

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет радиофизики и электроники

Кафедра системного анализа

**Методы, алгоритмы и программные средства обработки
атомно-эмиссионных спектров**

Дипломная работа

студента 5-го курса

Назарова Петра Владимировича

Руководитель:

заведующий кафедрой системного анализа

доктор физ.-мат. наук, профессор Апанасович

Владимир Владимирович

Рецензент:

канд. физ.-мат. наук, доцент

Семенчик Владимир Григорьевич

“Допустить к защите”

Заведующий кафедрой системного анализа

профессор

В.В. Апанасович

“ ___ ” _____ 2001 г.

МИНСК 2001

СОДЕРЖАНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ СОКРАЩЕНИЙ.....	3
ВВЕДЕНИЕ	4
ГЛАВА 1. ОСНОВНЫЕ ПРИНЦИПЫ И ПРОБЛЕМЫ АТОМНО-ЭМИССИОННОГО АНАЛИЗА.....	5
1.1. Принцип генерирования индукционной плазмы	5
1.2. Контур спектральной линии	8
1.3. Методика работы и проблемы АЭм анализа	10
1.4. Методы обработки спектров	14
ГЛАВА 2. АЛГОРИТМЫ ОБРАБОТКИ НА ОСНОВЕ КЛАССИЧЕСКИХ ПРИЕМОВ	16
2.1. Фильтрация	16
2.2. Коррекция погрешностей монохроматора.....	18
2.3. Коррекция влияний фона плазмы.....	19
2.4. Результаты обработки.....	20
ГЛАВА 3. НЕЙРОСЕТЕВЫЕ МЕТОДЫ ОБРАБОТКИ СПЕКТРОВ.....	23
3.1. Принцип функционирования и структура ИНС.....	23
3.2. Применение МСП для определения концентрации.....	30
ГЛАВА 4. АППРОКСИМАЦИЯ СПЕКТРОВ.....	32
4.1. Аппроксимация спектров методом наименьших квадратов.....	32
4.2. Аппроксимация спектров РБФ-сетями	34
4.3. Двухступенчатая аппроксимация	38
ЗАКЛЮЧЕНИЕ	40
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	41
ПРИЛОЖЕНИЕ 1. ПРОГРАММА ОБРАБОТКИ АЭМ СПЕКТРОВ.....	42

СПИСОК ИСПОЛЬЗОВАННЫХ СОКРАЩЕНИЙ

АЭм	—	атомно-эмиссионный;
ИСП	—	индуктивно-связная плазма;
ВЧ	—	высокочастотный;
с.к.о.	—	средне квадратическое отклонение;
ИНС	—	искусственные нейронные сети;
РБФ	—	радиальные базисные функции;
МСП	—	многослойный персептрон;
МНК	—	метод наименьших квадратов.

ВВЕДЕНИЕ

Атомно-эмиссионный (АЭм) анализ при атомизации пробы в индуктивно-связанной плазме (ИСП) является одним из самых эффективных и массовых методов спектрального анализа благодаря высокой чувствительности и большому числу определяемых элементов. С его помощью выполняют качественный и количественный анализ геологических объектов, продуктов металлургического производства (руд, концентратов, металлов, сплавов, шлаков, сточных вод и т.д.), продуктов химического производства. Особое значение он имеет для определения следов металлов (особенно токсичных) в объектах окружающей среды (в водах, почвах, морских и речных отложениях, в аэрозолях), в биологических объектах (растениях, кормах, тканях и биологических жидкостях животных, продуктах питания), в нефтепродуктах. Для многих перечисленных объектов в США, странах Западной Европы и других промышленно развитых странах ИСП-спектрометрия стала стандартным методом анализа. Высокие аналитические характеристики метода обусловлены высокочастотным индукционным разрядом в аргоне при атмосферном давлении. Он обладает хорошей временной стабильностью, высокой температурой и обеспечивает эффективную атомизацию и возбуждение поступающего в него анализируемого вещества. Тем не менее, для ряда аналитических задач необходимо увеличение чувствительности и других метрологических характеристик ИСП-спектрометров.

Цель настоящей работы состояла в разработке и исследовании методов снижения пределов обнаружения при определении концентраций элементов в эмиссионном спектральном анализе с применением современных методов обработки данных. Её актуальность обусловлена следующим. В настоящее время ИСП-спектрометры являются основным средством проведения рутинных многоэлементных анализов во многих аналитических лабораториях мира, но в Беларуси и в других странах СНГ такие спектрометры до сих пор серийно не выпускаются и находятся в стадии разработки.

Данная работа выполнена с использованием опытного образца прибора, созданного в лаборатории лазерной диагностики плазмы института молекулярной и атомной физики НАН РБ. Автор выражает признательность старшим научным сотрудникам лаборатории ЛДП института молекулярной и атомной физики НАН Беларуси, канд. физ.-мат. наук П. Я. Мисакову и А. В. Исаевичу за предоставленную информацию и помощь в выполнении работы.

ГЛАВА 1. ОСНОВНЫЕ ПРИНЦИПЫ И ПРОБЛЕМЫ АТОМНО-ЭМИССИОННОГО АНАЛИЗА

Прежде чем приступать к описанию методов обработки атомно-эмиссионных спектров необходимо чётко уяснить задачу и проблемы, с которыми придётся столкнуться при обработке. Именно этим вопросам и посвящена данная глава. Кроме того, в этой главе будут рассмотрены принципы работы ИСП спектрометра, физические принципы получения спектров и их форма.

1.1. Принцип генерирования индукционной плазмы

Впервые высокочастотная индукционная плазма в аргоне при атмосферном давлении была получена в начале 40-х годов в Ленинграде (ныне Ст.-Петербург) Бабатом. Он планировал использовать высокочастотный разряд для промышленных целей и генерировал его в закрытых системах при мощности поля на индукторе 30-50 кВт. Возможности плазмотронов как источника возбуждения для эмиссионной спектроскопии были выявлены в 1962 г. Гринфилдом в Англии и Фасселом в США [1].

Исследователи 60-70х гг. экспериментально отработали основные принципы, аппаратуру, в том числе и различные типы горелок, способы введения пробы и наметили области применения высокочастотной индукционной плазмы в анализе.

К настоящему времени плазменная спектрометрия уже прошла революционные стадии и вошла в эволюционный период развития. В этот период постепенно углубляется понимание сложной структуры и плазменного факела и механизмов возбуждения в нем. Исследования были направлены на постепенное снижение пределов обнаружения, мешающих влияний, на улучшение воспроизводимости и правильности, на усовершенствование аппаратуры, программного обеспечения (в том числе алгоритмов обработки), и, наконец, на расширение областей применения метода.

Источником возбуждения в спектрометре является безэлектродный высокочастотный разряд в аргоне, созданный и поддерживаемый в специальной горелке (плазмотроне) (рис. 1.1), представляющей собой сборку из трех, реже двух, концентрически расположенных тщательно подогнанных кварцевых трубок. В зазор

между внешней и промежуточной трубками подается внешний, или охлаждающий, поток газа (аргона или молекулярного газа), по средней трубке – промежуточный поток, по центральной трубке подают транспортирующий поток, который переносит аэрозоль в плазму. Открытый конец горелки окружен охлаждаемой водой индукционной катушкой, соединенной с генератором высокой частоты. Для возбуждения разряда необходима предварительная ионизация газа. Для этого чаще всего используют высоковольтную искру. После того, как газ стал электропроводным, возникает разряд, энергия в который подается магнитным полем. Ток высокой частоты, протекающий через катушку-соленоид, создает магнитное поле, также переменное, направленное внутри катушки вдоль ее оси. Под действием переменного магнитного потока внутри катушки индуцируется вихревое электрическое поле. Силовые линии его представляют собой замкнутые окружности, concentricкие с витками катушки. Это электрическое поле и поддерживает разряд, причем токи также замкнуты и протекают вдоль круговых линий электрического поля. Описанный разряд называют индукционным.[1]

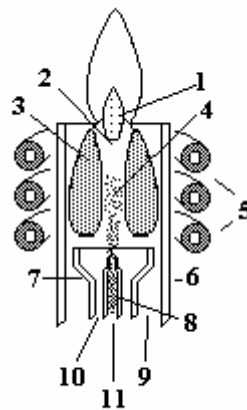


Рис. 1.1. Схема горелки для высокочастотного индукционного разряда.

Зоны в факеле плазмы: 1 - аналитическая; 2 - первичного излучения; 3 - разряда (скин-слой);

4 - центральный канал (зона предварительного нагрева).

Детали плазмотрона: 5 - индуктор; 6 - внешняя трубка; 7 - промежуточная трубка;

8 - центральная трубка.

Газовые потоки: 9 - внешний, 10- промежуточный; 11 - транспортирующий

Вихревой электрический ток нагревает и ионизирует поступающие снизу новые порции газа за счет джоулева тепла. Токопроводящая плазма аналогична

короткозамкнутой вторичной обмотке трансформатора, магнитное поле которой сжимает кольцевой ток в тор (скин-эффект).[1]

Анализируемое вещество в виде аэрозоля подается в потоке аргона в плазму, где осушается, диссоциирует и атомизируется. Аэрозоль пробы достаточно долго (несколько миллисекунд) находится в высокотемпературной (порядка 10000°K) зоне разряда, что наряду с инертной атмосферой (аргон), в которой находятся возбужденные атомы и ионы, обеспечивает полноту атомизации пробы, большую величину отношения сигналов к фону и большую ширину линейной части градуировочных зависимостей. [1]

На рис. 1.2 приведена упрощённая функциональная схема плазменного спектроанализатора. Его основными узлами являются: высокочастотный генератор с системой автоматической регулировки мощности и блоком согласования с плазмой (7); плазменная горелка с системой регулировки газовых потоков и ввода пробы (1-6); спектральные приборы (многоканальные квантометры или сканирующие монохроматоры) (9); фотоприемники и электронные блоки для усиления и интегрирования сигналов (10); и, наконец, электронно-вычислительная машина с внешними устройствами для управления спектрометром и обработки информации (11).

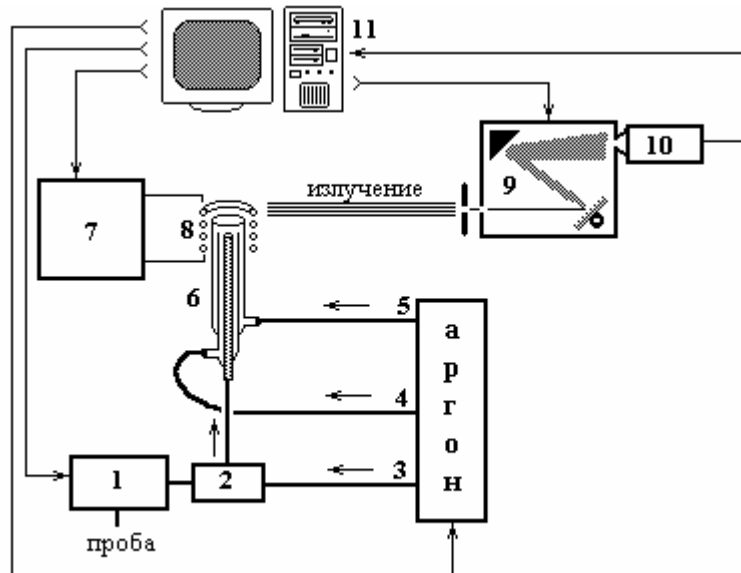


Рис. 1.2. Функциональная схема плазменного анализатора.

