к другу.

Далее рассмотрим такие примеры квазиньютоновских методов, как DFP, BFGS [`pract_optim`] и Barzilai-Borwein [`bb_method`].

4.4 Метод Barzilai-Borwein

Данный метод аппроксимирует гессиан диагональной матрицей. Пусть шаг алгоритма – величина переменная, подстраиваемая на каждой новой итерации. Распишем алгоритм градиентного спуска:

$$\begin{aligned} \mathbf{w}_n &= \mathbf{w}_{n-1} - \mu_{n-1} \nabla_J(\mathbf{w}_{n-1}) = \mathbf{w}_{n-1} - \mu_{n-1} \mathbf{I} \nabla_J(\mathbf{w}_{n-1}) = \\ &= \mathbf{w}_{n-1} - \left(\frac{1}{\mu_{n-1}} \mathbf{I} \right)^{-1} \nabla_J(\mathbf{w}_{n-1}) = \mathbf{w}_{n-1} - \mathbf{B}_{n-1}^{-1} \nabla_J(\mathbf{w}_{n-1}), \end{aligned} \quad (4.4.1)$$

Квазиньютоновское уравнение будет иметь вид приближенного равенства, поскольку $\mathbf{s}_{n-1}, \mathbf{f}_{n-1}$ заранее известны, а \mathbf{B}_n приближается всего одним числом, чего не доста-

точно для строгого равенства:

$$\mu_{n-1}^{-1} \mathbf{s}_{n-2} \approx \mathbf{f}_{n-2}, \quad (4.4.2)$$

где приближенное равенство формализуется следующей задачей минимизации `[bb_method]`:

$$\min_{\mu_{n-1}} \left\| \mathbf{s}_{n-2} - \mu_{n-1} \mathbf{f}_{n-2} \right\|_2 \Rightarrow \mu_{n-1} = \frac{\mathbf{s}_{n-2}^T \mathbf{f}_{n-2}}{\mathbf{f}_{n-2}^T \mathbf{f}_{n-2}}. \quad (4.4.3)$$

Выражение (4.4.3) задаёт способ обновления шага.

Данный метод не является градиентный спуском, поскольку учитывает два предыдущих вектора коэффициентов и градиента для получения нового вектора коэффициентов.

4.5 Метод DFP

Метод DFP (англ. Davidon-Fletcher-Powell) использует поиск оценки матрицы Гёссе на каждой новой итерации `[pract_optimiz]`. Учтём ранее упомянутое требование, которое заключается в близости оценок гессианов на соседних итерациях. Формульно такая задача может быть поставлена следующим образом `[pract_optimiz]`:

$$\begin{cases} \min_{\mathbf{B}_n} \left\| \mathbf{B}_{n-1} - \mathbf{B}_n \right\|_2, \\ \mathbf{B}_n = \mathbf{B}_n^T, \\ \mathbf{B}_n \mathbf{s}_{n-1} = \mathbf{f}_{n-1}. \end{cases} \quad (4.5.1)$$

Аналитическое решение задачи (4.5.1) имеет вид `[fletcher]`:

$$\mathbf{B}_n = (\mathbf{I} - \rho_{n-1} \mathbf{f}_{n-1} \mathbf{s}_{n-1}^T) \mathbf{B}_{n-1} (\mathbf{I} - \rho_{n-1} \mathbf{s}_{n-1} \mathbf{f}_{n-1}^T) + \rho_{n-1} \mathbf{f}_{n-1} \mathbf{f}_{n-1}^T, \quad (4.5.2)$$

$$\rho_{n-1} = \frac{1}{\mathbf{f}_{n-1}^T \mathbf{s}_{n-1}}. \quad (4.5.3)$$

По формуле Шермана-Вудбери-Моррисона `[matrix_calc]` (лемма об обращении матриц) решение (4.5.2) можно преобразовать:

$$\mathbf{B}_n^{-1} \equiv \mathbf{H}_n = \mathbf{H}_{n-1} - \frac{\mathbf{H}_{n-1} \mathbf{f}_{n-1} \mathbf{f}_{n-1}^T \mathbf{H}_{n-1}}{\mathbf{f}_{n-1}^T \mathbf{H}_{n-1} \mathbf{f}_{n-1}} + \frac{\mathbf{s}_{n-1} \mathbf{s}_{n-1}^T}{\mathbf{f}_{n-1}^T \mathbf{s}_{n-1}}. \quad (4.5.4)$$

Формула (4.5.4) задаёт аналитическое выражение для вычисления оценки обратного гессиана. Для вычисления вектора коэффициентов на новой итерации эта оценка подставляется в разностное уравнение (4.3.5).

Сложность хранения оценки обратной матрицы может быть оценена как $o(M^2)$ – такая же как и для метода Ньютона. Оценим вычислительную сложность пересчета оценки гессиана. В выражении (4.5.4) сложность подсчета третьего слагаемого оце-

нивается как $o(M^2)$, поскольку $\mathbf{s}_{n-1}\mathbf{s}_{n-1}^T$ представляет собой векторное произведение, в результате которого получается матрица размерности $M \times M$.

Второе слагаемое также вычисляется за $o(M^2)$. Выражение $\mathbf{H}_{n-1}\mathbf{f}_{n-1}$ вычисляется за $o(M^2)$. При этом $\mathbf{f}_{n-1}^T\mathbf{H}_{n-1} = (\mathbf{H}_{n-1}\mathbf{f}_{n-1})^T$ ввиду симметричности \mathbf{H}_{n-1} . Остается произвести векторное умножение в числителе, сложность которого $o(M^2)$, а в знаменателе скалярное умножение векторов, сложность которого $o(M)$.

4.6 Метод BFGS

Метод BFGS (англ. Broyden–Fletcher–Goldfarb–Shanno). Вместо того, чтобы искать оценку гессиана, напрямую поставим задачу поиска оценки обратного гессиана `[pract_optimiz]`:

$$\begin{cases} \min_{\mathbf{H}_n} \|\mathbf{H}_{n-1} - \mathbf{H}_n\|_2, \\ \mathbf{H}_n = \mathbf{H}_n^T, \\ \mathbf{H}_n\mathbf{f}_{n-1} = \mathbf{s}_{n-1}. \end{cases} \quad (4.6.1)$$

Аналитическое решение задачи (4.6.1) записывается в виде выражения `[bfgs_fletcher]`:

$$\mathbf{H}_n = (\mathbf{I} - \rho_{n-1}\mathbf{s}_{n-1}\mathbf{f}_{n-1}^T)\mathbf{H}_{n-1}(\mathbf{I} - \rho_{n-1}\mathbf{f}_{n-1}\mathbf{s}_{n-1}^T) + \rho_{n-1}\mathbf{s}_{n-1}\mathbf{s}_{n-1}^T, \quad (4.6.2)$$

$$\rho_{n-1} = \frac{1}{\mathbf{f}_{n-1}^T\mathbf{s}_{n-1}}. \quad (4.6.3)$$

BFGS обладает локальной сверхлинейной сходимостью.

На практике приведенный метод является наиболее предпочтительным квазиньютоновским методом поскольку обладает свойством самокоррекции.

Сложность хранения и обращения гессиана оценивается как $o(M^2)$, что следует из формулы (4.6.2).

Отметим, что для вычисления коэффициентов на новой итерации необходима не сама матрица Гёссе (или обратная к ней), а эффективная процедура умножения матрицы на вектор $\nabla_J(\mathbf{w}_{n-1})$. Кроме того, значения векторов \mathbf{f}, \mathbf{s} , полученные на первых итерациях могут портить оценки \mathbf{B}, \mathbf{H} на более поздних итерациях. Для подсчет оценки обратного гессиана предлагается хранить в очереди и использовать последние $m \ll M$ значений векторов \mathbf{f}, \mathbf{s} . На этой идее основана модификация Limited-Memory BFGS `[lbfgs]`.

5. Математические модели линейных и нелинейных искажений в системах связи

В этом разделе изложена теория, необходимая для описания слоёв модели Гаммерштейна, а также алгоритмы адаптации модели. Алгоритмы адаптации разделены на блочные – методы, применяющие накопление сигнала передатчика для вычисле-

ния шага алгоритма и стохастические – оценивающие градиент целевой функции по единственному отсчёту сигнала ошибки.

Рассматриваются методы первого и второго порядка, а также квазиньютоновские методы адаптации модели нелинейных помех.

Таким образом, обзор математических моделей паразитных помех разбит на следующие части:

- Аппроксимация импульсного отклика канала распространения помехи в чипсете
- Аппроксимация нелинейности усилителя мощности чипсета мобильного терминала

5.1 Линейная модель искажений сигнала в системах связи

КИХ-фильтр описывается набором комплексных коэффициентов $\{w_n \in \mathbb{C}\}_{n=0}^M$, которые можно представить в векторной форме $\mathbf{w} = \{w_n\}_{n=0}^M$, здесь M – порядок фильтра. Выход КИХ-фильтра описывается линейной свёрткой [dsp_layons]:

$$y_n = \sum_{k=0}^M w_k x_{n-k}. \quad (5.1.1)$$

В векторном виде выход КИХ-фильтра будет иметь следующий вид:

$$y_n = \mathbf{u}_n^T \mathbf{w} = \begin{pmatrix} x_n & x_{n-1} & \cdots & x_{n-M} \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_M \end{pmatrix}, \quad (5.1.2)$$

где \mathbf{u}_n – вектор состояния КИХ-фильтра M -ого порядка в момент времени n :

$$\mathbf{u}_n^T = \begin{pmatrix} x_n & x_{n-1} & \cdots & x_{n-M} \end{pmatrix}. \quad (5.1.3)$$

Рассмотрим последовательность из N комплексных отсчетов сигнала на входе фильтра:

$$\mathbf{x} = \{x_n \in \mathbb{C}\}_{n=0}^{N-1}. \quad (5.1.4)$$

Введём матрицу состояния [adapt_filt_haykin] фильтра $\mathbf{U} \in \mathbb{C}^{N \times M+1}$:

$$\mathbf{U} = \begin{pmatrix} \mathbf{u}_0^T \\ \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_{N-2}^T \\ \mathbf{u}_{N-1}^T \end{pmatrix} = \begin{pmatrix} x_{\frac{M-1}{2}} & \cdots & x_1 & x_0 & 0 & \cdots & 0 \\ x_{\frac{M-1}{2}+1} & \cdots & x_2 & x_1 & x_0 & \cdots & 0 \\ \vdots & & & \ddots & & & \vdots \\ 0 & \cdots & x_{N-1} & x_{N-2} & x_{N-3} & \cdots & x_{N-2-\frac{M-1}{2}} \\ 0 & \cdots & 0 & x_{N-1} & x_{N-2} & \cdots & x_{N-1-\frac{M-1}{2}} \end{pmatrix}. \quad (5.1.5)$$

Тогда отсчеты на выходе фильтра можно получить в виде векторно-матричного произведения:

$$\mathbf{y} = \mathbf{U} \mathbf{w}. \quad (5.1.6)$$

Матрица состояния фильтра в уравнении (5.1.6) учитывает задержку фильтра D_{FIR} равную половине длины фильтра. Такой способ выражения выходных отсчетов фильтра предлагается для того чтобы сохранить неизменными размерности векторов $\mathbf{y} \in \mathbb{C}^{N \times 1}$ и $\mathbf{x} \in \mathbb{C}^{N \times 1}$, что в дальнейшем будет необходимо для реализации численных алгоритмов.

Ввиду того, что изначально при симуляции алгоритмов адаптации неизвестна задержка сигнала \mathbf{x} передатчика от сигнала помехи \mathbf{d} , то не имеет значение, какой именно задавать задержку D_{FIR} . Для выравнивания отсчётов передатчика относительно отсчётов приёмника по времени, предлагается задать D_{FIR} равной половине длины фильтра, поскольку такая структура позволит описать как минимальнофазовые, так и линейно-фазовые каналы распространения помехи `[dsp_oppenheim]`. Корреляционная матрица `[adapt_filt_haykin]` является центральным объектом при построении алгоритмов адаптации линейных фильтров. Последовательность комплексных отсчетов входного сигнала является дискретным случайным процессом, автокорреляционная функция `[marpl_spectr]`, которого определяется:

$$r_{xx}(k, l) = \mathbb{E}(x_k x_l^*). \quad (5.1.7)$$

Здесь и далее будем считать, что такой процесс (5.1.4) является стационарным в широком смысле. Одним из необходимых условий стационарности процесса является тот факт, что его автокорреляционная функция не зависит одновременно от двух временных индексов k и l , а определяется только интервалами наблюдения $m = k - l$. Тогда определение (5.1.7) можно переписать следующим образом:

$$r_{xx}(k, l) = r_{xx}(m) = \mathbb{E}(x_{n+m} x_n^*), \quad (5.1.8)$$

где $\mathbb{E}(\cdot)$ – оператор математического ожидания по переменной n . Перечислим некоторые важные свойства автокорреляционной функции:

$$r_{xx}(-m) = r_{xx}^*(m) \quad (5.1.9)$$

$$r_{xx}(0) \geq |r_{xx}^*(m)| \quad \forall m \quad (5.1.10)$$

Опираясь на выражение (5.1.8) определим корреляционную матрицу сигнала, про-

ходящего через КИХ-фильтр порядка M , следующим образом:

$$\mathbf{R}_{xx} = \begin{pmatrix} r_{xx}(0) & r_{xx}(-1) & \cdots & r_{xx}(-M) \\ r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(-M+1) \\ \vdots & & \ddots & \vdots \\ r_{xx}(M) & r_{xx}(M-1) & \cdots & r_{xx}(0) \end{pmatrix}, \quad (5.1.11)$$

используя свойство (5.1.9) получим:

$$\mathbf{R}_{xx} = \begin{pmatrix} r_{xx}(0) & r_{xx}^*(1) & \cdots & r_{xx}^*(M) \\ r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}^*(M-1) \\ \vdots & & \ddots & \vdots \\ r_{xx}(M) & r_{xx}(M-1) & \cdots & r_{xx}(0) \end{pmatrix}. \quad (5.1.12)$$

Отметим, также, что корреляционная матрица является эрмитовой, поскольку из (5.1.12) видно, что:

$$\mathbf{R}_{xx} = \mathbf{R}_{xx}^H \quad (5.1.13)$$

Для численной реализации алгоритмов адаптации введём понятие оценки автокорреляционной функции и корреляционной матрицы. Предположим, как и ранее, что на вход КИХ-фильтра порядка M поступает ограниченная выборка из N комплексных отсчетов сигнала \mathbf{x} в соответствии с (5.1.4). Несмещенная, состоятельная оценка автокорреляционной функции задается выражением `[marpl_spectr]`:

$$\hat{r}_{xx}(m) = \begin{cases} \frac{1}{N-m} \sum_{n=0}^{N-1} x_{n+m} x_n^*, & 0 \leq m < N \\ \frac{1}{N+m} \sum_{n=0}^{N-1} x_{n-m}^* x_n, & -N < m < 0 \end{cases} \quad (5.1.14)$$

Отметим, что более привлекательной и удобной с точки зрения реализации является смещенная оценка автокорреляционной функции (5.1.15):

$$\check{r}_{xx}(m) = \begin{cases} \frac{1}{N} \sum_{n=0}^{N-1} x_{n+m} x_n^*, & 0 \leq m < N \\ \frac{1}{N} \sum_{n=0}^{N-1} x_{n-m}^* x_n, & -N < m < 0 \end{cases}, \quad (5.1.15)$$

поскольку при увеличении диапазона задержек m дисперсия оценки автокорреляционной функции на краях выборочной оценки $\check{r}_{xx}(m)$ не увеличивается в отличие от несмещенной оценки $\hat{r}_{xx}(m)$ `[marpl_spectr]`.