- 1 - перистальтический насос; 2 - распылитель пробы; 3 - транспортирующий поток; 4 - промежуточный поток; 5 - внешний (охлаждающий) поток; 6 - горелка; 7 - ВЧ блок питания; 8 - катушка; 9 - монохроматор; 10 - ФЭУ; 11 - ЭВМ.

Конструктивные особенности каждого из этих блоков существенно влияют на характеристики получаемых результатов и определяют производительность приборов.

Использованный при написании этой работы спектрометр был создан в лаборатории лазерной диагностики плазмы института Молекулярной и атомной физики НАН РБ. Прибор способен поочерёдно сканировать интенсивности излучения на длинах волн выбранного диапазона с шагом 0.00165 нм. Полученные спектры называют *сканами*.

1.2. Контур спектральной линии

Для изучения возможностей обработки сканов эмиссионных линий необходимо вначале рассмотреть поведение контуров линий в отсутствии шумов, фона и прочих мешающих факторов. Поэтому рассмотрим теоретическую форму контура спектральной линии (в отсутствии шумов и без учёта тонкой структуры), снимаемой ИСП-спектрометром. Не вдаваясь в подробности физики излучения квантов возбуждёнными атомами и причин уширения спектральных линий, остановимся на тех типах контуров, которые оказывают наибольшее влияние на форму результирующего спектра.

1.2.1. Доплеровский контур – контур, обусловленный тепловым движением излучающих атомов. Для него зависимость интенсивности от длины волны $I(\nu)$ выглядит следующим образом

$$I(\nu) = I_0 \exp \left[-\frac{\mu}{2RT} c^2 \left(\frac{\nu - \nu_0}{\nu_0} \right)^2 \right], \quad (1.1)$$

где I_0 – значение интенсивность в максимуме; ν_0 – частота, на которую приходится максимум; μ – атомная масса; R – универсальная газовая постоянная; T – абсолютная температура излучающего газа.[3]

Здесь предполагается, что движение атомов носит равновесный характер (т.е. подчиняется закону Максвелла) и, следовательно, пространство изотропно.

Для ширины доплеровского контура имеем

$$\Delta v_D = \frac{2v_0}{2} \sqrt{\frac{2RT \ln(2)}{\mu}} \approx 7.16 \cdot 10^{-7} v_0 \sqrt{\frac{T}{\mu}} \quad (1.2a)$$

или в единицах длин волн

$$\Delta \lambda_D = \frac{c\lambda_0}{2} \sqrt{\frac{2RT \ln(2)}{\mu}} \approx 7.16 \cdot 10^{-7} \lambda_0 \sqrt{\frac{T}{\mu}} . \quad (1.2b)$$

1.2.2. Лоренцовский контур. Лоренцовским или дисперсионным, называется контур, для которого $I(\nu)$ имеет вид

$$I(\nu) = I_0 \frac{(\gamma/2)^2}{4\pi^2 c (\nu_0 - \nu)^2 + (\gamma/2)^2} , \quad (1.3)$$

где γ – некоторая постоянная (коэффициент затухания).

Контур, описываемый этой формулой, справедлив для двух случаев: а) естественного расширения линии; б) расширения, вызванного соударениями атомов.[3]

В случае естественного расширения коэффициент затухания

$$\gamma = \gamma_k + \gamma_i = \frac{1}{\tau_k} + \frac{1}{\tau_i} , \quad (1.4)$$

где τ_k и τ_i – времена жизни тех энергетических уровней, при переходе между которыми линия излучается.

Ширина лоренцовского контура в шкале длин волн

$$\Delta \lambda_L = \frac{\gamma \lambda_0^2}{2\pi c} . \quad (1.5)$$

Как было отмечено в [3], для элементов среднего атомного веса даже при комнатной температуре естественная ширина линии на два порядка меньше, чем доплеровская. В опытах же, рассматриваемых в данной работе, температура газа в аналитической области плазмы достигает 6000°К, что позволяет пренебречь естественным уширением линии по сравнению с доплеровским.

Расширение спектральных линий, вызванное соударениями, было впервые рассмотрено Лоренцом. Он исходил из весьма упрощённой модели: электрон внутри атома совершает в течение времени τ_{cm} , соответствующего времени пролёта свободного пути, гармоническое колебание с частотой ν_0 . В момент столкновения с другим атомом колебания обрываются. Полное излучение складывается из отдельных некогерентных синусоидальных колебаний длительностью τ_{cm} . Путём разложения сумм таких

колебаний в ряд Фурье находим распределения интенсивности по частотам. Расчёт приводит к формуле (1.3), где $\gamma/2$ заменяется на $1/\tau_{cm}$. Здесь τ_{cm} - среднее время между двумя столкновениями.[3]

По кинетической теории газов

$$\frac{1}{\tau_{cm}} = 4\sigma^2 N_0 \sqrt{\frac{\pi RT}{\mu}}, \quad (1.6)$$

где N_0 – число частиц в единице объёма; σ^2 - эффективное сечение столкновения; μ – атомарный вес. Отсюда на основании равенства (1.5), в котором надо положить $\gamma=2/\tau_{cm}$, получим для ширины линии

$$\Delta\lambda_L = \frac{4\sigma^2 N_A p \lambda_0^2}{c \sqrt{\pi \mu RT}}, \quad (1.7)$$

где N_A – число Авогадро; p – давление светящегося газа.

1.2.3. Смешанный контур (Фойхтовский контур). При одновременном действии причин ведущих и к доплеровскому и к лоренцовскому расширениям, задача усложняется и имеет лишь приближённое решение.

Однако из (1.2б) и (1.7) видно, что $\Delta\lambda_D \sim (T)^{1/2}$, а $\Delta\lambda_L \sim (T)^{-1/2}$, следовательно при достаточно высоких температурах основное влияние на вид смешанного контура оказывает доплеровское уширение.

1.3. Методика работы и проблемы АЭм анализа

1.3.1. Методика работы. Определение неизвестных концентраций элементов в пробах различного состава с помощью ИСП-спектрометра проводится методом сравнения величины интенсивности аналитической спектральной линии ΔI или площади под её контуром ΔS (рис. 1.3), полученных при этой концентрации, с калибровочным графиком, полученным по ряду предварительно приготовленных эталонных растворов. Обе величины (ΔI и ΔS) в области малых концентраций имеют линейную зависимость от концентрации. [2]

Проиллюстрируем вышесказанное на примере интенсивностей. Пусть известен набор интенсивностей линии I_k , и соответствующие им концентрации C_k , $k=1 \dots M$.

Тогда коэффициенты прямой (1.8), наилучшим образом проходящей через совокупность точек (I_k, C_k) , задаются равенствами (1.9, 1.10).

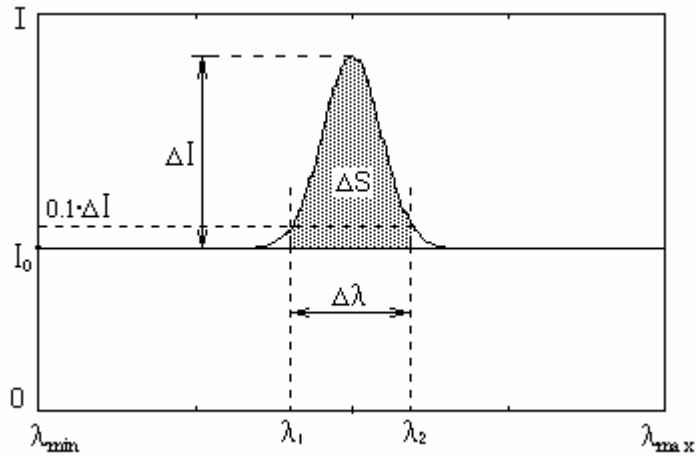


Рис. 1.3. Параметры скана.

$$C(I) = a \cdot I + b, \quad (1.8)$$

$$a = \frac{\sum_{k=1}^M I_k C_k - \frac{1}{N} \sum_{k=1}^M I_k \sum_{k=1}^M C_k}{\sum_{k=1}^N I_k^2 - \frac{1}{N} \left(\sum_{k=1}^N I_k \right)^2}, \quad (1.9)$$

$$b = \frac{1}{N} \left(\sum_{k=1}^N C_k - a \sum_{k=1}^N I_k \right). \quad (1.10)$$

Для оценки эффективности обработки предлагается сравнивать относительные с.к.о. параметров скана от калибровочных параметров для заданной концентрации. Пусть для некоторой концентрации C имеется набор из n сканов. Тогда погрешность $s(C)$ параметра $Y(C)$ запишется в виде:

$$s(C) = \frac{1}{\bar{Y}(C)} \sqrt{\frac{\sum_{i=1}^n (Y_i(C) - \bar{Y}(C))^2}{n-1}}, \quad (1.11)$$

где $\bar{Y}(C)$ – среднее значение параметра, полученное из калибровочного графика для концентрации C ; $Y_i(C)$ – параметр i -го скана при концентрации C .

1.3.2. Проблемы анализа на ИСП спектрометре. На рис. 1.4 представлены экспериментальные сканы линии свинца 220.353 нм. Видно, что они значительно отличаются от идеализированного скана (рис.1.3).

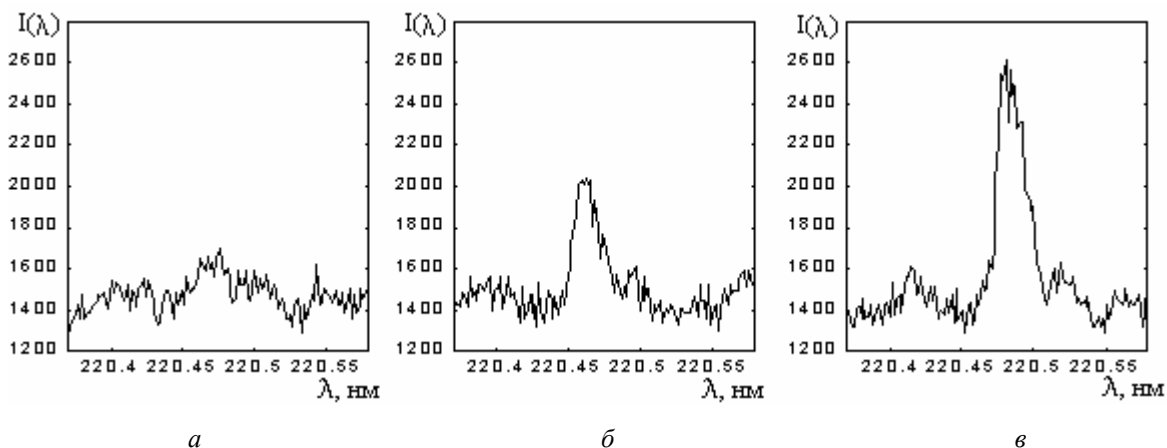


Рис. 1.4. Линия свинца на 220.353 нм при концентрациях
а) 1 мг/л; б) 5 мг/л; в) 10 мг/л.

Рассмотрим те факторы, в силу которых работа со сканами становится трудоёмкой и требует создания алгоритмов обработки.

1) Сканы зашумлены в результате неидеальности ФЭУ, шумов в линии передачи и т.д. Смещённая оценка плотности вероятности суммарного шума представлена на рис.1.5.

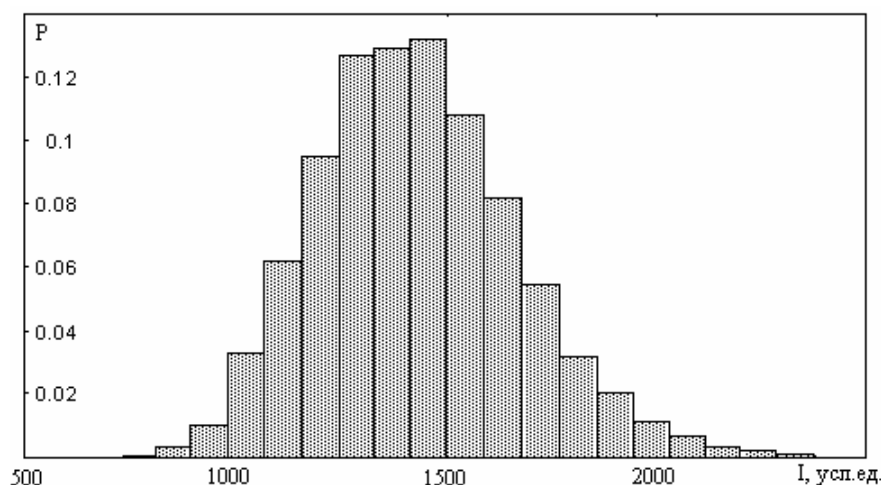


Рис. 1.5. Распределение плотности вероятности шумовых импульсов
(с учётом фона плазмы) в сканах.

2) Вследствие неточностей механики монохроматора, получаемые сканы могут быть сдвинутыми друг относительно друга, что делает невозможным их тривиальное усреднение (рис. 1.6).

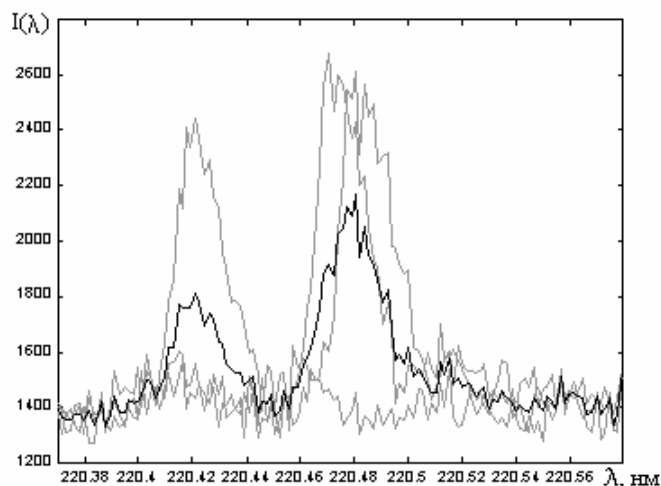


Рис. 1.6. Результат тривиального усреднения сканов.

3) Контур анализируемой линии суммируется не с постоянной величиной фонового излучения, как показано на идеализированном рисунке 1.3, а с детерминированным контуром фонового излучения сложной формы. При малых концентрациях аналита это приводит к тому, что даже в случае большого числа усредняемых сканов, пик линии аналита теряется в пиках фона (рис. 1.7).

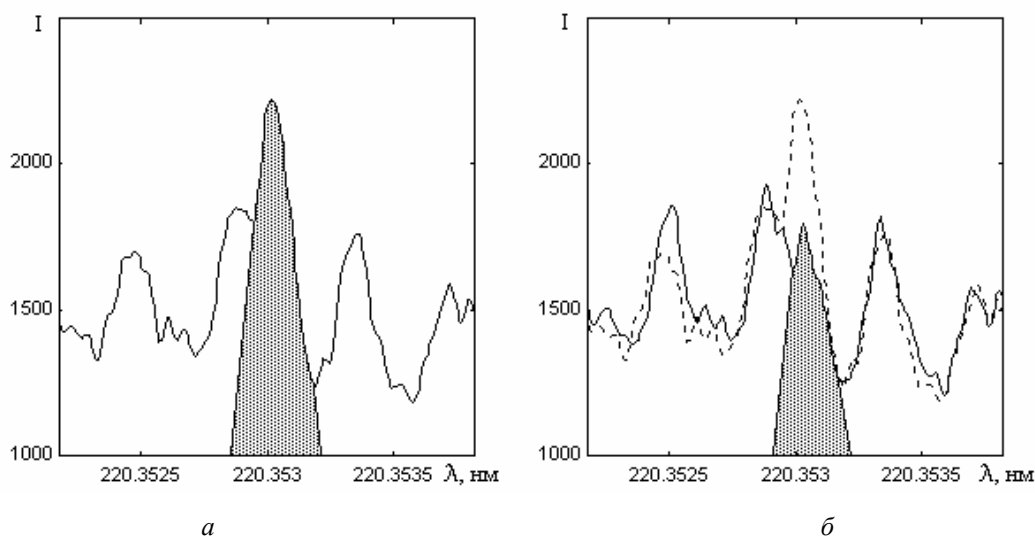


Рис. 1.7. Маскировка аналитического пика (выделен серым) пиками детерминированной составляющей фонового излучения при концентрациях аналита:

а) 2 мг/л; б) 0.5 мг/л.

Кроме того, свечение факела не достаточно стабильно, поэтому возможен дрейф интенсивности фонового излучения.

4) Характеристики сканов зависят от целого ряда внутренних параметров спектро-анализатора. Некоторые из них изменяются при каждом его включении, что отрицательно влияет на воспроизводимость опыта.

Учесть все эти факторы в рамках классического математического подхода к моделированию не представляется возможным, поэтому задача идентификации пиков и нахождения их параметров требует разработки новых методов, в том числе использующих последние достижения науки.

1.4. Методы обработки спектров

Для обработки спектров можно предложить два принципиальных подхода:

- 1) коррекция спектров с последующим выделением спектральной линии в чистом виде,
- 2) прямое определение концентрации анализируемого элемента (рис. 1.8).

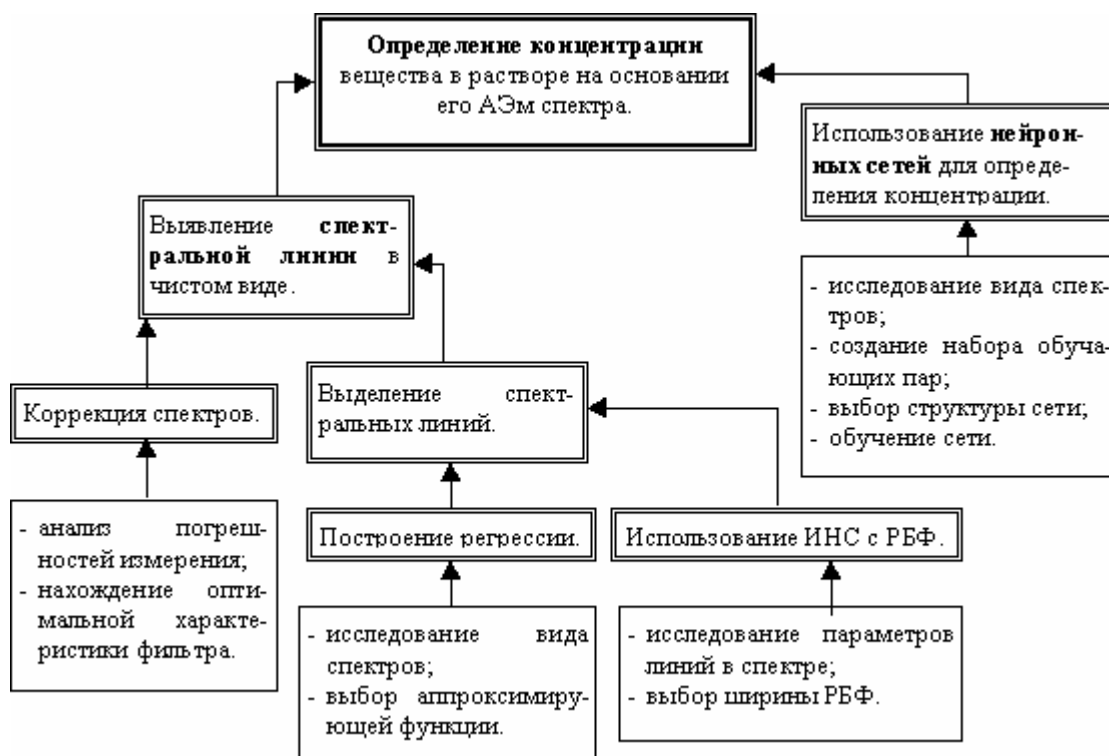


Рис. 1.8. Методы коррекции атомно-эмиссионных спектров.

Коррекцию спектров можно проводить опять же двумя способами: либо фильтрацией сканов, что требует выбора оптимальной АЧХ фильтра, либо их аппроксимацией с помощью аналитических функций. Для проведения аппроксимации необходимо выбрать вид аппроксимирующей функции и метод аппроксимации.

В последующих главах последовательно рассматриваются все выше перечисленные методы.

ГЛАВА 2. АЛГОРИТМЫ ОБРАБОТКИ НА ОСНОВЕ КЛАССИЧЕСКИХ ПРИЕМОМ

Как уже отмечалось ранее, различные негативные факторы (см. п.1.3) не позволяют достичь малых (по мировым стандартам) пределов обнаружения, что заставляет создавать специальные алгоритмы обработки АЭм спектров. В этой главе будут продемонстрированы специально созданные алгоритмы коррекции, основанные на таких классических методах, как преобразование Фурье, фильтрация и построение корреляционных функций.

2.1. Фильтрация

Можно предложить несколько путей упразднения стохастических шумов в сканах. Одним из наиболее очевидных является низкочастотная (НЧ) фильтрация, которая может быть осуществлена, например, с помощью цифровых фильтров. Было рассмотрено несколько фильтров (фильтр Буттерворта, эллиптический, Чебышева [4]) но ни один из них не дал удовлетворительных результатов при коррекции сканов (рис. 2.1в,г).

Поэтому фильтрация сканов производилась с помощью спектральных окон и преобразований Фурье [4]. Пусть $I(i)$ — отсчёты скана $i=1..N$, $W(k)$ — спектральное окно. Алгоритм фильтрации выглядит следующим образом.

1. Находилось преобразование Фурье корректируемого скана.

$$S_k = \sum_{i=1}^N I_i e^{-j\left(\frac{2\pi}{N}\right)ki}, \quad k = 1..N. \quad (2.1)$$

2. Действительная и мнимая его части умножались на спектральное окно требуемой формы.

$$S_k^* = S_k \cdot W_k, \quad k = 1..N. \quad (2.2)$$

3. Производилось обратное преобразование Фурье.

$$I_i^* = \frac{1}{N} \operatorname{Re} \left[\sum_{k=1}^N S_k^* e^{j\left(\frac{2\pi}{N}\right)ki} \right], \quad i = 1..N. \quad (2.3)$$

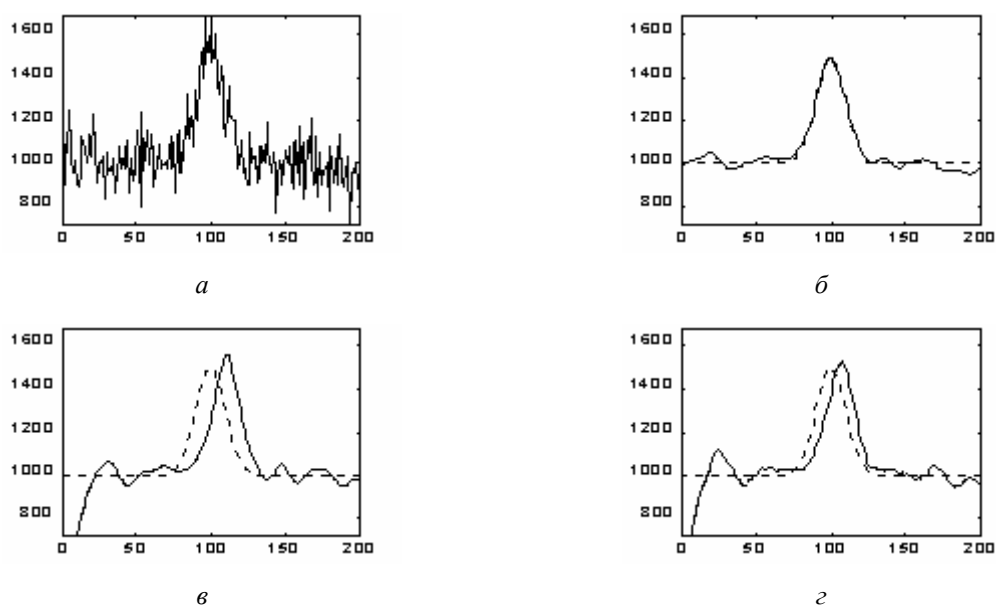


Рис. 2.1. Результаты работы различных фильтров;
 а) исходный скан, б) фильтрация с помощью спектрального окна, в) фильтр Буттерворта,
 г) эллиптический фильтр.