Корреляционную матрицу также можно описать при помощи векторного произведения векторов (5.1.3) состояния КИХ-фильтра M -ого порядка. Обозначим век-

торное произведение:

$$P_n = \left(\mathbf{u}_n \mathbf{u}_n^H \right)^T = \mathbf{u}_n^* \mathbf{u}_n^T. \quad (5.1.16)$$

Распишем подробно:

$$P_n = \left[\begin{pmatrix} x_n^* \\ x_{n-1}^* \\ \vdots \\ x_{n-M}^* \end{pmatrix}, \begin{pmatrix} x_n & x_{n-1} & \cdots & x_{n-M} \end{pmatrix} \right] =$$

$$= \begin{pmatrix} x_n^* x_n & x_n^* x_{n-1} & \cdots & x_n^* x_{n-M} \\ x_{n-1}^* x_n & x_{n-1}^* x_{n-1} & \cdots & x_{n-1}^* x_{n-M} \\ \vdots & & \ddots & \vdots \\ x_{n-M}^* x_n & x_{n-M}^* x_{n-1} & \cdots & x_{n-M}^* x_{n-M} \end{pmatrix}. \quad (5.1.17)$$

Просуммируем каждый элемент матрицы по индексу времени n :

$$\frac{1}{N} \begin{pmatrix} \sum_{n=0}^{N-1} x_n^* x_n & \sum_{n=0}^{N-1} x_n^* x_{n-1} & \cdots & \sum_{n=0}^{N-1} x_n^* x_{n-M} \\ \sum_{n=0}^{N-1} x_{n-1}^* x_n & \sum_{n=0}^{N-1} x_{n-1}^* x_{n-1} & \cdots & \sum_{n=0}^{N-1} x_{n-1}^* x_{n-M} \\ \vdots & & \ddots & \vdots \\ \sum_{n=0}^{N-1} x_{n-M}^* x_n & \sum_{n=0}^{N-1} x_{n-M}^* x_{n-1} & \cdots & \sum_{n=0}^{N-1} x_{n-M}^* x_{n-M} \end{pmatrix} =$$

$$= \begin{pmatrix} \check{r}_{xx}(0) & \check{r}_{xx}^*(1) & \cdots & \check{r}_{xx}^*(M) \\ \check{r}_{xx}(1) & \check{r}_{xx}(0) & \cdots & \check{r}_{xx}^*(M-1) \\ \vdots & & \ddots & \vdots \\ \check{r}_{xx}(M) & \check{r}_{xx}(M-1) & \cdots & \check{r}_{xx}(0) \end{pmatrix} = \check{\mathbf{R}}_{xx} = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{u}_n^* \mathbf{u}_n^T. \quad (5.1.18)$$

Из (5.1.18) следует, что оценка корреляционной матрицы сигнала может быть представлена в виде оценки матожидания векторного произведения векторов состояния фильтра:

$$\check{\mathbf{R}}_{xx} = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{u}_n^* \mathbf{u}_n^T \equiv \check{\mathbb{E}} \left(\mathbf{u}_n^* \mathbf{u}_n^T \right). \quad (5.1.19)$$

С другой стороны корреляционную матрицу (5.1.18) можно представить в виде произведения матриц состояния фильтра (5.1.5):

$$\check{\mathbf{R}}_{xx} = \mathbf{U}^H \mathbf{U} \quad (5.1.20)$$

Выражения (5.1.19) и (5.1.20) демонстрируют важное свойство, связанное с тем, что оценку корреляционной матрицы можно получить двумя способами.

Уравнение (5.1.19) отражает итеративный способ накопления корреляционно мат-

рицы:

$$\check{R}_{xx,n} = \check{R}_{xx,n-1} + \mathbf{u}_{n-1}^* \mathbf{u}_{n-1}^T \quad (5.1.21)$$

В данном случае оценка тем ближе к истинному значению матожидания, чем больше отсчетов сигнала поступило на вход системы. Такой подход связан с реализацией стохастических методов и успешно применяется в случае потоковой обработки сигнала.

В выражении (5.1.20) матрица вычисляется по ограниченной выборке отсчётов сигнала. Такой способ вычисления применяется в случае блочной обработки сигнала и удобен на этапе симуляции алгоритмов.

5.2 Полиномиальная нелинейных искажений в системах связи

Существует множество моделей, аппроксимирующих нелинейность усилителей мощности **[pa_models]**, таких как модель Винера, Гаммерштейна, Винера-Гаммерштейна, а также полиномиальные модели с памятью и без памяти **[dpd_models]**. Перечисленные модели являются следствием упрощения ряда Вольтерра.

Основной моделью паразитных нелинейных помех в приёмном тракте мобильного терминала является модель Гаммерштейна (рис. ??). Поскольку модель содержит КИХ-фильтр, который помимо задержек распространения помехи по различным путям от приёмника к передатчику учитывает инерционность усилителя мощности, то модель усилителя мощности можно выбирать безынерционной. Среди перечисленных моделей нелинейности усилителя безынерционной является полиномиальная модель без памяти.

В данном разделе рассматриваются две основные модели нелинейности: полиномиальная модель без памяти, а также метод улучшения численной устойчивости при адаптации этой модели, и модель кусочно-линейной аппроксимации. Перед тем, как рассмотреть полиномиальную модель нелинейности усилителя мощности, рассмотрим общий способ описания нелинейной амплитудной характеристики.

На вход модели усилителя $g(\cdot)$ в модели Гаммерштейна (рис. ??) поступает последовательность из N отсчетов модуля входного сигнала $\{|x_n|\}_{n=0}^{N-1}$. Общая модель нелинейности усилителя мощности описывается выражением **[dpd_models]**:

$$g(|x_n|) = \sum_{p=0}^P h_p \varphi_p(|x_n|) \quad (5.2.1)$$

где $\{\varphi_p(|x|)\}_{p=0}^P$ – набор базисных функций модели усилителя, P – порядок модели усилителя. На вход КИХ фильтра (рис. ??) поступают отсчеты:

$$z_n = g(|x_n|)x_n^2 = \left[\sum_{p=0}^P h_p \varphi_p(|x_n|) \right] x_n^2, \quad (5.2.2)$$

выражение (5.2.2) учитывает тот факт, что в качестве нелинейной помехи рассматривается вторая гармоника сигнала передатчика, а значит в модели Гаммертшейна (рис. ??) степень $q = 2$.

Рассмотрим полиномиальную модель без памяти, описывающую нелинейность усилителя мощности. Базисные функции полиномиальной модели без памяти имеют вид `[dpd_models]`:

$$\varphi_p(|x_n|) = |x_n|^p, \quad (5.2.3)$$

тогда модель нелинейности будет иметь вид:

$$g(|x_n|) = \sum_{p=0}^P h_p |x_n|^p. \quad (5.2.4)$$

В случае использования данной модели для описания нелинейной характеристики АМ-АМ `[dpd_models]` усилителя мощности на вход КИХ-фильтра в модели Гаммерштейна поступают отсчеты:

$$z_n = g(|x_n|)x_n^2 = \left[\sum_{p=0}^P h_p |x_n|^p \right] x_n^2, \quad (5.2.5)$$

в векторной форме входные отсчёты КИХ-фильтра будут иметь вид:

$$\begin{aligned} z_n &= \mathbf{h}_n^T \mathbf{v}_n = \begin{pmatrix} h_0 & h_1 & \cdots & h_P \end{pmatrix} \begin{pmatrix} x_n^2 \varphi_0(|x_n|) & x_n^2 \varphi_1(|x_n|) & \cdots & x_n^2 \varphi_P(|x_n|) \end{pmatrix}^T = \\ &= \begin{pmatrix} h_0 & h_1 & \cdots & h_P \end{pmatrix} \begin{pmatrix} x_n^2 & x_n^2 |x_n| & \cdots & x_n^2 |x_n|^P \end{pmatrix}^T, \end{aligned} \quad (5.2.6)$$

где \mathbf{v} – вектор состояния модели нелинейности, который вводится по аналогии с вектором состояния фильтра (5.1.3).

Матрицу состояния полиномиальной модели для случая блочной обработки по ограниченной выборке сигнала можно ввести по аналогии с матрицей состояния КИХ-фильтра через вектора состояния (5.1.5):

$$\mathbf{V} = \begin{pmatrix} \mathbf{v}_0^T \\ \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_{N-1}^T \end{pmatrix} = \begin{pmatrix} x_0^2 & x_0^2 |x_0| & \cdots & x_0^2 |x_0|^P \\ x_1^2 & x_1^2 |x_1| & \cdots & x_1^2 |x_1|^P \\ \vdots & & \ddots & \vdots \\ x_{N-1}^2 & x_{N-1}^2 |x_{N-1}| & \cdots & x_{N-1}^2 |x_{N-1}|^P \end{pmatrix}. \quad (5.2.7)$$

Введём понятие корреляционной матрицы полиномиальной модели без памяти $\mathbf{R}_{pp} \in \mathbb{C}^{P+1 \times P+1}$ по аналогии с понятием корреляционной матрицы сигнала (5.1.12). Эта матрица отражает корреляционные связи между значениями базисных функций для каждого нового отсчёта. Оценка корреляционной матрицы полиномиальной модели

без памяти вводится по аналогии с выражением (5.1.18):

$$\begin{aligned}
 \check{R}_{pp} &= \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{v}_n^* \mathbf{v}_n^T = \\
 &= \frac{1}{N} \begin{pmatrix} \sum_{n=0}^{N-1} (x_n^2)^* x_n^2 \varphi_0(|x_n|) \varphi_0(|x_n|) & \cdots & \sum_{n=0}^{N-1} (x_n^2)^* x_n^2 \varphi_0(|x_n|) \varphi_P(|x_n|) \\ \sum_{n=0}^{N-1} (x_n^2)^* x_n^2 \varphi_1(|x_n|) \varphi_0(|x_n|) & \cdots & \sum_{n=0}^{N-1} (x_n^2)^* x_n^2 \varphi_1(|x_n|) \varphi_P(|x_n|) \\ \vdots & \ddots & \vdots \\ \sum_{n=0}^{N-1} (x_n^2)^* x_n^2 \varphi_P(|x_n|) \varphi_0(|x_n|) & \cdots & \sum_{n=0}^{N-1} (x_n^2)^* x_n^2 \varphi_P(|x_n|) \varphi_P(|x_n|) \end{pmatrix} = \\
 &= \frac{1}{N} \begin{pmatrix} \sum_{n=0}^{N-1} (x_n^2)^* x_n^2 & \sum_{n=0}^{N-1} (x_n^2)^* x_n^2 |x_n| & \cdots & \sum_{n=0}^{N-1} (x_n^2)^* x_n^2 |x_n|^P \\ \sum_{n=0}^{N-1} (x_n^2)^* x_n^2 |x_n| & \sum_{n=0}^{N-1} (x_n^2)^* x_n^2 |x_n|^2 & \cdots & \sum_{n=0}^{N-1} (x_n^2)^* x_n^2 |x_n|^{P+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{n=0}^{N-1} (x_n^2)^* x_n^2 |x_n|^P & \sum_{n=0}^{N-1} (x_n^2)^* x_n^2 |x_n|^{P+1} & \cdots & \sum_{n=0}^{N-1} (x_n^2)^* x_n^2 |x_n|^{P^2} \end{pmatrix}. \quad (5.2.8)
 \end{aligned}$$

В выражении (5.2.8) для матрицы, описывающей корреляционные связи между базисными функциями, степень P может принимать сколь угодно большие конечные значения. Неортогональность степенных базисных функций $|x_n|^p$ приводит к ухудшению обусловленности матрицы \check{R}_{pp} с ростом степени P . Максимальный порядок, при котором можно использовать численные алгоритмы при гарантированной численной устойчивости $P = 6$.

Для улучшения обусловленности корреляционной матрицы модели используются ортогональные полиномы. Использование ортогональных полиномов позволяет повысить порядок полинома вплоть до $P = 100$ с сохранением численной устойчивости алгоритмов адаптации. В случае полиномиальной модели без памяти базисные функции будут иметь вид `[dpd_models]`:

$$\varphi_p(|x_n|) = P_p(|x_n|^p). \quad (5.2.9)$$

Полиномы Лежандра `[spec_func]`:

$$P_{p+1}(x) = \frac{2p+1}{p+1} x P_p(x) - \frac{p}{p+1} P_{p-1}(x), \quad (5.2.10)$$

$$P_0(x) = 1, P_1(x) = x,$$

являются ортогональными на отрезке $[-1; 1]$.

Полиномы Чебышёва 1-ого рода `[spec_func]`:

$$T_{p+1}(x) = 2xT_p(x) - T_{p-1}(x), \quad (5.2.11)$$

$$T_0(x) = 1, T_1(x) = x,$$

являются ортогональными на отрезке $[-1; 1]$. Кроме того, полином Чебышёва 1-ого рода степени n меньше всего отклоняется от нуля на отрезке $[-1; 1]$ среди полиномов степени n .

Полиномы Чебышёва 2-ого рода `[spec_func]`:

$$U_{p+1}(x) = 2xU_p(x) - U_{p-1}(x), \quad (5.2.12)$$

$$U_0(x) = 1, U_1(x) = 2x,$$

являются ортогональными на отрезке $[-1; 1]$. Кроме того, интеграл модуля полинома Чебышёва 2-ого рода степени n меньше всего отклоняется от нуля на отрезке $[-1; 1]$ среди полиномов степени n .

Полиномы Эрмита `[spec_func]`:

$$H_{p+1}(x) = 2xH_p(x) - 2pH_{p-1}(x), \quad (5.2.13)$$

$$H_0(x) = 1, H_1(x) = 2x,$$

являются ортогональными на всей числовой оси.

5.3 Модель Вольтерра нелинейных искажений в системах связи

5.4 Модель Гаммерштейна нелинейных искажений в системах связи

Ранее отмечалось, что выходные отсчёты модели Гаммерштейна описываются выражением $(??)$, где функция $g(|x_n|)$ определяется выражениями (5.2.4), $(??)$ для случаев, когда в качестве модели нелинейности амплитудной характеристики усилителя мощности используются полиномиальная модель без памяти и кусочно-линейная модель соответственно.

В модели Гаммерштейна на вход КИХ-фильтра поступает вектор отсчётов \mathbf{z} , элементы которого определяются выражением $(??)$. Матрица состояния фильтра будет

$$U = \begin{pmatrix} z_{D_{FIR}} & \cdots & z_1 & z_0 & 0 & \cdots & 0 \\ z_{D_{FIR}+1} & \cdots & z_2 & z_1 & z_0 & \cdots & 0 \\ \vdots & & & \ddots & & & \vdots \\ \boxed{z_{D_{FIR}+n} & \cdots & z_{n+1} & z_n & z_{n-1} & \cdots & z_{n-D_{FIR}}} \\ \vdots & & & \ddots & & & \vdots \\ \boxed{z_{3D_{FIR}+n} & \cdots & z_{n+2D_{FIR}+1} & z_{n+2D_{FIR}} & z_{n+2D_{FIR}-1} & \cdots & z_{n+D_{FIR}}} \\ \vdots & & & \ddots & & & \vdots \\ 0 & \cdots & z_{N-1} & z_{N-2} & z_{N-3} & \cdots & z_{N-2-D_{FIR}} \\ 0 & \cdots & 0 & z_{N-1} & z_{N-2} & \cdots & z_{N-1-D_{FIR}} \end{pmatrix} \quad \text{--- } Q_n$$

Рис. 6. Выделение подматрицы Q_n из матрицы состояния блока нелинейности V

иметь вид:

$$U = \begin{pmatrix} z_{D_{FIR}} & \cdots & z_1 & z_0 & 0 & \cdots & 0 \\ z_{D_{FIR}+1} & \cdots & z_2 & z_1 & z_0 & \cdots & 0 \\ \vdots & & & \ddots & & & \vdots \\ 0 & \cdots & z_{N-1} & z_{N-2} & z_{N-3} & \cdots & z_{N-2-D_{FIR}} \\ 0 & \cdots & 0 & z_{N-1} & z_{N-2} & \cdots & z_{N-1-D_{FIR}} \end{pmatrix}, \quad (5.4.1)$$

где $D_{FIR} = \frac{M-1}{2}$. Выход модели Гаммерштейна определяется выражением (5.1.6).

Аналогично выход модели Гаммерштейна можно выразить через вектор коэффициентов h и модификацию матрицы состояния V модели нелинейности амплитудной характеристики усилителя мощности $g(\cdot)$. Матрица состояния модели нелинейности определяется выражениями (5.2.7), (??) для полиномиальной модели без памяти и кусочно-линейной модели соответственно.

Представим n -ый отсчет на выходе модели в виде матрично-векторного произведения. Для этого выделим из матрицы состояния V подматрицу Q_n размерности $M \times L$ (рис. 6), где L – длина вектора состояния v_n блока нелинейности $g(\cdot)$. Отсчет y_n на выходе модели можно представить в виде:

$$y_n = w^T Q_n h = h^T (Q_n^T w). \quad (5.4.2)$$

Ввиду ассоциативности векторно-матричного умножения из (5.4.2) следует, что сначала можно умножить вектор коэффициентов фильтра на Q_n , после чего на вектор h^T , но построчное умножение $Q_n^T w$ равносильно фильтрации столбцов матрицы V .

Таким образом, вектор отсчетов на выходе модели Гаммерштейна можно представить:

$$\mathbf{y} = \mathbf{V}_f \mathbf{h}, \quad (5.4.3)$$

где в \mathbf{V}_f каждый столбец – свёртка столбца матрицы \mathbf{V} с вектором коэффициентов фильтра.

5.5 Модель Винера нелинейных искажений в системах связи

5.6 Модель Винера-Гаммерштейна нелинейных искажений в системах связи

5.7 Нейросетевые структуры аппроксимации нелинейных искажений приемо-передающего тракта

Глава II

Исследование эффективности методов адаптивной компенсации нелинейных помех в приёмном тракте мобильного терминала

1. Структура тестовой платформы

В данном разделе описывается тестовая установка, на которой сформирован сигнал, имитирующий сигнал паразитной помехи в приёмном тракте мобильного устройства.

На рис. 7 изображена установка формирования сигнала, имитирующего паразитную помеху на передатчике вследствие прохождения сигнала передатчика через усилитель мощности.

Сигнал передатчика x записывается в генератор сигналов (рис. 7). Длительность сигнала 2457600 отсчётов, что во времени соответствует 20 мс. На вход радиочастотного усилителя мощности поступает сигнал передатчика в аналоговом виде на несущей частоте 1.7 ГГц.

Цифровой осциллограф запрограммирован на выделение второй гармоники сигнала на частоте 3.4 ГГц на выходе радиочастотного усилителя мощности. Частота дискретизации АЦП осциллографа $f_s = 122.88$ МГц выбрана в соответствии со стандартом 5G [3gpp_sample_rate]. Обработка сигнала помехи ведётся на компьютере в системе Python.

Выделение второй гармоники f из потока цифрового сигнала происходит путём подсчета взаимной корреляции сигнала f и квадрата сигнала передатчика x^2 , как изображено на рис. 8. Последующие 2457600 отсчётов сигнала f , начиная с отсчёта, при котором зафиксирован максимум взаимной корреляции, считаются копией сигнала второй гармоники.

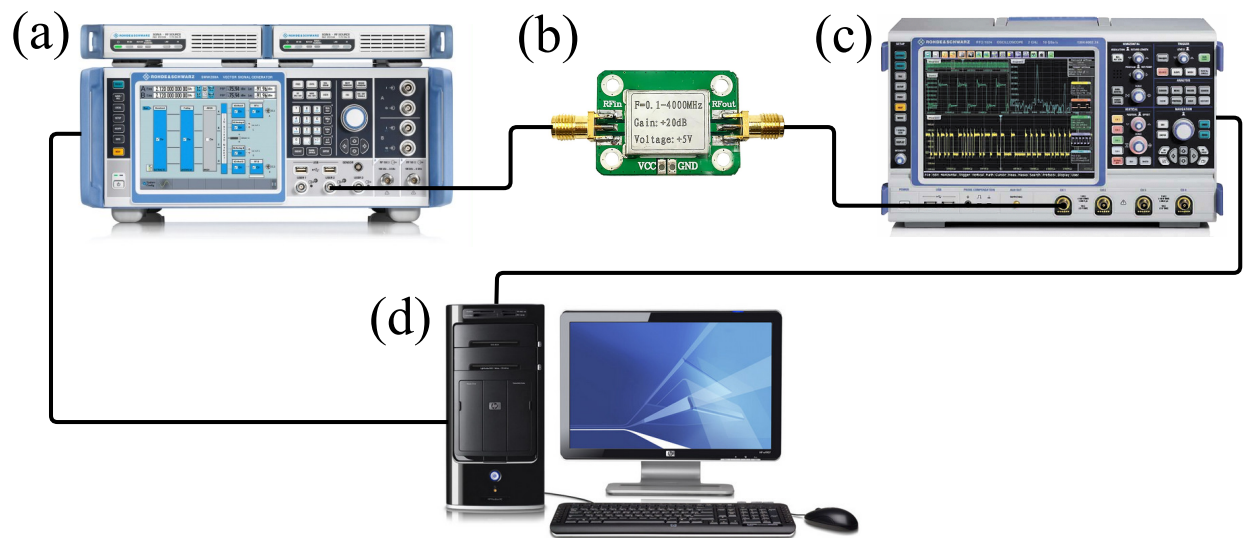


Рис. 7. Установка формирования второй гармоники сигнала передатчика: (а) Цифровой генератор сигналов; (б) Аналоговый усилитель мощности; (в) Цифровой осциллограф; (д) Компьютер.

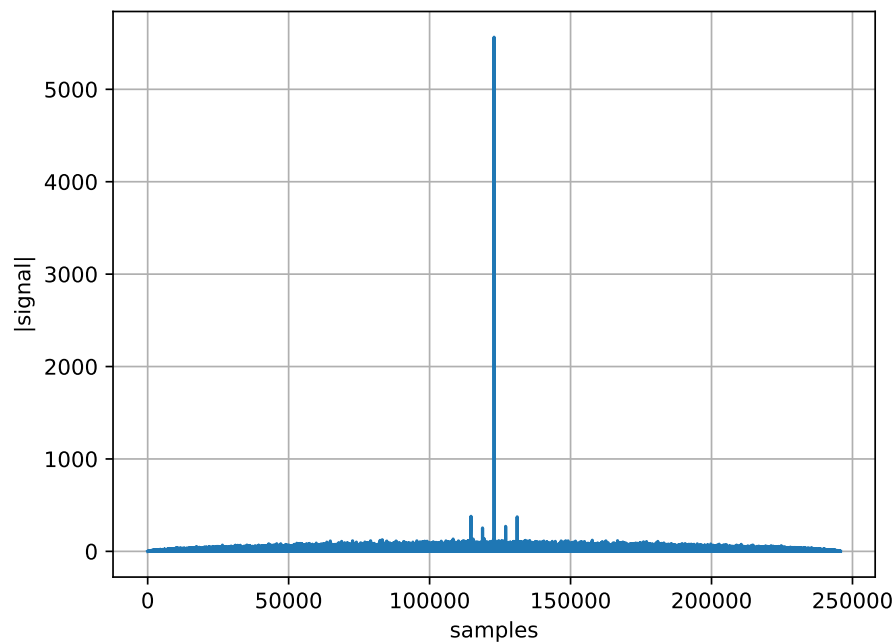


Рис. 8. Модуль функции взаимной корреляции сигнала второй гармоники f и квадрата сигнала передатчика x^2

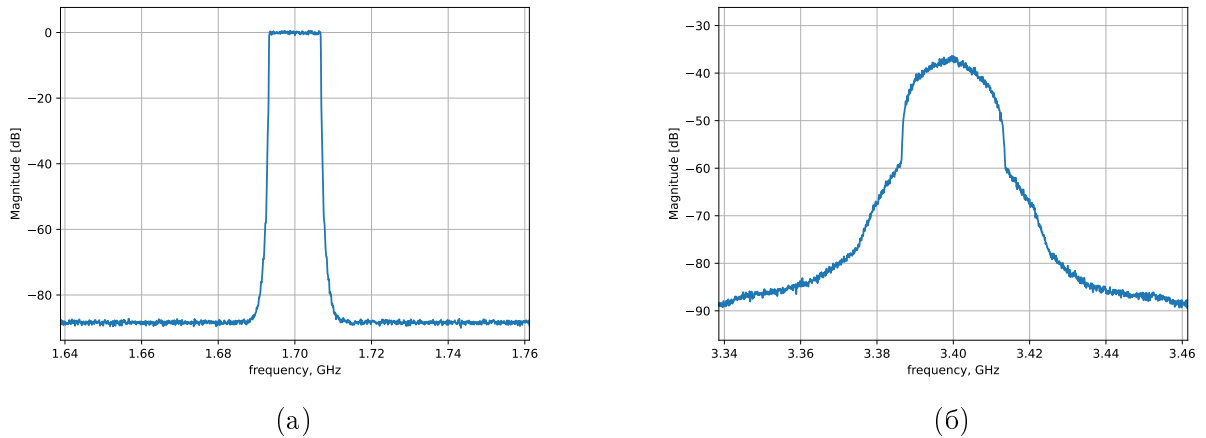


Рис. 9. Спектральная плотность мощности: (а) сигнал, формируемый на передатчике для отправки в канал связи x ; (б) вторая гармоника f , образованная после прохождения сигнала передатчика через усилитель мощности

В результате имеются два массива длиной по 2457600 отсчётов каждый: сигнал передатчика x и сигнал второй гармоники f . Спектральные плотности мощности этих сигналов изображены на рис. 9а, 9б.