Этот метод позволяет создавать фильтры с любой наперед заданной характеристикой. В качестве фильтрующих окон были апробированы следующие окна: прямоугольное (2.4), экспоненциальное (2.5) и гауссово (2.6).

$$W_k = \begin{cases} 1, & k \leq k_{nop} \\ 0, & k > k_{nop} \end{cases}, \quad k = 1..N/2, \quad (2.4)$$

$$W_k = e^{-k/\sigma}, \quad k = 1..N/2, \quad (2.5)$$

$$W_k = e^{-(k/\sigma)^2}, \quad k = 1..N/2, \quad (2.6)$$

где ω_{nop} , σ — параметры, отвечающие за ширину окна. Отсчёты окон (2.4-2.6) для $k > N/2$ получаются зеркальным отражением отсчётов для $k \leq N/2$.

Для определения оптимальных параметров был проведён вычислительный эксперимент. Моделировался набор сканов (256 отсчётов каждый) с различными соотношениями сигнал/шум. В качестве шума брался гауссов. Для каждого значения зашумлённости генерировалось по 200 сканов. Эти сканы подвергались фильтрации окнами (2.4 - 2.6) при различных значениях параметров. Было установлено, что хотя эти окна позволяют снизить погрешности до практически одинаковых величин, наилучшим, с точки зрения формы контура линии, является окно (2.6). Это связано с тем, что (2.6) описывает спектр сигнала гауссовой формы вида (1.1) с точностью до нормировочного множителя. Ширина полосы пропускания оптимального фильтра

зависит от соотношения сигнал/шум. У фильтра с АЧХ вида (2.6) эмпирическое соотношение для коэффициента ширины окна выглядит следующим образом:

$$\sigma = \gamma \sqrt{\frac{1}{7.3 \cdot V}}, \quad (2.7)$$

где γ — относительная ширина линии по уровню 0.5, V — среднеквадратическое отклонение шума. Результаты фильтрации представлены на рис. 2.2.

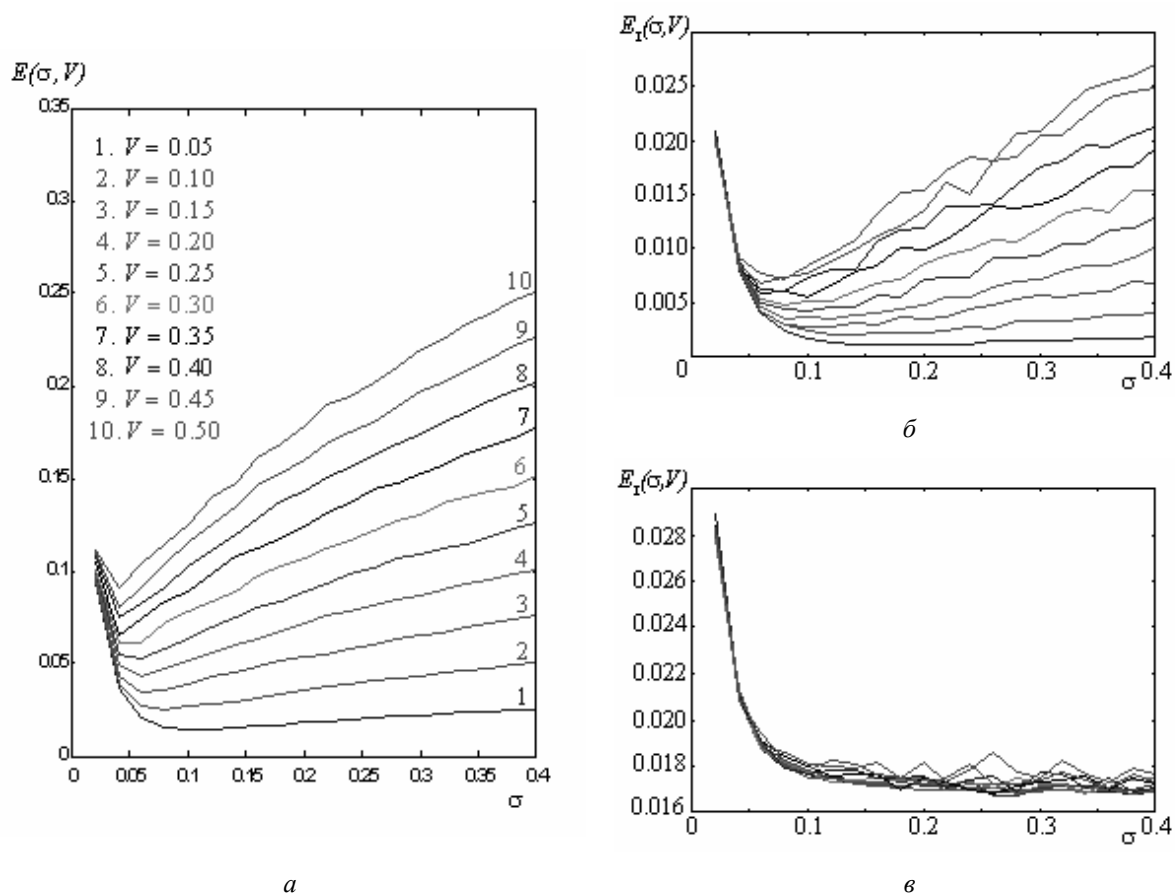


Рис. 2.2. Фильтрация гауссовым окном;
а) ошибка контуров, б) ошибка по интенсивности, в) ошибка по площади.

2.2. Коррекция погрешностей монохроматора

Из рис. 1.6 видно, что в результате погрешностей монохроматора сканы могут оказаться смещёнными по длине волны. Это влечёт за собой ряд трудностей при обработке, в том числе затрудняет выбор "ворот" при вычисления площади под

контуром. Для устранения этих погрешностей можно применить аппарат корреляционных функций.

Пусть имеются некоторый сигнал $X(t)$ с нулевым математическим ожиданием. Получим из него сигнал $Y(t)$, сдвинув на τ_0 . Если теперь построить функцию корреляции $R(\tau)$ между этими двумя сигналами

$$R(\tau) = \int_0^T X(t)Y(t + \tau)dt, \quad (2.8)$$

то ее максимум придется на $\tau = -\tau_0$.

Для совмещения сканов использовался некоторый эталонный скан. В качестве эталона может быть использована, например, функция Гаусса с максимумом в центре. Однако опыт показывает, что для достижения лучших результатов эталон должен являться сканом спектральной области вокруг линии аналита. При этом желательно наличие линии с интенсивностью приблизительно в три раза превышающей фоновые линии (эмпирический результат). Недостающие отсчеты скана заполнялись средним значением.

На практике корреляционные функции вычислялись через Фурье-образы сигналов по теореме Винера-Хинчина [4]. Важно помнить, что перед вычислением корреляционной функции сигналы должны быть центрированы, т.е. их математическое ожидание должно равняться 0.

2.3. Коррекция влияний фона плазмы

Фон плазмы вносит две существенные погрешности. Во-первых, его дрейф сильно сказывается на разбросе параметров аналитической линии; во-вторых, при малых концентрациях происходит маскировка линии пиками фона. Если первую погрешность довольно просто скорректировать, вычитая из скана его среднее, то вторая требует внимательного изучения области спектра вокруг эмиссионной линии. Один из возможных, но не всегда реализуемых на практике способов коррекции маскировки описан ниже.

Чтобы устранить влияние фона плазмы, можно перед началом экспериментов снять спектр фона в области аналитической линии, а затем вычесть его из отсчетов скана. Это дает положительный результат, только если экспериментальный скан и скан

фона не смещены друг относительно друга. Поэтому приходится предварительно совмещать скан фона и экспериментальный скан с помощью корреляционных функций. Но корреляционная функция может указать не на смещение между фоном и корректируемым сканом, а на смещение между максимумом скана и максимальным фоновым пиком.

2.4. Результаты обработки

Всё сказанное выше можно обобщить в одном алгоритме, следующего вида.

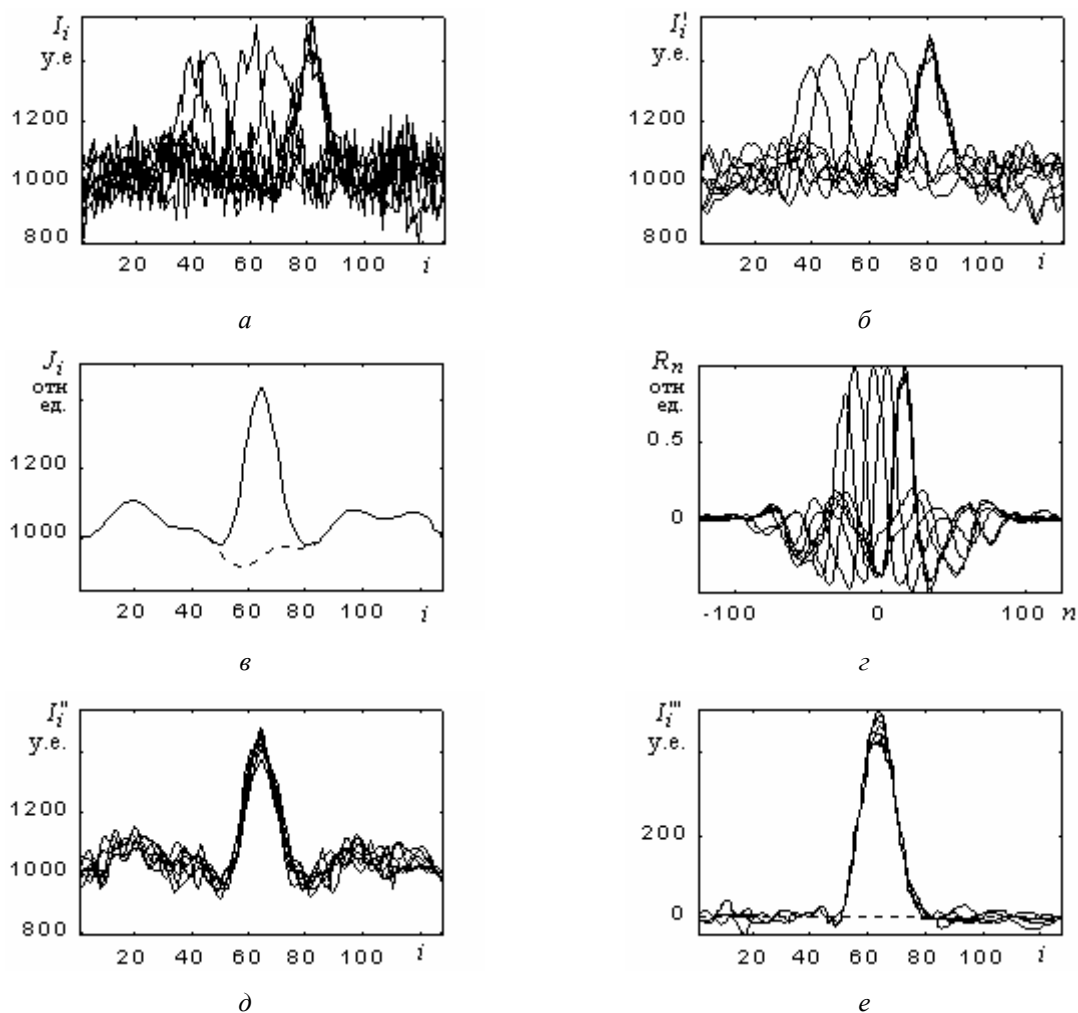


Рис. 2.3. Стадии работы алгоритма на примере линии свинца 261.218 нм;
a) исходные сканы, *б*) сканы после фильтрации, *в*) эталон и фон, *г*) корреляционные функции, *д*) совмещённые сканы, *е*) сканы без фона плазмы

1. Низкочастотная фильтрация сканов (рис. 2.3б).
2. Построение корреляционных функций и совмещение сканов (2.3г,д).
3. Коррекция фона плазмы (рис. 2.3е).