Для симуляции прохождения второй гармоники по различным путям распространения от передатчика к приёмнику производится свёртка массива отсчётов f с коэффициентами w импульсной характеристики КИХ-фильтра:

$$d = f * w \quad (1.0.1)$$

Таким образом, получаем сигнал d , имитирующий сигнал паразитной второй гармоники сигнала передатчика на приёмнике мобильного терминала.

Импульсная характеристика w формируется в системе Python в соответствии с данными о реальном канале распространения помехи.

Частотная характеристика канала распространения помехи от передатчика к приёмнику изображена на рис. 10а, спектральная плотность мощности сигнала паразитной помехи d изображена на рис. 10б.

2. Формирование тестового набора данных

OFDM сигнал передатчика сформирован в соответствии со стандартом LTE. Согласно документации 3GPP [3gpp_36_211] сигнал делится на ресурсные блоки по ширине полосы и длительности сигнала (англ. Resource Blocks).

Каждый ресурсный блок содержит 12 поднесущих с сигналами с шириной полосы равной 15 кГц [3gpp_36_211]. Таким образом, каждый такой блок имеет ширину полосы равную 180 кГц.

В данной работе рассматриваются 5 различных сигналов передатчика. Каждый

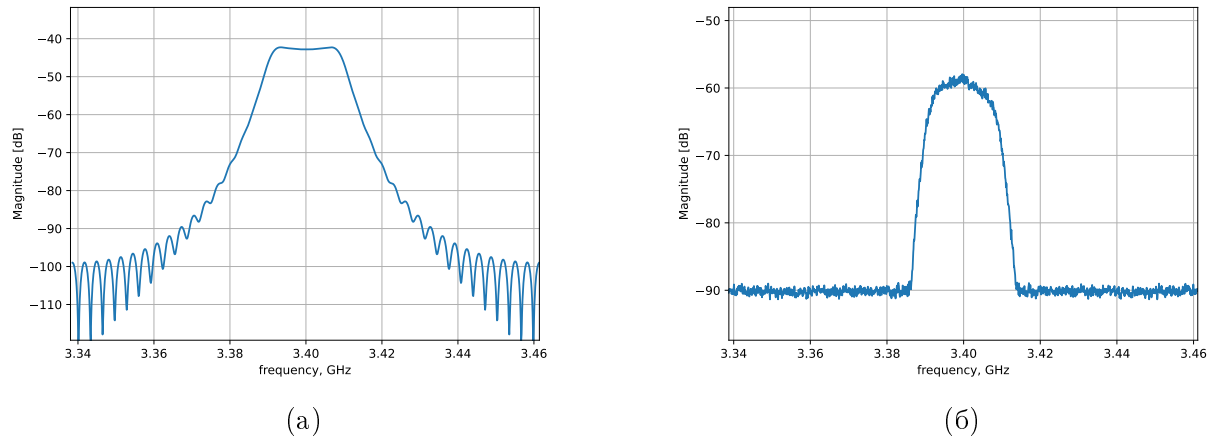


Рис. 10. (а) Амплитудно-частотная характеристика канала распространения второй гармоники от передатчика к приёмнику; (б) паразитная помеха на приёмнике d

сигнал содержит RB_{num} ресурсных блоков:

$$RB_{num} = \{1, 25, 50, 75, 100\}.$$

Кроме того, рассматриваются 3 различных импульсных характеристики, сгенерированные на языке программирования Python и соответствуют 3-м различным оценкам канала распространения паразитной помехи. Оценки канала были получены на тестовой платформе. Импульсные характеристики приведены в следующем разделе на рис. 12б, 14б, 16б. Обозначим их соответственно:

$$multipath = \{path_0, path_1, path_2\}.$$

Таким образом, набор данных, на которых будет проводиться исследование алгоритмов компенсации помехи, содержит 15 различных сигналов паразитной помехи. В таблице 1 отображены условия, в которых формируются сигналы, имитирующие сигналы паразитной помехи. Эти сигналы, как отмечалось ранее, формируются в зависимости от ширины полосы сигнала передатчика и от канала распространения помехи от передатчика к приёмнику.

Отметим также, что ввиду требований стандарта LTE [3gpp_36_211], выходная мощность усилителя возрастает в зависимости от ширины полосы сигнала передатчика (таблица 2).

Исследование эффективности блочных и стохастических алгоритмов будем проводить в условиях стационарной паразитной помехи.

Стационарность помехи означает, что условия формирования сигнала помехи (таблица 1) должны быть неизменными на протяжении адаптации. То есть неизменными должны быть канал распространения помехи, а также ширина полосы сигнала передатчика.

Запустим блочные и стохастические алгоритмы на тренировочных блоках дан-

	RB_{num}				
<i>multipath</i>	RB1; $path_0$	RB25; $path_0$	RB50; $path_0$	RB75; $path_0$	RB100; $path_0$
	RB1; $path_1$	RB25; $path_1$	RB50; $path_1$	RB75; $path_1$	RB100; $path_1$
	RB1; $path_2$	RB25; $path_2$	RB50; $path_2$	RB75; $path_2$	RB100; $path_2$

Таблица 1. Условия формирования сигналов, имитирующих сигнал паразитной помехи на приёмнике мобильного устройства

PA power, dBm	6.7	11.7	17.2	22.8	26.8
RB_{num}	RB1	RB25	RB50	RB75	RB100

Таблица 2. Соответствие между выходной мощностью аналогового усилителя и количеством ресурсных блоков сигнала передатчика

ных, состоящих из 2457600 отсчётов сигналов, условия формирования которых отображены в таблице 1. Затем сравним критерии адаптации, достигаемые методами первого порядка за это время с результатами, которые получены методами второго порядка.

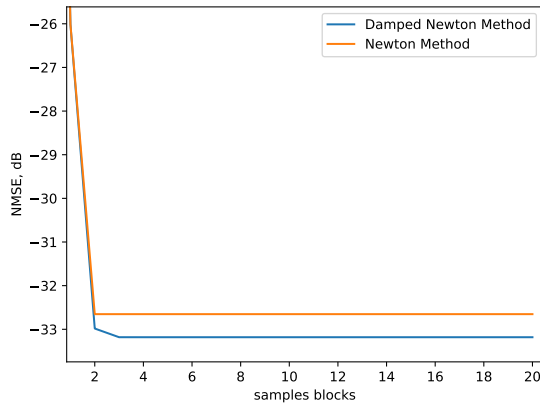
3. Исследование алгоритмов компенсации стационарной паразитной помехи на приёмнике

3.1 Аппроксимация стационарных нелинейных искажений на приемнике устройства связи на основе классических и нейросетевых структур

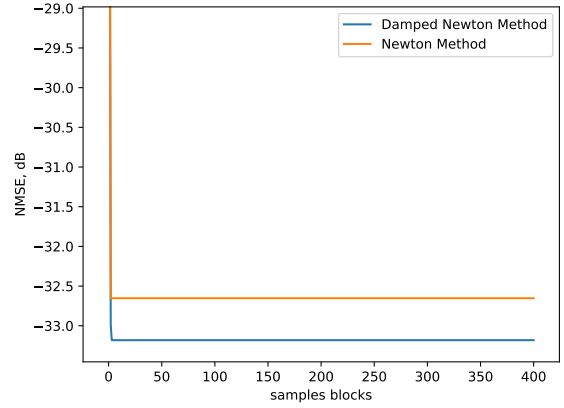
3.2 Исследование опорных уровней компенсации стационарной помехи методами второго порядка

Сравнение уровня компенсации паразитной помехи различными алгоритмами будем проводить относительно критерия нормированного среднего квадрата ошибки (1.5.2), поскольку NMSE позволяет оценить уровень отклонения выхода модели от помехи на приёмнике независимо от динамического диапазона помехи.

Методы второго порядка обеспечивают быстрое приближение к точке оптимума по сравнению с методами первого порядка. В связи с этим целесообразно использо-



(а) Сигналы помехи и на передатчике состоят из 20 блоков длиной 122880 отсчётов



(б) Сигналы помехи и на передатчике состоят из 400 блоков длиной 122880 отсчётов

Рис. 11. Кривые адаптации модели Гаммерштейна методом Ньютона и демпфированным методом Ньютона. Случай $\{RB25, path_0\}$

вать значения NMSE, полученные в результате работы методов второго порядка, в качестве опорных уровней подавления помехи для дальнейшего сравнения с ними значений NMSE, полученных градиентными методами первого порядка.

Следует отметить, что метод Ньютона может стогнироваться при определенных условиях. Это может произойти, например, в случае если коэффициенты модели примут значения, соответствующие точке в окрестности седловой точки целевой функции. В этом случае глобальный минимум не может быть достигнут в результате работы метода Ньютона.

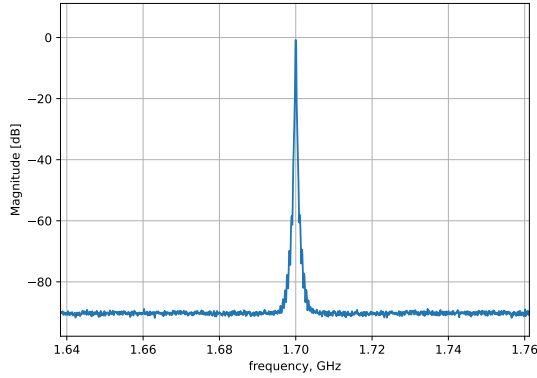
На рис. 11 изображены кривые адаптации метода Ньютона и демпфированного метода Ньютона в случае, когда сигнал передатчика имеет ширину полосы равную 4.5 МГц (RB25), а канал распространения помехи $path_0$. При этом сигналы передатчика и помехи состоят из 20 блоков (рис. 11а) и 400 блоков (рис. 11б) длиной 122880 отсчётов.

Таким образом, предпочтительней использовать значения критерия NMSE, полученные в результате работы демпфированного метода Ньютона в качестве опорных значений.

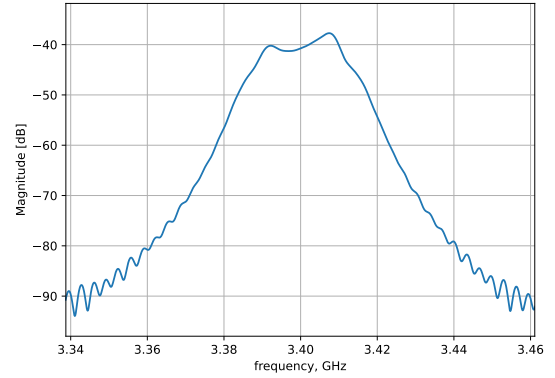
В дальнейшем при исследовании работы стохастических алгоритмов будем сравнивать значения критерия NMSE, получаемые в результате работы алгоритмов SGD-SGD, SGD-DCD со значениями NMSE, получаемыми в результате работы демпфированного метода Ньютона.

Вычислим опорные значения критерия NMSE, полученные при помощи демпфированного метода Ньютона для каждого случая, отображённого в таблице 1.

Рассмотрим подавление паразитной помехи в случае $\{RB1, path_0\}$ (рис. 12), когда ширина полосы сигнала передатчика определяется одним ресурсным блоком, 180 кГц (рис. 12а), а канал распространения определяется частотной характеристикой $path_0$ (рис. 12б).



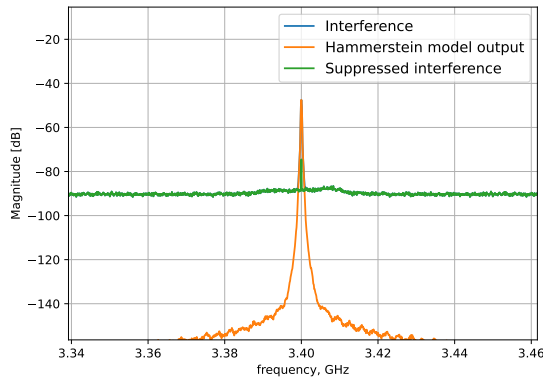
(а) Спектральная плотность мощности сигнала передатчика с одним ресурсным блоком



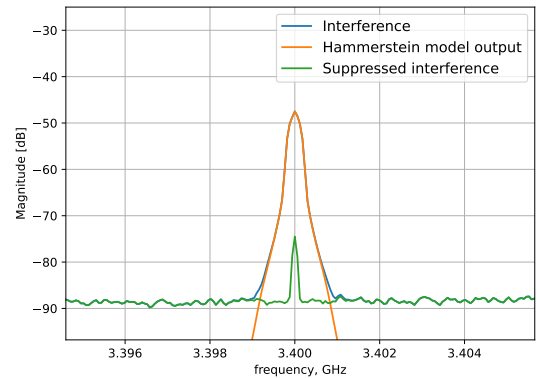
(б) Амплитудно-частотная характеристика канала распространения помехи $path_0$

Рис. 12. Условия формирования сигнала помехи на приёмнике $\{RB1, path_0\}$

На рис. 13 изображены спектральные плотности мощности паразитной помехи, сигнала на выходе модели Гаммерштейна после адаптации демпфированным методом Ньютона, а также отклонения выхода модели Гаммерштейна от сигнала помехи. Значение критерия NMSE в данном случае составляет -23.8 dB.