В качестве иллюстрирующей линии была выбрана линия свинца 261.218 нм, поскольку она находится между двумя ярко выраженными фоновыми пиками. Скан фона и эталон строились подбором аналитических функций.

Работа алгоритма тестировалась на 8-ми концентрациях свинца: 0, 0.5, 1, 2, 3, 4, 5, 10 мг/л, на линии 220.353 нм, и 4-х концентрациях меди: 0.01, 0.025, 0.05, 0.1 мг/л, на линии 324.754 нм. Результаты работы приведены на рис. 2.4, 2.5.

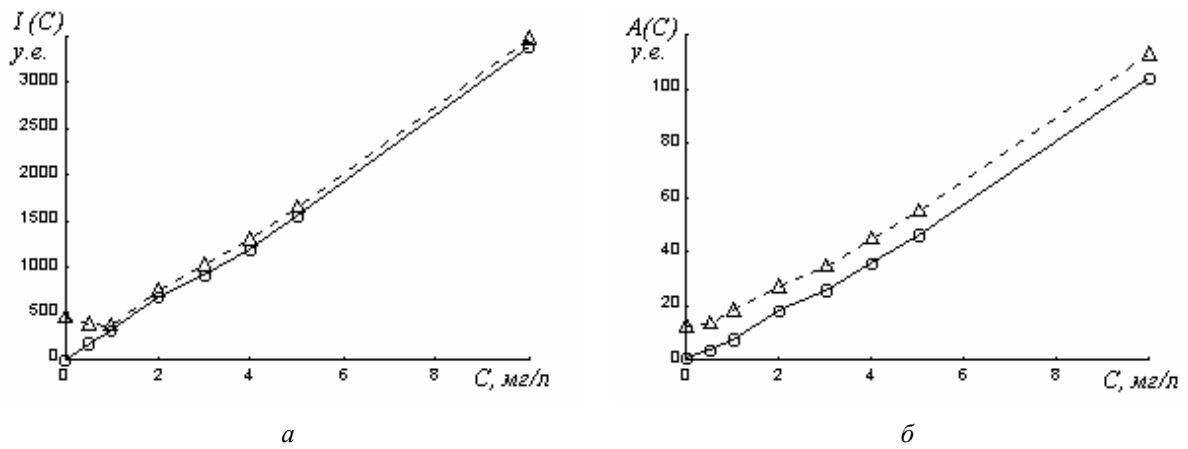


Рис. 2.4. Калибровочные графики (Pb 220.353 нм);

а) по интенсивности, б) по площади.

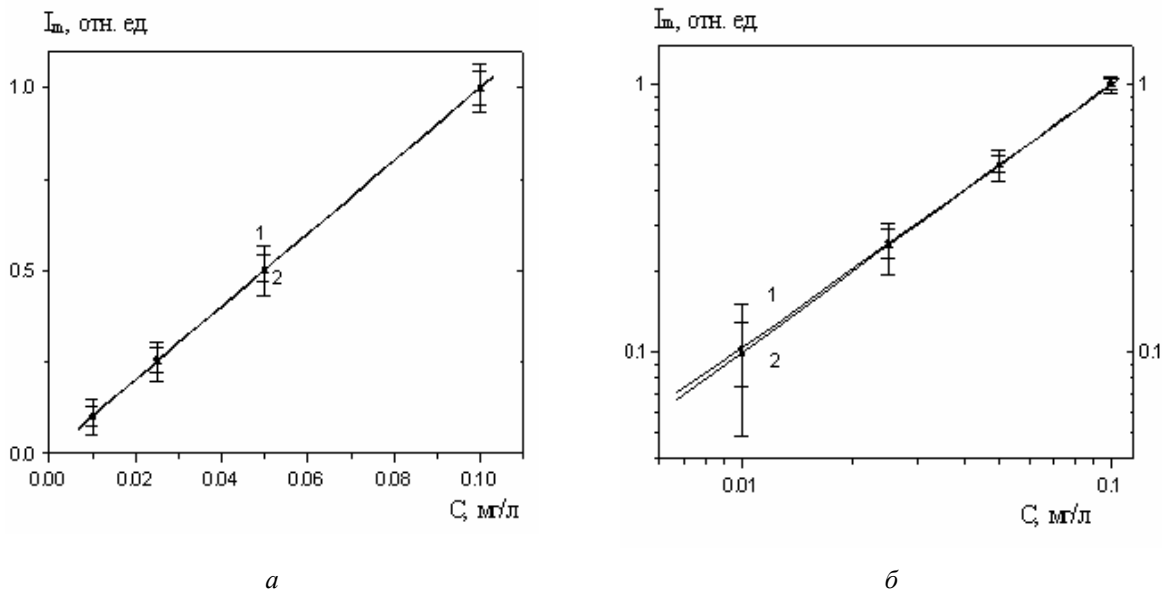


Рис. 2.5. Нормированные калибровочные графики по интенсивности (Cu 324.754 нм);

а) линейный масштаб, б) логарифмический масштаб.

На рис. 2.4 заметно снижение пределов обнаружения (по интенсивности). Рис. 2.5 иллюстрирует уменьшение погрешностей определения концентраций.

Применение классических методов коррекции позволило снизить предел обнаружения свинца в 2 раза, и уменьшить относительное с.к.о. значений по интенсивности в 2 раза, и по площади в 1.7 раза. По меди было получено снижение относительного с.к.о. в 1.6 и в 1.2 раза соответственно.

ГЛАВА 3. НЕЙРОСЕТЕВЫЕ МЕТОДЫ ОБРАБОТКИ СПЕКТРОВ

Искусственные нейронные сети (ИНС) позволяют успешно решать многие трудно формализуемые задачи. К числу таких задач можно отнести и анализ атомно-эмиссионных спектров в спектрометрах с индуктивно связанной плазмой с целью определения малых концентраций элементов в пробе.

Одна из первых функционирующих моделей ИНС была предложена Ф. Розенблаттом в 1965 г., который назвал эту сеть "*персептроном*". Им же была доказана теорема об обучении такой сети. Однако в 1969 г. в своей работе о персептронах [7] Минский и Пейперт строго показали, что однослойные персептроны имеют большие ограничения при обучении и работе. Например, невозможно обучить однослойный персептрон такой, казалась бы, простой операции как "исключающее или". Поскольку в то время отсутствовали методы обучения многослойных сетей, то интенсивность исследований в этой области резко снизилась. Однако, несмотря на это, несколько оптимистически настроенных учёных продолжали работу над разработкой алгоритмов обучения и моделей, и как показало время, не напрасно. За два десятка лет, которые нейронные сети находились в своеобразной опале, была создана мощная теоретическая база для исследований, и, что наиболее важно, появились новые технологии, упростившие моделирование и улучшившие возможности для внедрения и исследования нейронных сетей. Всё это и обусловило всплеск интереса к ним в 80-х годах, после того как были созданы методы обучения и работы с многослойными сетями и разработаны сети с латеральными и обратными связями.

3.1. Принцип функционирования и структура ИНС

3.1.2. Искусственный нейрон. Маккалок и Питтс, заложившие основы нейросетевого подхода к вычислениям, в 40х гг. изучали совокупность формальных нейронов, обменивающихся информацией посредством направленных связей. Формальный нейрон Маккалока - Питтса состоит из набора межсоединений, сумматора и нелинейного оператора. Значение каждого входа X_i умножается на свой вес W_i , затем суммируется и из суммы вычитается порог θ . После этого на результат действует

некоторая активационная функция F и преобразованная таким образом информация подаётся на выход нейрона Y (рис. 3.1). [5,6]

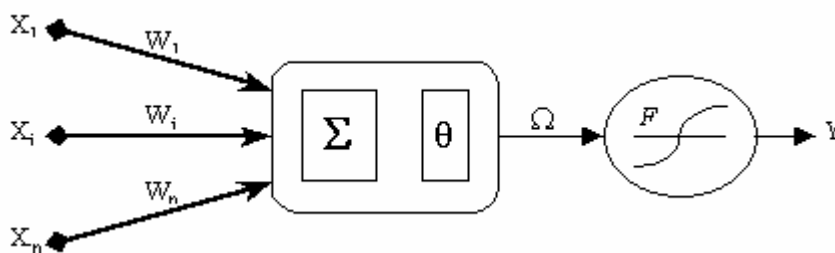


Рис. 3.1. Модель искусственного нейрона.

Таким образом, выход каждого нейрона вычисляется по формуле:

$$Y = F \left\{ \sum_{i=1}^n (X_i W_i) - \theta \right\} \quad (3.1)$$

В дальнейшем условимся обозначать сумму произведений $X_i W_i$ через переменную Ω .

Был проведен ряд опытов, результаты которых показали целесообразность использования в выходной функции порога θ , чтобы избавить её от нечётности и симметричности вообще, так как и то и другое влечёт за собой симметричность расположения классов в гиперпространстве признаков.

3.1.2. Влияние активационных функций. В качестве выходных функций нейронов принято использовать ограниченные на множестве значений нелинейные функции, такие как: пороговая (случай сигнума) (рис. 3.2а), комбинации пороговой и линейной функций, сигмоидальная функция (рис. 3.2б), однотипная с ней функция гиперболического тангенса (рис. 3.2в), а также радиальные базисные функции (рис. 2.2г).

При пороговой функции (3.2) нейрон остаётся неактивным до тех пор, пока суммарный сигнал с его входов не достигает порогового значения. Когда этот порог достигнут, нейрон возбуждается и даёт на выходе 1.

$$Y = \begin{cases} 0, & \Omega \leq \theta \\ 1, & \Omega > \theta \end{cases} \quad (3.2)$$

Пороговая функция наиболее точно моделирует нелинейную передаточную характеристику биологического нейрона и благодаря своей простоте представляет

нейронным сетям большие возможности. При классификации образов, ИНС с такой активационной функцией разбивает пространство на многогранники [6,5].