(а)



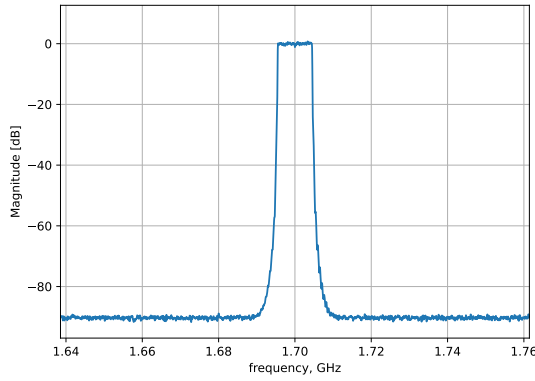
(б)

Рис. 13. Спектральная плотность мощности сигнала паразитной помехи после адаптации методом Ньютона. Случай $\{RB50, path_1\}$

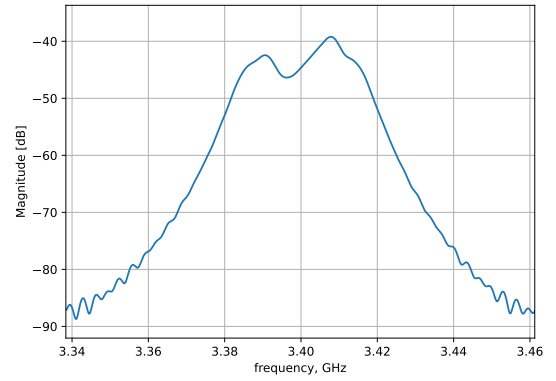
Рассмотрим подавление паразитной помехи в случае $\{RB50, path_1\}$ (рис. 14), когда ширина полосы сигнала передатчика определяется 50-ю ресурсными блоками, 9 МГц (рис. 14а), а канал распространения определяется частотной характеристикой $path_1$ (рис. 14б).

На рис. 15 изображены спектральные плотности мощности паразитной помехи, сигнала на выходе модели Гаммерштейна после адаптации демпфированным методом Ньютона, а также отклонения выхода модели Гаммерштейна от сигнала помехи. Значение критерия NMSE в данном случае составляет -36.2 dB.

Рассмотрим подавление паразитной помехи в случае $\{RB100, path_2\}$ (рис. 16), когда ширина полосы сигнала передатчика определяется 100 ресурсными блоками,

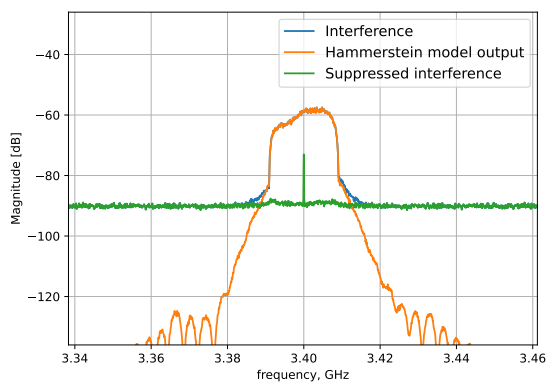


(а) Спектральная плотность мощности сигнала передатчика с 50-ю ресурсными блоками

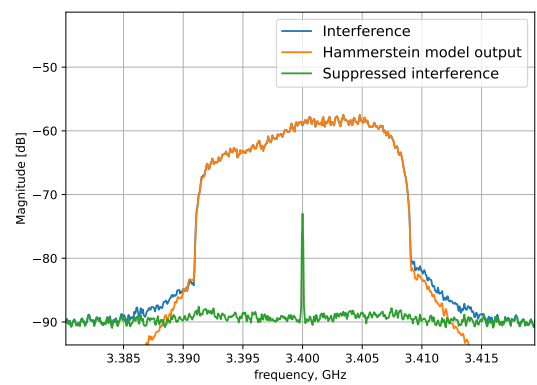


(б) Амплитудно-частотная характеристика канала распространения помехи $path_1$

Рис. 14. Условия формирования сигнала помехи на приёмнике $\{RB50, path_1\}$

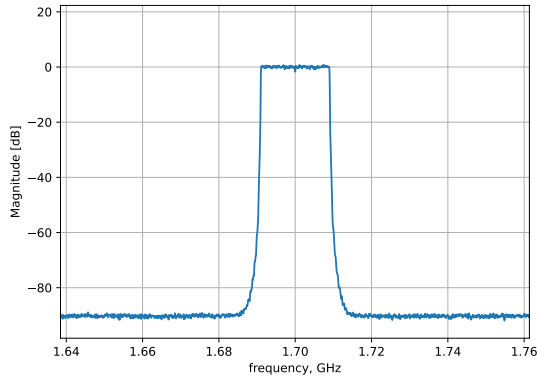


(а)

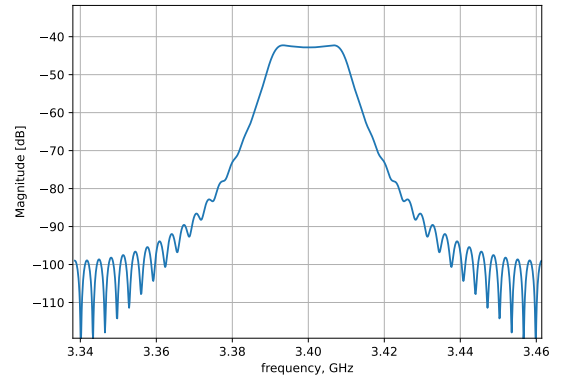


(б)

Рис. 15. Спектральная плотность мощности сигнала паразитной помехи после адаптации методом Ньютона. Случай $\{RB50, path_1\}$

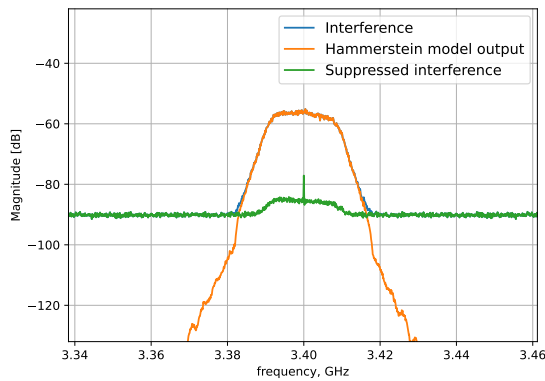


(а) Спектральная плотность мощности сигнала передатчика с 100 ресурсными блоками

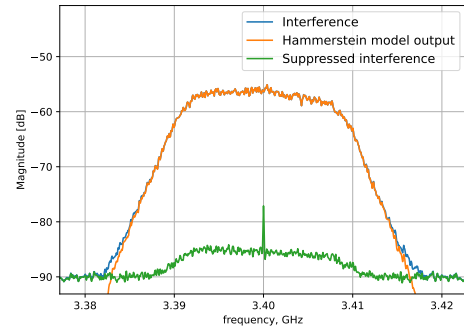


(б) Амплитудно-частотная характеристика канала распространения помехи $path_2$

Рис. 16. Условия формирования сигнала помехи на приёмнике $\{RB100, path_2\}$



(а)



(б)

Рис. 17. Спектральная плотность мощности сигнала паразитной помехи после адаптации методом Ньютона. Случай $\{RB100, path_2\}$

18 МГц (рис. 16а), а канал распространения определяется частотной характеристикой $path_2$ (рис. 16б).

На рис. 17 изображены спектральные плотности мощности паразитной помехи, сигнала на выходе модели Гаммерштейна после адаптации демпфированным методом Ньютона, а также отклонения выхода модели Гаммерштейна от сигнала помехи. Значение критерия NMSE в данном случае составляет -34.3 dB.

Спектральные плотности мощности помехи до и после подавления паразитной помехи демпфированным методом Ньютона для других случаев, приведённых в таблице 1, представлены в приложении А1.

Таблица 3 отражает опорные значения критерия NMSE dB, получаемые в результате адаптации модели Гаммерштейна демпфированным методом Ньютона для всех случаев формирования паразитной помехи, представленных в таблице 1.

Таким образом, в результате работы демпфированного метода Ньютона подавление паразитной помехи для сигнала шириной полосы 180 кГц не хуже 22 дБ для

	RB1	RB25	RB50	RB75	RB100
$path_0$	-24.1	-33.2	-31.7	-31.8	-29.7
$path_1$	-22.2	-31.7	-31.6	-31.3	-29.2
$path_2$	-25.4	-34.1	-31.8	-32.0	-30.0

Таблица 3. Опорные значения критерия NMSE dB, полученные в результате адаптации демпфированным методом Ньютона для каждого случая формирования паразитной помехи

трех исследованных характеристик линейного канала распространения.

Подавление помехи в случае сигналов передатчика с шириной полосы 4.5 МГц, 9 МГц и 13.5 МГц не хуже 31 дБ для трёх рассмотренных оценок канала распространения помехи.

В случае сигнала передатчика с шириной полосы 18 МГц подавление помехи демпфированным методом Ньютона не хуже 29 дБ также для всех рассмотренных оценок канала распространения.

4. Исследование моделей цифрового предискажения сигнала передающего устройства в условиях динамического изменения выходной мощности нелинейного усилителя

- 4.1 Аппроксимация динамической нелинейности на основе классических и нейросетевых структур
- 4.2 Исследование опорных уровней компенсации динамических нелинейных искажений

Глава III

Практическая реализация нелинейных корректоров в системах связи

1. Исследование компенсации нелинейных искажений методами не требующими явного вычисления матрицы Гессе

Проведём сравнение кривых адаптации, а также значений критерия NMSE, полученных в результате работы блочной реализации градиентного спуска и стохастических методов SGD-SGD, SGD-DCD.

Метод SGD-SGD подразумевает адаптацию обоих слоёв модели Гаммерштейна методом стохастического градиентного спуска. Метод SGD-DCD подразумевает адаптацию блока нелинейности амплитудной характеристики усилителя мощности методом стохастического градиентного спуска и адаптацию КИХ-фильтра стохастической реализацией метода DCD.

Будем строить кривые адаптации по значениям критерия NMSE (1.5.2), который подсчитывается после адаптации на блоке сигнала длиной 122880 отсчётов.

На рис. 18 приведены кривые адаптации данных алгоритмов для случая, представленного на рис. 12, при котором ширина полосы сигнала передатчика определяется одним ресурсным блоком, 180 кГц (рис. 12а), а канал распространения определяется частотной характеристикой $path_0$ (рис. 12б).

На рис. 18 по оси абсцисс отмечены порядковые номера блоков длиной 122880 отсчётов.

Помимо кривых адаптации градиентных методов первого порядка на рис. 18 приведена кривая адаптации демпфированного метода Ньютона, отражающая опорные значения критерия NMSE.

Рис. 18 отражает тот факт, что наибольшей скоростью сходимости среди градиентных методов обладает алгоритм SGD-DCD для данных условий формирования

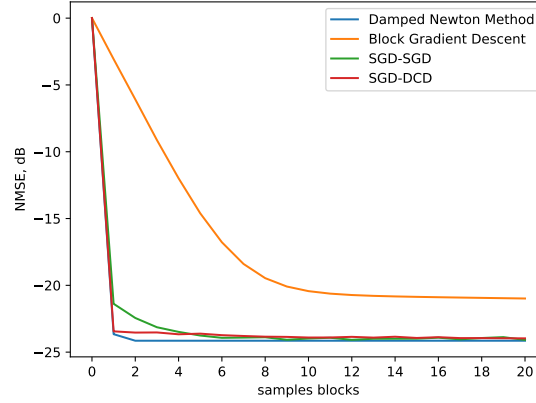


Рис. 18. Кривые адаптации модели Гаммерштейна для случая сигнала передатчика с шириной полосы равной 180 кГц и канала распространения помехи $path_0$

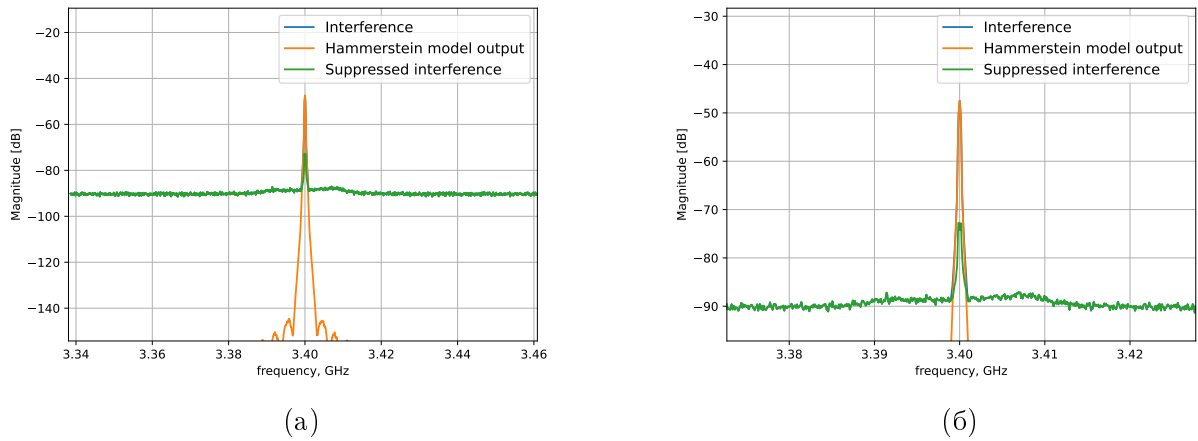


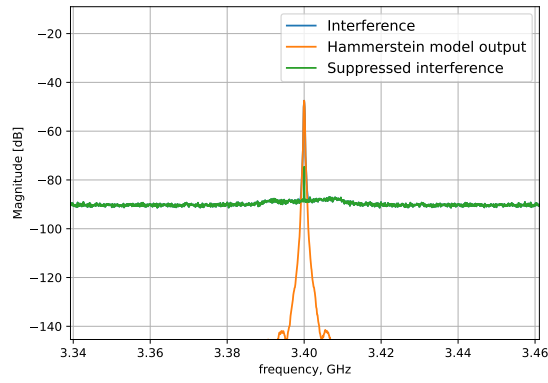
Рис. 19. Спектральная плотность мощности сигнала паразитной помехи после адаптации блочным методом градиентного спуска. Случай $\{RB1, path_0\}$

помехи. Кроме того, данный метод обеспечивает наилучшее подавление близкое к значениям NMSE, полученным в результате работы демпфированного метода Ньютона.