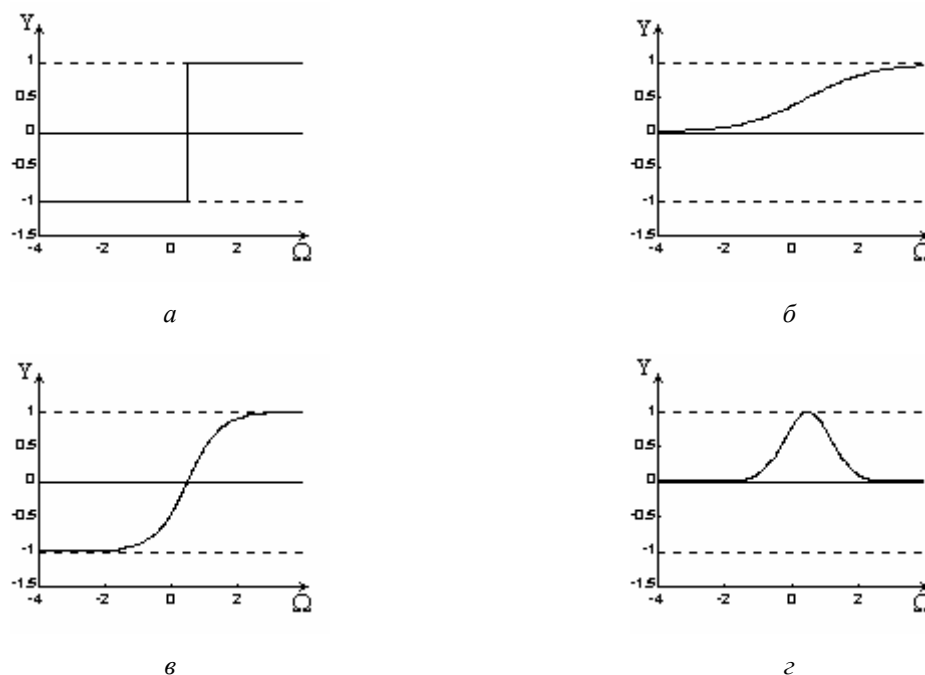


Рис 3.2. Наиболее распространённые виды активационных функций (порог равен 0.5); а) пороговая; б) сигмоидальная; в) гиперболический тангенс; г) радиальная базисная.

При использовании сигмоидальной (иначе логистической) функции (3.3) диапазон изменений величины X сужается таким образом, что при любых значениях X значения Y принадлежат некоторому конечному интервалу. К тому же эта функция удобно дифференцируема, что немаловажно в алгоритмах обучения [5,8].

$$Y = \frac{1}{1 + e^{-(\Omega-0)}}. \quad (3.3)$$

Функция гиперболического тангенса является расширением логистической функции на интервал $(-1,1)$ и имеет вид

$$Y = 1 - \frac{2}{1 + e^{2(\Omega-0)}}. \quad (3.4)$$

Наиболее интересным и перспективным на сегодняшний день (особенно в вопросах интерполяции и аппроксимации) является класс т.н. *радиальных базисных функций* (РБФ), к которым принято относить функции, имеющие глобальный экстремум и ведущие себя монотонно по мере удаления от него. Например, к радиальным базисным функциям относится функция вида:

$$Y = e^{-\frac{1}{2}(\Omega-0)^2}. \quad (3.5)$$

Подробнее о радиальных базисных функциях и РБФ-сетях будет сказано в гл. 4.

3.1.3. Модели и обучение нейронных сетей. Очевидно, что вычислительная ценность нейрона, представленного на рис. 3.1 невелика. Такой нейрон (в сущности, это персептрон Розенблатта с одним выходом) позволяет разделять точки в n -мерном пространстве одной гиперплоскостью. Если же мы объединим некоторое количество нейронов в сеть, то получим классификатор, способный разделять заданные точки произвольными гиперповерхностями, интерполировать и экстраполировать значения в произвольных точках пространства, значения классов, в которых априори неизвестно.

Информационная ценность нейронной сети заключается в способе соединения нейронов (модели сети) и перенастраиваемых весовых коэффициентах каждого нейрона. Чтобы заставить сеть работать для решения конкретной задачи, веса перестраивают определённым образом. Этот процесс называют обучением сети. Алгоритмы обучения принято разделять на два больших класса: обучение с учителем (при заданной классификации) и без учителя (при неизвестной классификации). В самом общем случае обучение с учителем представимо в виде следующего алгоритма:

1) Выбирается множество параметров, по которым будет проводиться классификация объекта. На языке многомерных пространств это соответствует выбору размерности пространства.

2) Задаётся множество объектов (с их параметрами), принадлежащих к известным классам. Совокупность входных параметров и целевого класса объекта называют обучающей парой.

3) Параметры каждого объекта последовательно подаются на входы сети. Сеть обрабатывает их и выдаёт результат классификации.

4) Результаты классификации сравниваются с целевыми классами, и генерируется вектор отклонений полученных классов от целевых. Если суммарная ошибка меньше некоторой наперёд заданной, то алгоритм прекращает свою работу.

5) Веса сети корректируются с использованием вектора отклонений и, в общем случае, входных и выходных значений сети. Алгоритм переходит к п. 3).

При обучении без учителя, как правило, либо корректируют веса таким образом, чтобы не сильно отличающиеся друг от друга входы сети вызывали похожие выходы, либо сеть сама, в процессе функционирования, перестраивает свои веса по какому либо алгоритму.

Среди всего многообразия моделей нейронных сетей, появившихся в последнее время, следует, пожалуй, выделить класс т.н. многослойных перцептронов (МСП), как самый распространённый и изученный. Многослойным перцептроном (рис. 3.3) называют нейронную сеть, нейроны в которой расположены слоями, причём нейроны каждого слоя связаны только с нейронами предыдущего слоя.

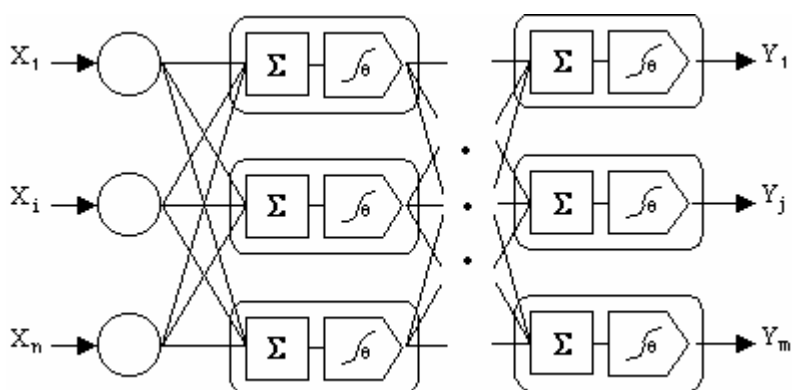


Рис. 3.3. Многослойный перцептрон.

Для сетей этого класса существует довольно хороший "классический" метод обучения — т.н. метод обратного распространения ошибки (*Back Propagation Error*), который относится к разряду методов градиентного спуска [5,6,8]. Он сыграл не последнюю роль в пробуждении интереса к нейронным сетям в начале 80-х, а его модификации и по сей день остаются лучшими для рассматриваемого класса сетей.

3.1.4. Метод обратного распространения ошибки. С использованием процедуры обратного распространения решаются задачи, в которых для некоторого входного вектора $X(x_1, \dots, x_n)$ сеть на выходе формирует требуемый вектор $Y(y_1, \dots, y_n)$. На этапе обучения используется ряд пар векторов (X, Y^*) , называемых обучающей парой, где Y^* – целевой вектор. Используя обучающие пары, сеть подстраивает свои веса таким образом, чтобы адекватно реагировать на входной вектор X . Рассмотрим работу алгоритма обратного распространения ошибки в случае, когда в качестве активационной функции выступает сигмоидальная функция.

При инициализации сети случайным образом задаются начальные веса и сдвиги сети. Алгоритм обучения сети обратного распространения включает в себя следующие шаги:

1. Выбрать очередную обучающую пару (X, Y^*) из обучающего множества и подать входной вектор X на вход сети.
2. Вычислить выход сети Y .
3. Вычислить разность между реальным (вычисленным) выходом сети и требуемым выходом (целевым вектором обучающей пары).
4. Подкорректировать веса сети так, чтобы минимизировать ошибку.
5. Повторить шаги с 1 по 4 для каждого вектора обучающего множества до тех пор, пока ошибка на всем множестве не достигнет приемлемой величины.

Шаги 1 и 2 используются как на этапе обучения сети, так и при функционировании уже обученной сети.

Вычисления в сети выполняются послойно. На шаге 3 каждый из выходов сети Y вычитается из соответствующей компоненты целевого вектора с целью получения ошибки. Эта ошибка используется на шаге 4 для коррекции весов сети, причем величина изменений определяются алгоритмом обучения.

Шаги 1 и 2 можно рассматривать как "проход вперед", а 3 и 4 — как проход назад, так как сигнал ошибки распространяется обратно по сети и используется для подстройки весов. Эти два прохода можно выразить математически.

На входе имеем вектор X , на основе которого вычисляется выходной вектор Y . Вектор Y вычитается из целевого вектора Y^* с целью получения ошибки ε :

$$\varepsilon = Y^* - Y. \quad (3.6)$$

Величина Ω каждого нейрона первого слоя вычисляется как взвешенная сумма входов нейрона. Затем активационная функция F сжимает Ω и дает величину OUT для каждого нейрона в этом слое. Полученное выходное множество OUT является входом для следующего слоя. Процесс повторяется слой за слоем, пока не будет получено заключительное множество сети.

На этапе обратного прохода происходит подстройка весов выходного слоя. Так как для каждого нейрона выходного слоя задано целевое значение, то подстройка весов легко осуществляется с помощью дельта правила. Внутренние слои не имеют целевых значений и называются скрытыми слоями.

Рассмотрим процесс подстройки одного веса от нейрона p в скрытом слое j к нейрону q в выходном слое. Выход OUT слоя k вычитаясь из целевого значения Y^* , дает ошибку, которая умножается на производную сжимающей функции (в нашем случае $OUT(1-OUT)$), вычисленную для этого нейрона слоя k , давая, таким образом, величину

$$\delta = OUT(1 - OUT)(Y^* - OUT) \quad (3.7)$$

Затем δ умножается на величину OUT нейрона j , из которого выходит рассматриваемый вес. Это произведение в свою очередь умножается на коэффициент обучения η ($0,01 \leq \eta \leq 1$) и результат прибавляется к весу.

$$\Delta w_{pq,k} = \eta \delta_{q,k} \cdot OUT_{p,q}, \quad (3.8)$$

где $\delta_{q,k}$ - величина δ для нейрона q в выходном слое k ; $OUT_{p,q}$ - величина выхода для нейрона в скрытом слое j .

$$w_{pq,k}^{(n+1)} = w_{pq,k}^n + \Delta w_{pq,k}, \quad (3.9)$$

где $w_{pq,k}^n$ – величина веса от нейрона в скрытом слое k к нейрону q в выходном слое на шаге n , $w_{pq,k}^{(n+1)}$ – величина веса на шаге $n+1$ после коррекции. Такая же процедура выполняется для каждого веса от нейрона скрытого слоя к нейрону в выходном слое.

Рассмотрим один нейрон в скрытом слое, предшествующем выходному слою. При проходе вперед этот нейрон передает свой выходной сигнал нейронам в выходном слое через соединяющие их веса. Во время обучения эти веса функционируют в обратном порядке, пропуская величину δ от выходного слоя назад к скрытому слою. Каждый из этих весов умножается на величину δ нейрона, к которому он присоединен в выходном слое. Величина δ , необходимая для нейрона скрытого слоя, получается суммированием всех таких произведений и умножением на производную сжимающей функции:

$$\delta_{p,j} = OUT_{p,j} (1 - OUT_{p,j}) \sum_q \delta_{q,k} w_{pq,k} \quad (3.10)$$

Когда значение δ получено, веса между входным слоем и скрытым слоем j могут быть скорректированы с помощью формул (3.9) и (3.10), в которых индексы необходимо модифицировать в соответствии со слоем. То есть процесс обучения представляет собой вычисление δ для каждого нейрона в данном слое и коррекцию всех весов данного слоя.

Для ускорения обучения сети вводят нейронное смещение и импульс.

Введение нейронного смещения позволяет сдвигать начало отсчета передаточной функции и по сути является процедурой аналогичной подстройке порога персептронного нейрона. Смещение вводится посредством добавления к каждому слою нейронов дополнительного нейрона, на который подается сигнал, равный +1, а не выходу нейрона предыдущего слоя. В процессе обучения вес данного нейрона корректируется также как и остальные веса нейронов.

Введение импульса позволяет ускорить обучение сети при использовании алгоритма обратного распространения. Этот метод заключается в добавлении к коррекции веса члена, пропорционального величине предыдущего изменения веса. Как только происходит коррекция, она запоминается и служит для модификации всех последующих коррекций. Уравнение коррекции принимает следующий вид:

$$w_{pq,k}^{(n+1)} = \alpha \Delta w_{pq,k}^n + \eta (\delta_{q,k} OUT_{pj}) \quad (3.11)$$

Затем вычисляется изменение веса:

$$w_{pq,k}^{n+1} = w_{pq,k}^n + \Delta w_{pq,k}^{(n+1)} \quad (3.12)$$

где α - коэффициент импульса, обычно ≈ 0.9 . [5,6,8]

Описанный метод обучения МСП был применён в данной работе.

3.2. Применение МСП для определения концентрации

Для обработки ИСП спектров был использован трёхслойный МСП с 32 входами и одним выходом. Первый и второй слои включали по 16 нейронов. В качестве активационных функций первого слоя были выбраны линейные, второго и третьего слоёв — сигмоидальные функции (рис. 3.4).

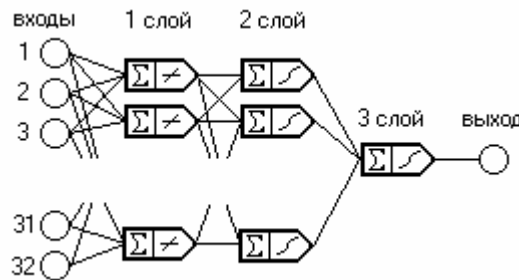


Рис. 3.4. МСП, использовавшийся для определения концентрации.

На входы сети подавались нормированные сканы, усредненные по частоте, чтобы свести длину скана от N отсчетов к 32 (обычно в качестве N выбиралась степень 2, или скан искусственно растягивался до ближайшего значения, кратного 32):

$$X_i = \frac{\sum_{k=n(i-1)}^{n \cdot i - 1} I_k}{n \cdot m_x}, \quad (3.6)$$

где X_i – i -й вход сети ($i=1, 2, \dots, 32$); I_k – k -тая выборка скана, m_I – среднее всех выборок скана, n – количество отсчётов скана, подаваемых на один вход сети. Сигнал на выходе МСП пропорционален натуральному логарифму концентрации.

Сеть обучалась в несколько циклов (достаточно 3-4). Каждый из них состоял в обучении МПС на 200 сканах. На рис. 3.5 представлены результаты такого использования ИНС. По осям абсцисс отложены настоящие концентрации, по осям ординат – распознанные. Пунктиром показана идеальная зависимость. Видно, что лучшие результаты получаются при использовании предварительной фильтрации сканов. В этом случае метод позволяет снизить предел обнаружения на порядок и уменьшить погрешность более чем в 2 раза. Это значительно превосходит результат обработки, полученный классическими методами в предыдущей главе.

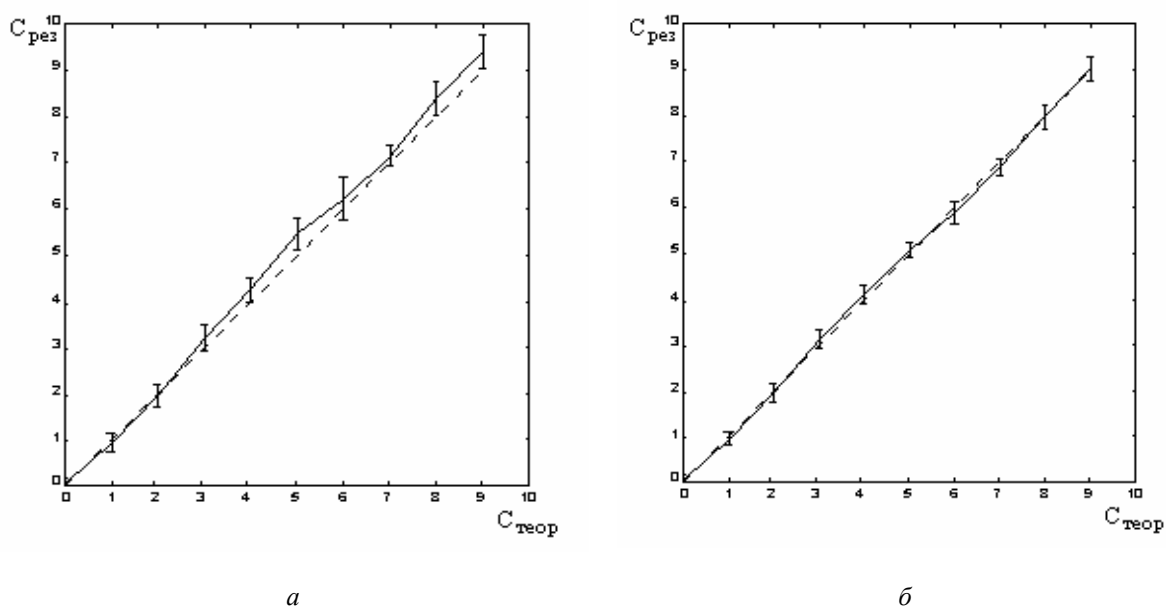


Рис 3.5. Результаты использования ИНС для определения концентрации:
а) без предварительной фильтрации; б) с предварительной фильтрацией.

Однако, применение рассмотренного метода на практике пока затруднено. Дело в том, что получение такого большого (600-800) объёма данных сопряжено с большими временными и материальными затратами.

Следует отметить, что применение нейронных сетей не ограничено рассмотренной архитектурой ИНС. Значительный практический интерес представляют алгоритмы аппроксимации экспериментальных данных с использованием ИНС.

ГЛАВА 4. АППРОКСИМАЦИЯ СПЕКТРОВ

Для облегчения работы с эмиссионной линией и изучения окружающих её линий фона, можно аппроксимировать участок спектра набором аналитических функций, описывающих физические эмиссионные пики. Это позволит избежать проблем, связанных с маскировкой пиков и упростить работе с аналитической линией. В этой главе будут рассмотрены два способа аппроксимации — аппроксимация методом наименьших квадратов (МНК) и с помощью ИНС с радиальными базисными функциями (РБФ-сети). Также будет синтезирован метод, сочетающий в себе МНК и РБФ-сети, и позволяющий добиться хороших результатов при многопараметрической аппроксимации.

4.1. Аппроксимация спектров методом наименьших квадратов

Для удобства рассмотрим абстрактный участок спектра, состоящий из трёх фоновых пиков и пика аналита. Пусть необходимо извлечь из этого спектра все 4 пика в явном виде, или, по крайней мере, пик анализируемого элемента.

4.1.1. Аппроксимация одним пиком. Будем считать, что пик аналита больше пиков фона. В этом случае можно попробовать выделить его путём аппроксимации всего скана функцией Гаусса (4.1), что и было проделано.

$$f(x, \vec{p}) = p_1 + p_2 \exp \left\{ - \left(\frac{x - p_3}{p_4} \right)^2 \right\} \quad (4.1)$$

Перед началом аппроксимации задавалось начальное приближение вектора параметров \vec{p} и допустимые границы его изменения. Для минимизации функции ошибок использовался метод локальных вариаций. Результат аппроксимации представлен на рис. 4.1а.

Функция (4.1) аппроксимировала линию лишь в её верхней части, т.е. из аппроксимации нельзя извлечь информацию ни об интенсивности пика ни об его базовом уровне. В случае же зашумлённости исходных данных аппроксимация вообще теряет всякий физический смысл (рис. 4.1б).

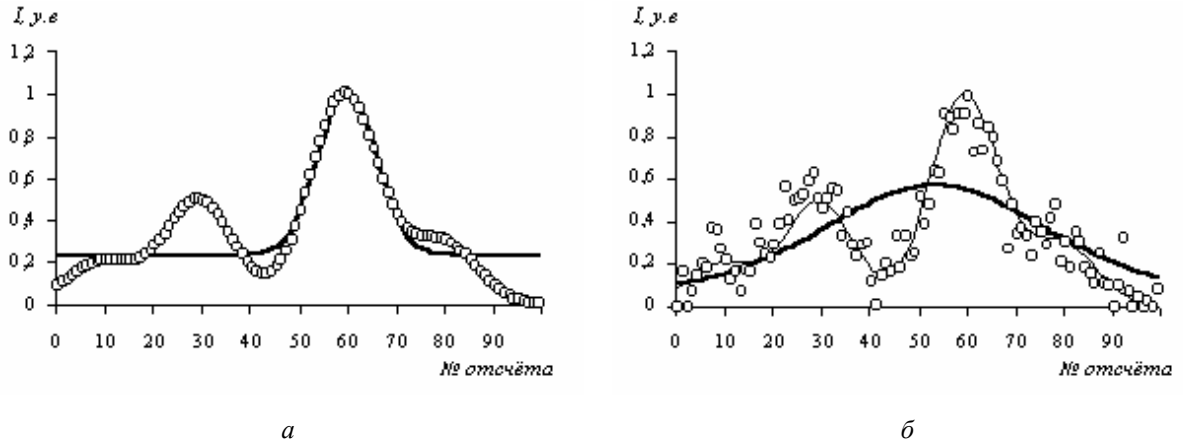


Рис. 4.1. Результат МНК-аппроксимации одним пиком;
 а) незашумлённые данные; б) данные с шумом.

4.1.2. Аппроксимация набором пиков. Для того, чтобы сделать аппроксимацию более точной приходится использовать несколько пиков. Тогда аппроксимирующая функция из N пиков принимает вид:

$$f(x, \vec{p}) = p_1 + \sum_{i=0}^{N-1} p_{3i+2} \exp \left\{ - \left(\frac{x - p_{3i+3}}{p_{3i+4}} \right)^2 \right\}. \quad (4.2)$$

Количество её параметров при этом становится равным $3N + 1$.