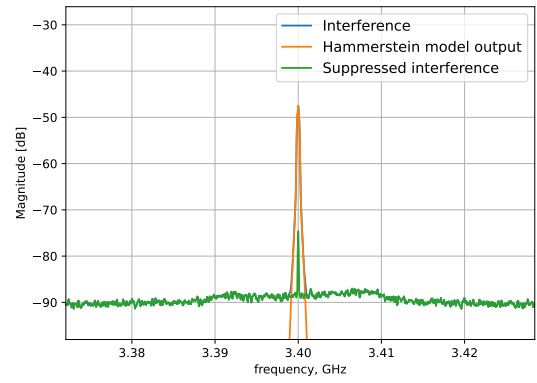
На рис. 19–21 изображены спектральные плотности мощности паразитной помехи, сигнала на выходе модели Гаммерштейна после адаптации, а также отклонения выхода модели Гаммерштейна от сигнала помехи. Значения критериев NMSE в данном случае составляют -21.0 dB, -24.1 dB и -24.0 dB для блочного метода градиентного спуска, SGD-SGD и SGD-DCD соответственно.

Кривые адаптации данных алгоритмов для случая $\{RB50, path_1\}$ (рис. 14) приведены на рис. 22. Ширина полосы сигнала передатчика определяется 50-ю ресурсными блоками, 9 МГц (рис. 14а), а канал распространения определяется частотной характеристикой $path_1$ (рис. 14б).

Рис. 22 отражает тот факт, что наибольшей скоростью сходимости среди градиентных методов обладает алгоритм SGD-SGD для данных условий формирования

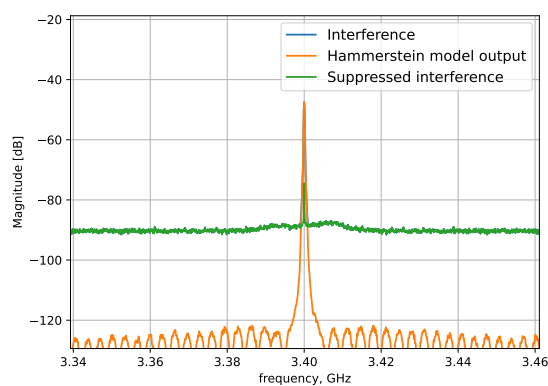


(a)

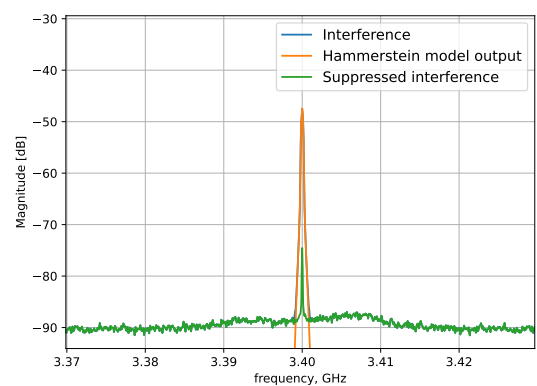


(б)

Рис. 20. Спектральная плотность мощности сигнала паразитной помехи после адаптации методом SGD-SGD. Случай $\{RB1, path_0\}$



(a)



(б)

Рис. 21. Спектральная плотность мощности сигнала паразитной помехи после адаптации методом SGD-DCD. Случай $\{RB1, path_0\}$

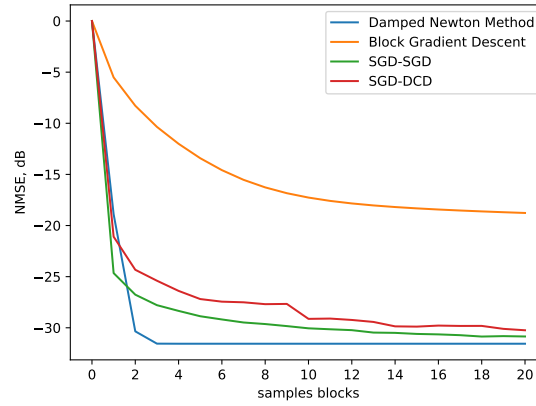
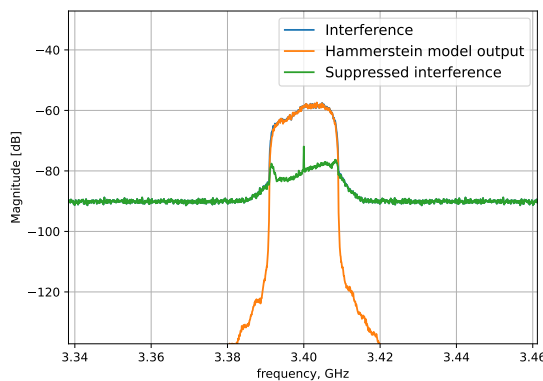


Рис. 22. Кривые адаптации модели Гаммерштейна для случая сигнала передатчика с шириной полосы равной 9 МГц и канала распространения помехи $path_1$

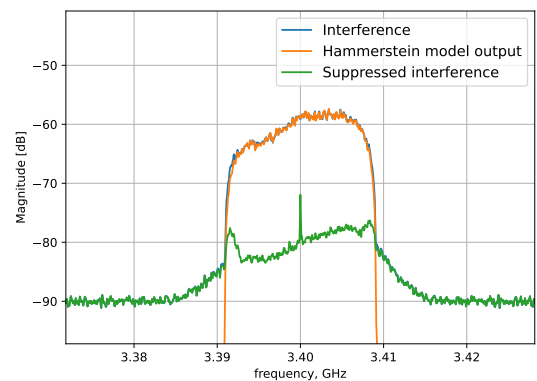
помехи. Кроме того, данный метод обеспечивает наилучшее подавление близкое к значениям NMSE, полученным в результате работы демпфированного метода Ньютона. При этом алгоритм SGD-SGD сходится к опорным значениям NMSE более плавно по сравнению с методом SGD-DCD.

Отметим также, что блочная реализация градиентного спуска адаптирует модель Гаммерштейна существенно медленнее, чем алгоритмы SGD-SGD и SGD-DCD.

На рис. 23–25 изображены спектральные плотности мощности паразитной помехи, сигнала на выходе модели Гаммерштейна после адаптации, а также отклонения выхода модели Гаммерштейна от сигнала помехи. Значения критериев NMSE в данном случае составляют -18.8 dB, -30.9 dB и -30.2 dB для блочного метода градиентного спуска, SGD-SGD и SGD-DCD соответственно.



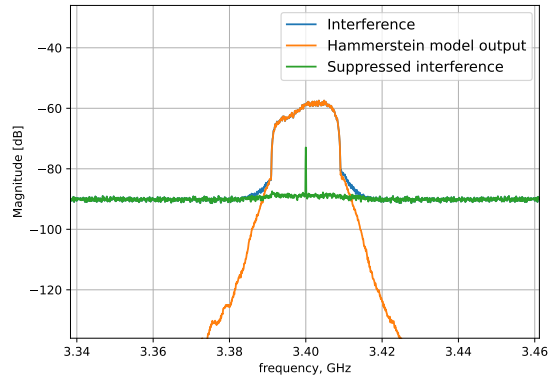
(а)



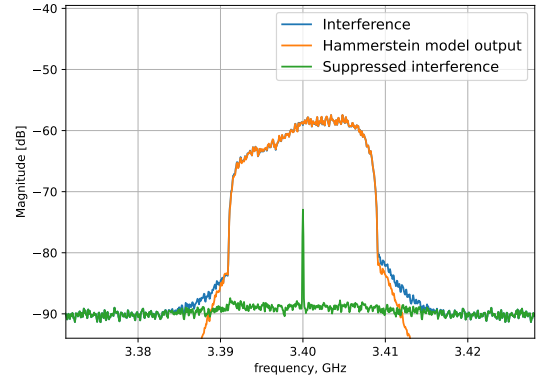
(б)

Рис. 23. Спектральная плотность мощности сигнала паразитной помехи после адаптации блочным методом градиентного спуска. Случай $\{RB50, path_1\}$

Кривые адаптации данных алгоритмов для случая $\{RB100, path_2\}$ (рис. 16) приведены на рис. 26. Ширина полосы сигнала передатчика определяется 100 ресурсными блоками, 18 МГц (рис. 16а), а канал распространения определяется частотной

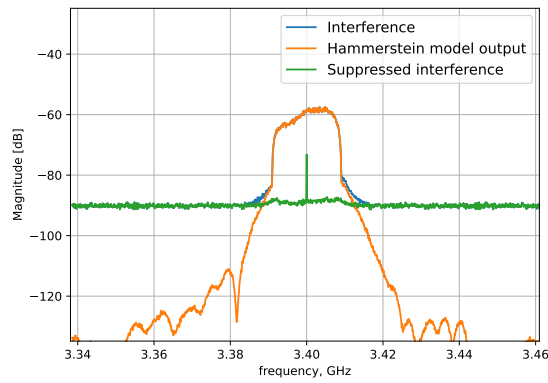


(a)

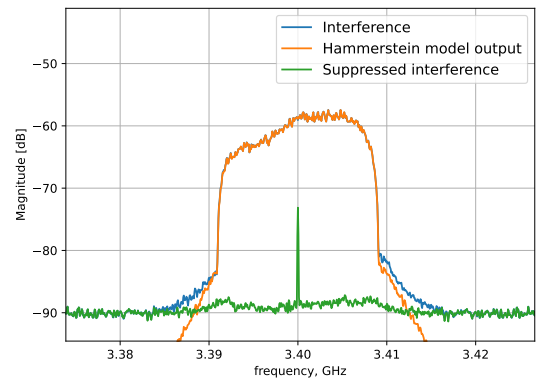


(б)

Рис. 24. Спектральная плотность мощности сигнала паразитной помехи после адаптации методом SGD-SGD. Случай $\{RB50, path_1\}$



(a)



(б)

Рис. 25. Спектральная плотность мощности сигнала паразитной помехи после адаптации методом SGD-DCD. Случай $\{RB50, path_1\}$

характеристикой $path_2$ (рис. 16б).

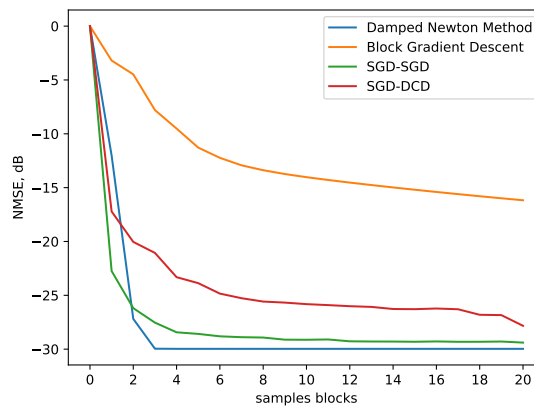


Рис. 26. Кривые адаптации модели Гаммерштейна для случая сигнала передатчика с шириной полосы равной 18 МГц и канала распространения помехи $path_2$

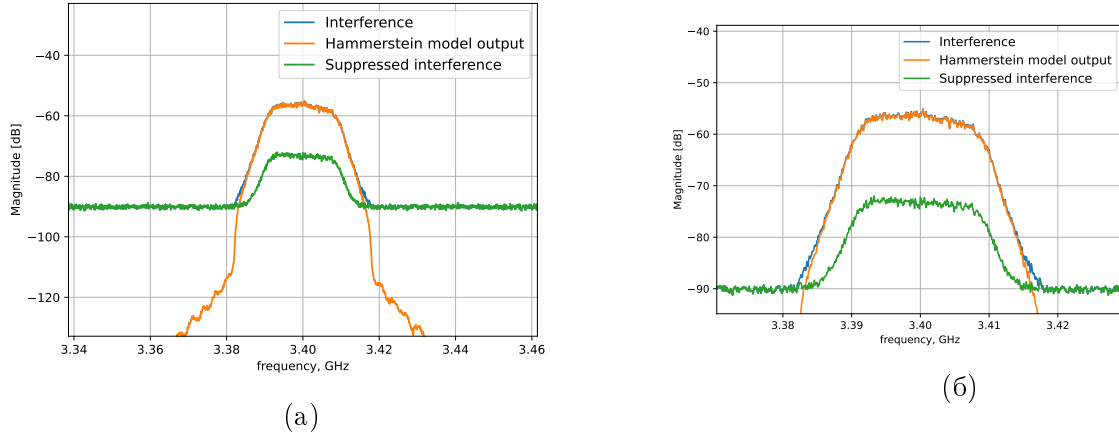


Рис. 27. Спектральная плотность мощности сигнала паразитной помехи после адаптации блочным методом градиентного спуска. Случай $\{RB100, path_2\}$

Рис. 26 отражает тот факт, что наибольшей скоростью сходимости среди градиентных методов обладает алгоритм SGD-SGD для данных условий формирования помехи. Кроме того, данный метод обеспечивает наилучшее подавление близкое к значениям NMSE, полученным в результате работы демпфированного метода Ньютона.

На рис. 27–29 изображены спектральные плотности мощности паразитной помехи, сигнала на выходе модели Гаммерштейна после адаптации, а также отклонения выхода модели Гаммерштейна от сигнала помехи. Значения критериев NMSE в данном случае составляют -16.2 dB, -29.4 dB и -27.8 dB для блочного метода градиентного спуска, SGD-SGD и SGD-DCD соответственно.

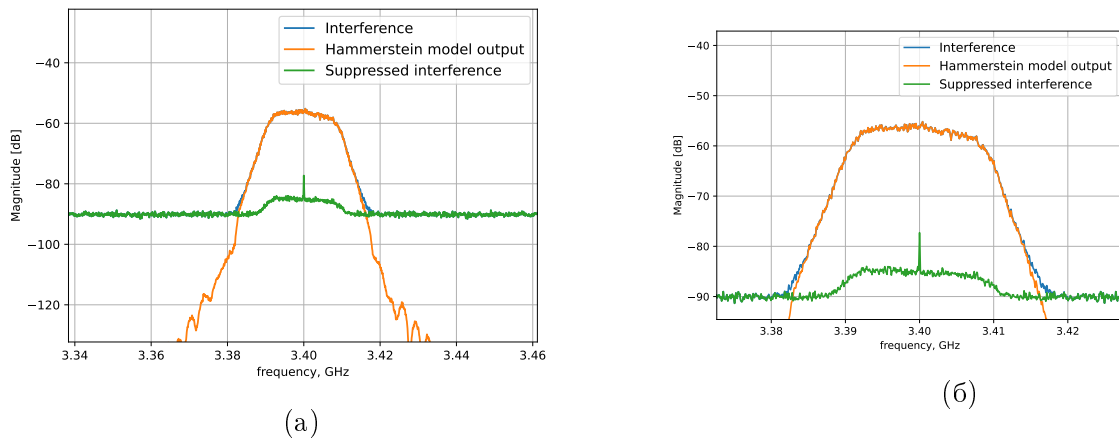


Рис. 28. Спектральная плотность мощности сигнала паразитной помехи после адаптации методом SGD-SGD. Случай $\{RB100, path_2\}$

Кривые адаптации блочного метода градиентного спуска, SGD-SGD, SGD-DCD, а также спектральные плотности мощности помехи после адаптации данными методами для всех случаев формирования помехи из таблицы 1 представлены в приложении А2.

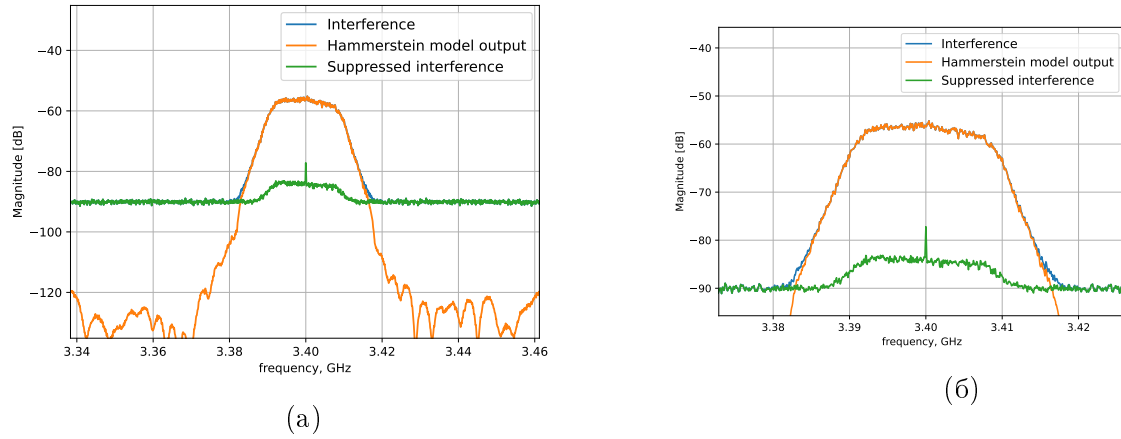


Рис. 29. Спектральная плотность мощности сигнала паразитной помехи после адаптации методом SGD-DCD. Случай $\{RB100, path_2\}$

Таблица 4 отражает значения критерия NMSE dB, полученные в результате адаптации модели Гаммерштейна блочным методом градиентного спуска на сигналах стационарной помехи длительностью 2457600 отсчётов для каждого случая формирования паразитной помехи (таблица. 1).

	RB1	RB25	RB50	RB75	RB100
$path_0$	-21.0	-22.8	-19.9	-16.4	-18.8
$path_1$	-20.0	-21.2	-18.8	-15.0	-16.3
$path_2$	-21.6	-23.9	-20.0	-15.3	-16.2

Таблица 4. Значения критерия NMSE dB, полученные в результате адаптации блочным методом градиентного спуска для каждого случая формирования паразитной помехи

Таблица 5 отражает значения критерия NMSE dB, полученные в результате адаптации модели Гаммерштейна методом SGD-SGD на сигналах стационарной помехи длительностью 2457600 отсчётов для каждого случая формирования паразитной помехи (таблица. 1).

Таблица 6 отражает значения критерия NMSE dB, полученные в результате адаптации модели Гаммерштейна методом SGD-DCD на сигналах стационарной помехи длительностью 2457600 отсчётов для каждого случая формирования паразитной помехи (таблица. 1).

Из таблицы 4 следует, что за время работы алгоритмов адаптации равное 2457600 отсчётов блочная реализация метода градиентного спуска обеспечивает подавление паразитной помехи не меньше, чем на 20 дБ для случаев сигнала передатчика с

	RB1	RB25	RB50	RB75	RB100
$path_0$	-24.1	-32.3	-31.4	-31.4	-29.1
$path_1$	-22.2	-30.5	-30.9	-30.4	-28.6
$path_2$	-25.3	-32.7	-31.5	-31.2	-29.4

Таблица 5. Значения критерия NMSE dB, полученные в результате адаптации методом SGD-SGD для каждого случая формирования паразитной помехи

	RB1	RB25	RB50	RB75	RB100
$path_0$	-24.0	-32.5	-31.2	-27.7	-27.5
$path_1$	-22.0	-30.5	-30.2	-26.1	-26.8
$path_2$	-25.0	-32.8	-30.6	-27.1	-27.8

Таблица 6. Значения критерия NMSE dB, полученные в результате адаптации методом SGD-DCD для каждого случая формирования паразитной помехи

шириной полосы равной 180 кГц и 4.5 МГц для рассмотренных случаев оценки канала распространения помехи.

Для случая сигнала передатчика с шириной полосы 9 МГц подавление паразитной нелинейной помехи составляет как минимум 18 дБ.

При этом для случая сигнала передатчика с шириной полосы 13.5 МГц и 28 МГц подавление помехи не хуже 15 дБ.

Из таблицы 6 следует, что за 2457600 отсчётов стохастический алгоритм SGD-DCD обеспечивает компенсацию помехи для случая сигнала передатчика с шириной полосы 180 кГц как минимум на 21 дБ для всех рассмотренных каналов распространения, что на 1 дБ больше, чем в случае блочного градиентного спуска.

Для случая сигналов передатчика с шириной полосы 4.5 МГц и 9 МГц компенсация помехи не хуже 30 дБ, что больше на 9 дБ и 12 дБ соответственно, чем в случае блочного градиентного спуска.

Компенсация паразитной помехи в случае сигнала передатчика шириной полосы 13.5 и 18 МГц не меньше, чем 26 дБ, что больше на 11 дБ и 10 дБ соответственно, чем в случае блочной реализации градиентного спуска.

Из таблицы 5 следует, что за 2457600 отсчётов стохастический алгоритм SGD-SGD компенсирует паразитную помеху не хуже, чем на 22 дБ для всех оценок канала

распространения помехи, что на 1 дБ лучше, чем в случае алгоритма SGD-DCD. При этом из графиков кривых адаптаций на рис. 18, ??, ?? следует, что скорость сходимости алгоритма SGD-SGD для данного случая выше, чем для SGD-DCD и блочного градиентного спуска.

Подавление нелинейной помехи в случае сигнала передатчика с шириной полосы 4.5 МГц, 9 МГц не меньше, чем 30 дБ, что совпадает с результатом работы алгоритма SGD-DCD за 2457600 отсчётов для всех рассмотренных оценок канала распространения помехи. При этом из графиков кривых адаптаций на рис. ??, ??, ?? для сигнала передатчика RB25 и на рис. ??, 22, ?? для сигнала передатчика RB50 следует, что скорость сходимости алгоритма SGD-SGD для данных случаев выше, чем скорость сходимости SGD-DCD и блочной реализации метода градиентного спуска.

Компенсация нелинейной помехи в случае сигнала передатчика с шириной полосы 13.5 МГц не меньше, чем 30 дБ, что больше, чем в случае алгоритма SGD-DCD на 4 дБ. Кроме того из графиков кривых адаптаций на рис. ??, ??, ?? следует, что скорость сходимости алгоритма SGD-SGD выше, чем скорость сходимости SGD-DCD и блочного метода градиентного спуска.

Для случая сигнала передатчика с шириной полосы 18 МГц подавление паразитной помехи не хуже 28 дБ, что на 1.5 дБ больше, чем в случае алгоритма SGD-DCD. Графики кривых адаптации на рис. ??, ??, 26 отражают тот факт, что скорость сходимости SGD-SGD выше, чем в случае SGD-DCD и блочной реализации градиентного спуска.

Таким образом подавление паразитной помехи методами SGD-SGD, SGD-DCD и блочной реализацией градиентного спуска на сигнале длительностью 2457600 отсчётов отличается от опорных значений компенсации, сформированных демпфированным методом Ньютона меньше, чем на 1 дБ, 4 дБ и 15 дБ соответственно, для рассмотренных случаев формирования помехи, представленных в таблице 1.

Отметим также, что за время работы алгоритмов адаптации равное 2457600 отсчётов среди рассмотренных градиентных методов первого порядка наилучшую компенсацию паразитной помехи, а также наибольшую скорость сходимости обеспечивает стохастический метод SGD-SGD для рассмотренных случаев формирования паразитной нелинейной помехи.

2. Сокращение ресурсов при реализации моделей нелинейных искажений засчет канонического тензорного разложения
3. Сокращение ресурсов при реализации моделей нелинейных искажений засчет канонического тензорного разложения
4. Сокращение ресурсов при реализации моделей нелинейных искажений путем оптимизации гиперпараметров модели
- 4.1 Метод Orthogonal Least Squares
- 4.2 Метод Proximal Policy Optimization
5. Методы экономии ресурсов при реализации адаптивных фильтров

В предыдущих разделах была рассмотрена компенсация паразитной помехи в приёмнике мобильного устройства путём адаптации модели Гаммерштейна в реальном времени. Стохастические алгоритмы позволяют потактово обновлять коэффициенты модели.

Предположим, что импульсный отклик канала распространения паразитной помехи от передатчика к приёмнику меняется незначительно в процессе работы мобильного устройства. В таком случае коэффициенты КИХ-фильтра можно зафиксировать после адаптации, а в реальном времени будут адаптироваться только коэффициенты слоя нелинейности амплитудной характеристики усилителя.

Поскольку в таком случае коэффициенты фильтра могут быть зафиксированы после адаптации, то есть возможность организовать экономию таких ресурсов, как мощность потребляемая цифровой схемой, а также площадь, занимаемая фильтром на кристалле RF-чипсета.

В экспериментах, приведённых в предыдущих разделах, использовались длинные КИХ-фильтры порядка $M = 38$ и выше, что соответствует аппаратной реализации на кристалле RF-чипсета не менее 39 умножителей.

Реализация умножителей является наиболее трудной с точки зрения используемых ресурсов по сравнению с другими элементами прямой формы реализации КИХ-

фильтра, то есть сумматорами и элементами задержки. Это связано с тем, что мощность, потребляемая умножителем, а также площадь, занимаемая на кристалле, растут квадратично в зависимости от разрядности умножителя, в то время как потребление этих же ресурсов сумматорами и элементами задержки растёт линейно в зависимости от разрядности.

В связи с этим предлагается использовать метод квантования коэффициентов КИХ-фильтра, позволяющий сократить количество используемых умножителей за счёт выражения одних коэффициентов фильтра через произведение других коэффициентов и степеней двойки.

С точки зрения аппаратной реализации умножители заменяются на сумматоры и битовые сдвиги. Приведём пример из статьи [fir_quant]. Пусть имеется КИХ-фильтр, реализованный в прямой форме, со следующими коэффициентами:

$$\mathbf{h} = (h_0 \ h_1 \ h_2 \ h_3 \ h_4)^T = (1 \ 2 \ 4 \ 3 \ 6)^T. \quad (5.1)$$

В данном примере коэффициенты h_0, h_1, h_3 точно выражаются через коэффициенты h_2, h_4 , однако в общем случае такой возможности нет. Алгоритм, приведённый в статье [fir_quant] формирует импульсную характеристику, соответствующую уменьшенному числу умножителей таким образом, чтобы минимизировать отклонение исходных коэффициентов от тех, что формируется алгоритмом. При этом желаемое количество умножителей задаётся пользователем.

Приведём примеры работы алгоритма для сокращения числа умножителей линейно-фазового Фильтра Нижних Частот, линейно-фазового Полосового Фильтра, а также минимально фазового Фильтра Нижних Частот.

Рассмотрим линейно-фазовый Фильтр Нижних Частот порядка $N = 138$. Пусть коэффициенты его импульсной характеристики $\mathbf{h} = \{h_n\}, n \in \overline{0, N-1}$. Пусть при этом желаемое количество умножителей $M = 25$. Импульсную характеристику сформированного алгоритмом фильтра обозначим $\mathbf{g} = \{g_n\}, n \in \overline{0, N-1}$.

На рис. 30 изображена импульсная характеристика исходного фильтра нижних частот h_n , а также отклонение исходных коэффициентов от коэффициентов фильтра, упрощенного с точки зрения аппаратной реализации $h_n - g_n$. Это отклонение мало и составляет порядка 10^{-3} .

На рис. 31 изображена частотная характеристика исходного фильтра $H(\omega)$, фильтра, построенного на 25 умножителях $G(\omega)$, а также разность частотных характеристик $E(\omega) = H(\omega) - G(\omega)$.

Из рис. 31 видно, что величина отклонения частотной характеристики фильтра, построенного на 25 умножителях от частотной характеристики исходного фильтра порядка $N = 138$, реализованного в прямой форме, составляет -60 dB.

Рассмотрим линейно-фазовый Полосовой Фильтр порядка $N = 138$. Пусть коэффициенты его импульсной характеристики $\mathbf{h} = \{h_n\}, n \in \overline{0, N-1}$. Пусть при этом желаемое количество умножителей $M = 25$. Импульсную характеристику сформированного алгоритмом фильтра обозначим $\mathbf{g} = \{g_n\}, n \in \overline{0, N-1}$.

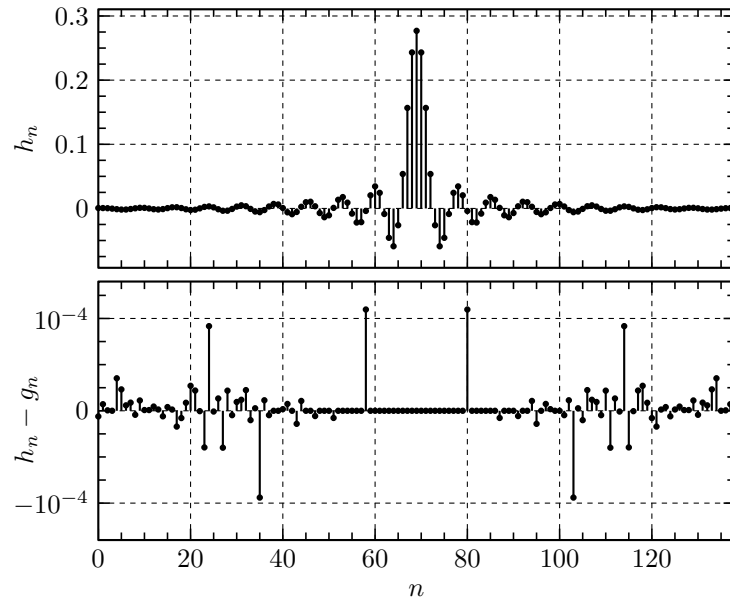


Рис. 30. Линейно-фазовый Фильтр Нижних Частот. Исходная импульсная характеристика h_n и отклонение $h_n - g_n$

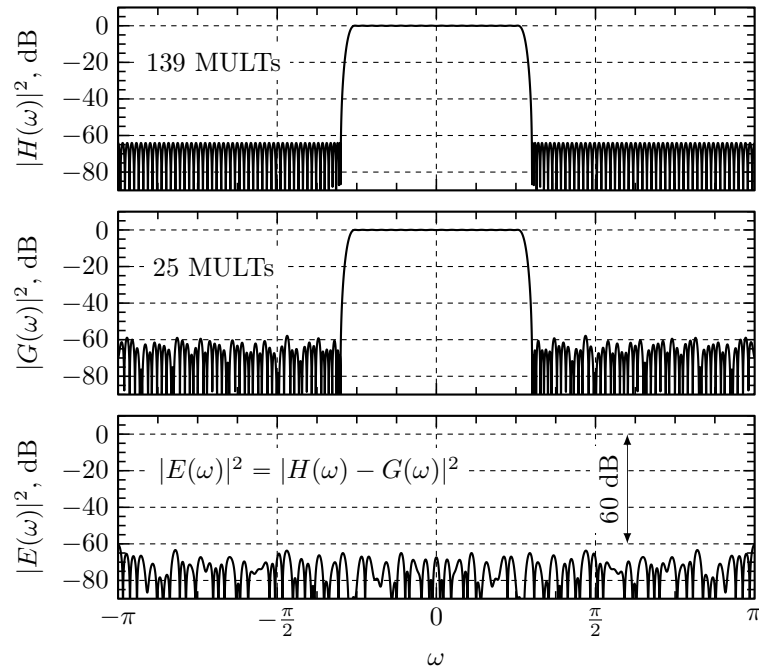


Рис. 31. Линейно-фазовый Фильтр Нижних Частот. Исходная частотная характеристика $H(\omega)$, частотная характеристика $G(\omega)$ и отклонение $E(\omega)$

рованного алгоритмом фильтра также обозначим $\mathbf{g} = \{g_n\}, n \in \overline{0, N-1}$.

На рис. 32 изображена импульсная характеристика исходного полосового фильтра h_n , а также отклонение исходных коэффициентов от коэффициентов фильтра, упрощенного с точки зрения аппаратной реализации $h_n - g_n$. Это отклонение мало и составляет порядка 10^{-5} .

На рис. 33 изображена частотная характеристика исходного фильтра $H(\omega)$, фильтра, построенного на 25 умножителях $G(\omega)$, а также разность частотных характери-

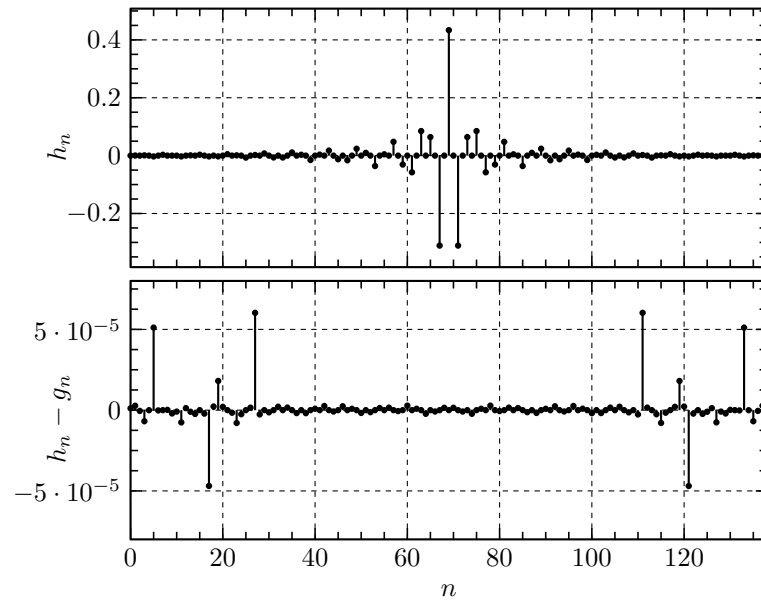


Рис. 32. Линейно-фазовый Полосовой Фильтр. Исходная импульсная характеристика h_n и отклонение $h_n - g_n$

стик $E(\omega) = H(\omega) - G(\omega)$.

Как видно из рис. 33 величина отклонения частотной характеристики фильтра, построенного на 25 умножителях от частотной характеристики исходного фильтра порядка $N = 138$, реализованного в прямой форме, составляет, так же как и для линейно-фазового ФНЧ, -60 dB.

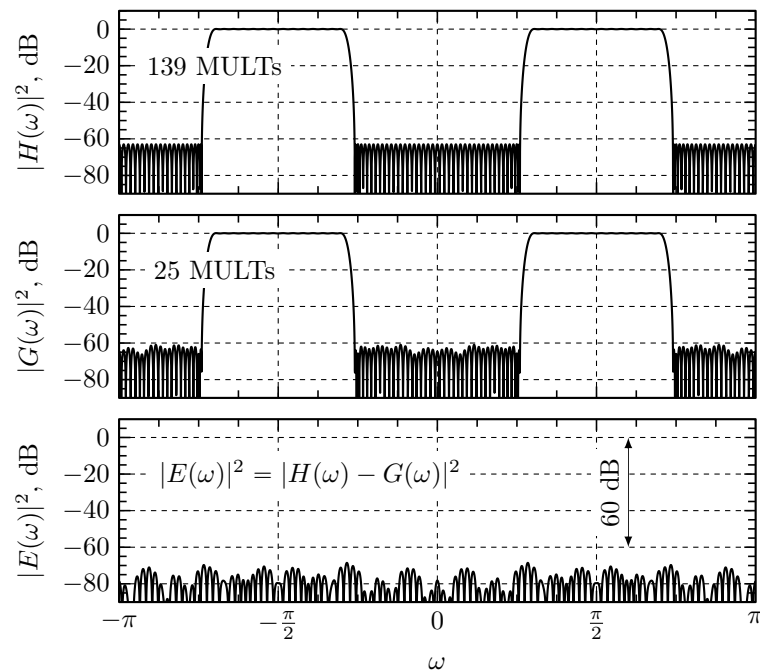


Рис. 33. Линейно-фазовый Полосовой Фильтр. Исходная частотная характеристика $H(\omega)$, частотная характеристика $G(\omega)$ и отклонение $E(\omega)$

Рассмотрим минимально фазовый Фильтр Нижних Частот порядка $N = 138$. Пусть коэффициенты его импульсной характеристики $\mathbf{h} = \{h_n\}, n \in \overline{0, N-1}$. $\mathbf{g} =$

$\{g_n\}, n \in \overline{0, N-1}$ – импульсная характеристика сформированного алгоритмом фильтра. Рассмотрим случай уменьшения количества умножителей до $M = 25$ и до $M = 70$.

Данный пример интересен в связи с тем, что в данном случае импульсная характеристика исходного фильтра $\mathbf{h} = \{h_n\}$ не является симметричной.

На рис. 34 изображена импульсная характеристика исходного фильтра нижних частот h_n и отклонение $h_n - g_n$ исходных коэффициентов от коэффициентов фильтра, построенного на 25 умножителях. Это отклонение составляет порядка 10^{-4} .

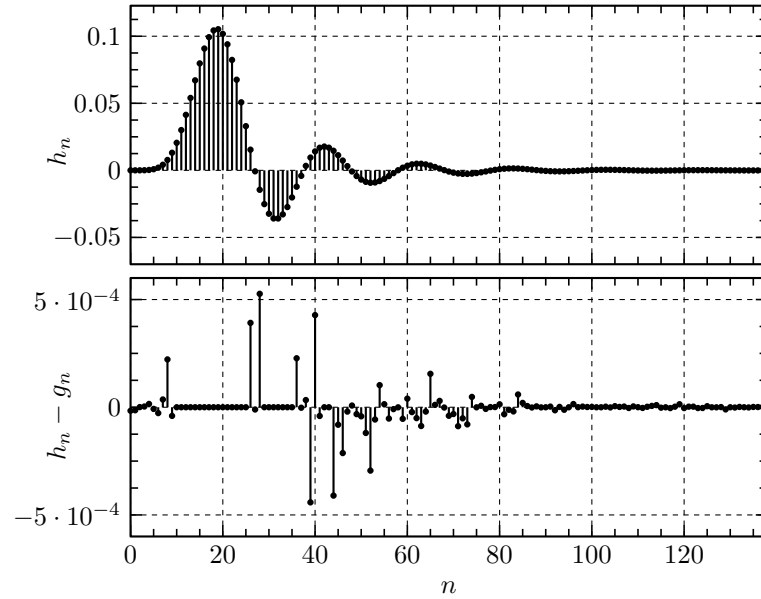


Рис. 34. Минимально фазовый Фильтр Нижних Частот. Исходная импульсная характеристика h_n и отклонение $h_n - g_n$

На рис. 35 изображена частотная характеристика исходного фильтра $H(\omega)$, фильтра, построенного на 25 и 70 умножителях $G(\omega)$, а также разность частотных характеристик $E(\omega) = H(\omega) - G(\omega)$ для обоих случаев.

Из рис. 35 видно, что величина отклонения частотной характеристики фильтра, построенного на 25 умножителях от частотной характеристики исходного фильтра порядка $N = 138$, реализованного в прямой форме, составляет -50 dB.

При этом отклонения частотной характеристики фильтра, построенного на 70 умножителях от частотной характеристики исходного фильтра порядка $N = 138$ составляет -70 dB.

Таким образом, алгоритм, приведённый в статье [fir_quant], позволяет существенно сократить мощность потребляемую цифровым КИХ-фильтром, а также занимаемую площадь кристалла RF-чипсета, за счёт замены части умножителей более простыми с точки зрения аппаратной реализации и затрачиваемых ресурсов битовыми сдвигами и дополнительными сумматорами.

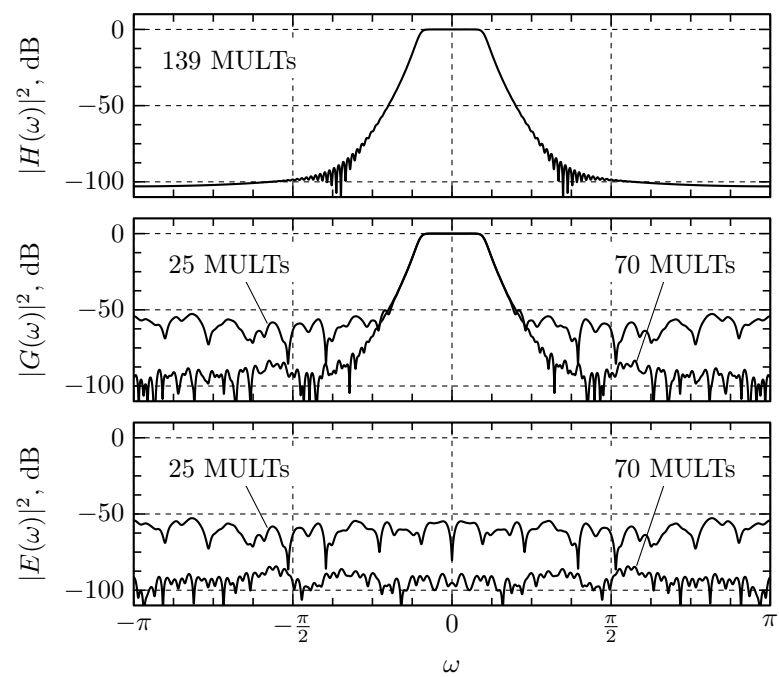


Рис. 35. Минимально фазовый Фильтр Нижних Частот. Исходная частотная характеристика $H(\omega)$, частотная характеристика $G(\omega)$ и отклонение $E(\omega)$

Заключение