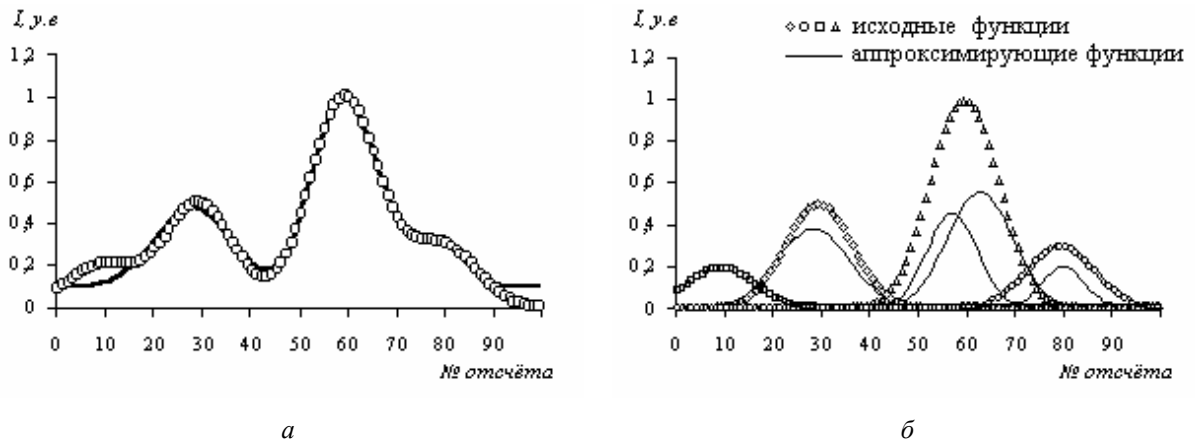


Рис. 4.2. Результат МНК-аппроксимации четырьмя пиками;
 а) общий вид; б) функциональный состав.

Была произведена аппроксимация абстрактного скана функцией при $N = 4$. Аппроксимация стала гораздо ближе к оригиналу (4.2а), чем в предыдущем случае. Однако пики, составляющие (4.2), по-прежнему далеки от исходных (4.2б).

Кроме того, этот метод, как и предыдущий, не устойчив к зашумлению данных, а при достаточно больших N он и вовсе не работоспособен. Это связано в первую

очередь с тем, что функция ошибок имеет некоторое количество локальных минимумов, в которые "проваливается" \vec{p} при аппроксимации.

Невозможность построения достаточно точной и физически оправданной аппроксимации эмиссионных спектров классическими методами заставляет искать новые подходы к этой проблеме. Одним из сравнительно новых подходов является аппроксимация функций РБФ-сетями.

4.2. Аппроксимация спектров РБФ-сетями

Исследования проблемы использования нейронных сетей для аппроксимации функций привели к созданию т.н. РБФ-сетей, которые были изначально разработаны для аппроксимации данных в многомерном пространстве. Впервые свойство универсальной аппроксимации РБФ-сетей было продемонстрировано Хартманом. Затем появились результаты Парка и Сандберга, исследовавших большой класс радиальных базисных функций.

Радиальные функции – это специальный класс функций, основной характеристикой которых является то, что их отклик монотонно изменяется в зависимости от расстояния до центральной точки. Сеть с радиальными базисными функциями считается нелинейной, если базисная функция может двигаться вдоль осей или изменять размер, или если сеть содержит более одного скрытого слоя.

4.2.1. РБФ-нейрон. На рис. 4.3 приведена структура нейрона, расположенного во входном слое РБФ-сети. Подающиеся на вход сети данные из внешней среды, поступают к нейрону вместе с вектором, определяющим центр базисной функции. От найденного модуля разности $\|P-C\|$, умноженного на смещение b , вычисляется передаточная радиальная базисная функция, сигнал на выходе которой и является выходом нейрона. [9]

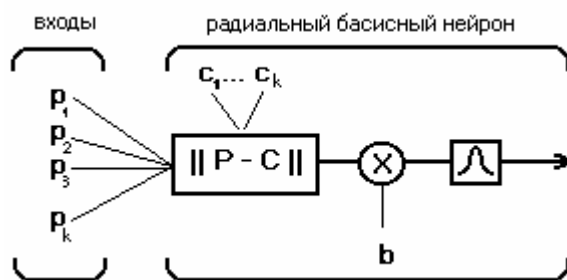


Рис. 4.3. Радиальный базисный нейрон.

С уменьшением расстояния между входным вектором и вектором, определяющим центр базисной функции, выход нейрона увеличивается. Таким образом, радиальный базисный нейрон работает как детектор, т.е. на его выходе будет максимальное значение когда вход P эквивалентен вектору C . Смещение b компенсирует различие в средних значениях от входа и выхода нейрона. [9]

4.2.2 Архитектура РБФ-сети. Представленная на рис. 4.4 РБФ-сеть состоит из двух слоев: скрытый слой радиальных базисных нейронов и выходной слой с классической передаточной функцией и классическими нейронами. Когда мы подаем входной вектор на такую сеть, каждый нейрон в радиальном базисном слое выдает на выходе величину соответствия входного вектора и вектора центров нейрона, которая близка к нулю в случае их различия и тогда не оказывает практически никакого влияния на выходной слой. Наиболее оптимальной для точной аппроксимации оказывается такая структура сети, где число нейронов скрытого слоя равно числу входных данных. В таком случае центры нейронов скрытого слоя практически совпадают со значениями входных векторов. Данная структура РБФ-сети является стандартной, то есть сети с более сложной структурой не рассматриваются. Некоторые изменения имеют место в случае постановки задачи получения гладкой аппроксимирующей функции:

1. число базисных функций будет меньше числа входов;
2. центры нейронов РБФ-слоя уже не будут настроены на конкретные входные вектора;
3. в общем случае, ширина каждой базисной функции должна определяться в процессе обучения.

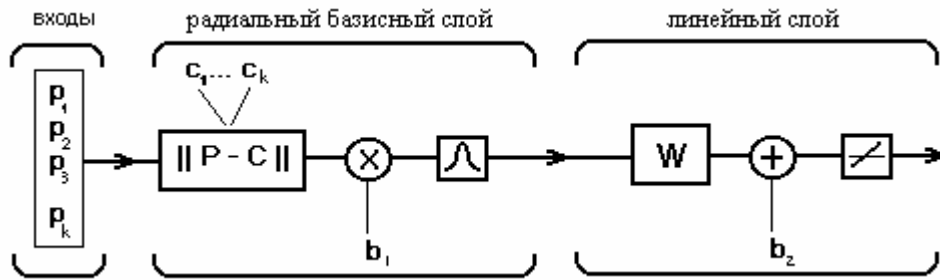


Рис.4.4. РБФ-сеть

Ключевой особенностью РБФ-сети является различие функций первого и второго слоя сети. Весовые коэффициенты первого слоя (т.е. параметры базисных функций) определяются в результате применения методов обучения без учителя. Это приводит к 2-х этапному процессу обучения. На первом этапе определяется смещение и положение центров базисных функций. На втором этапе определяются весовые коэффициенты следующего слоя с помощью стандартных методов с учителем. За счет применения методики обучения без учителя на первом этапе значительно ускоряется процесс обучения сети, по сравнению с сетями классической структуры. После обучения веса первого слоя указывают на положения радиальных функций, второго – на их интенсивности.

Формальное доказательство того, что линейная суперпозиция радиальных функций пригодна для универсальной аппроксимации, было получено Хартманом для сетей с гауссовской функцией, в которой ширина функций была настраиваемым параметром. Более общие результаты были получены Парком и Сандбергом. Однако они не давали процедур построения сетей. Затем было доказано, что РБФ-сети обладают свойством наилучшей аппроксимации (т.е. в наборе аппроксимирующих функций $\exists!$ функция, которая дает минимум ошибки аппроксимации для любой аппроксимируемой функции). Там же показано, что многослойные персептроны не обладают таким свойством. [9]

Также существенным отличием от классической сети является то, что в РБФ-сети для данного входного вектора только несколько нейронов дают существенные отклики, в то время как в персептроне для этого используется комбинация очень большого числа нейронов, что, в свою очередь, приводит к замедлению процесса обучения и к усложнению структуры сети.[9]

4.2.3. Наилучшая аппроксимация. Рассмотрим аппроксимацию участка эмиссионного спектра набором гауссовых функций с постоянной шириной. При увеличении числа входных нейронов точность аппроксимации улучшается. Однако значительное улучшение будет иметь место лишь до некоторого граничного числа нейронов. После этого рост числа нейронов будет приводить к незначительным улучшениям. Так, в случае аппроксимации линии свинца 220.353 нм (рис. 4.3) оказалось достаточно 9-ми нейронов.

На рис. 4.5б проиллюстрировано состояние базисных функций РБФ-сети после обучения. Сеть успешно справилась с аппроксимацией, однако имеют место пики с отрицательной амплитудой. Это не соответствует физике процесса. Видно, что такая аппроксимация эквивалентна фильтрации скана — она делает его гладким и максимально приближенным к незашумленному, но даёт достаточно спорную информации об отдельных пиках. Был проведен вычислительный эксперимент по аппроксимации РБФ-сетью произвольной суммы гауссов. Сеть великолепно раскладывала эту сумму в базис РБФ функций, но во-первых базис всегда был больше исходного числа гауссов, во-вторых всегда присутствовали РБФ с отрицательными амплитудами и, наконец, в-третьих все пики имели одинаковую ширину.

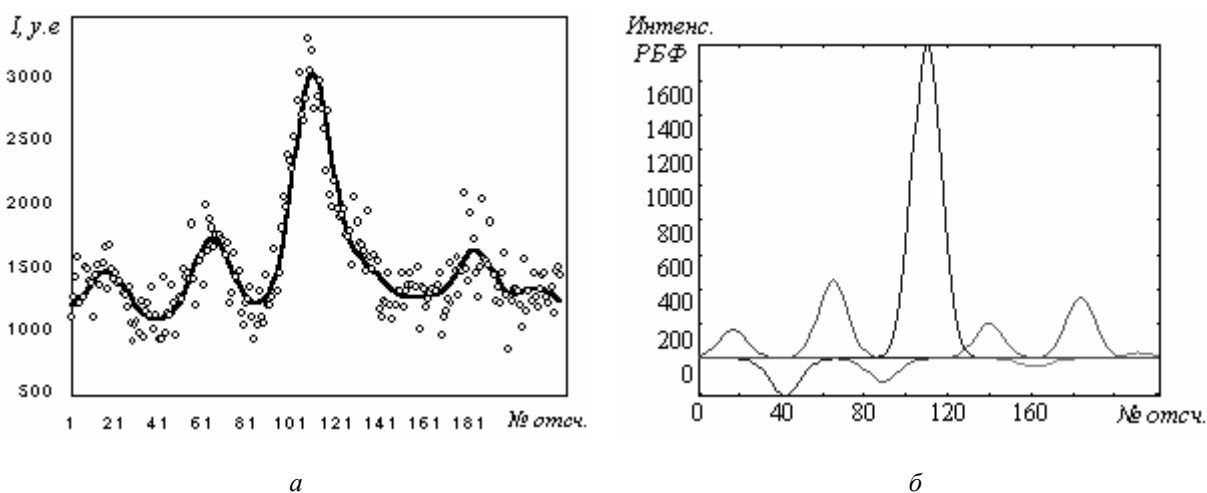


Рис.4.5. Аппроксимация участка спектра (свинец, 220.353 нм) РБФ-сетью;
 а) результат аппроксимации; б) вид радиальных базисных функций.

4.2.4. Аппроксимация заданным количеством пиков. Чтобы избежать появления РБФ с отрицательной амплитудой можно аппроксимировать скан небольшим числом функций. Возьмём 4 гауссовых функции, сдвинутых между собой и

построим их сумму. Пусть это будет исходный скан. Затем построим аппроксимацию РБФ, причём во входном слое пусть тоже будет тоже 4 нейрона. На рис. 4.5 представлен результат такой аппроксимации. Непрерывной линией показаны исходные данные, пунктиром — результат работы сети.

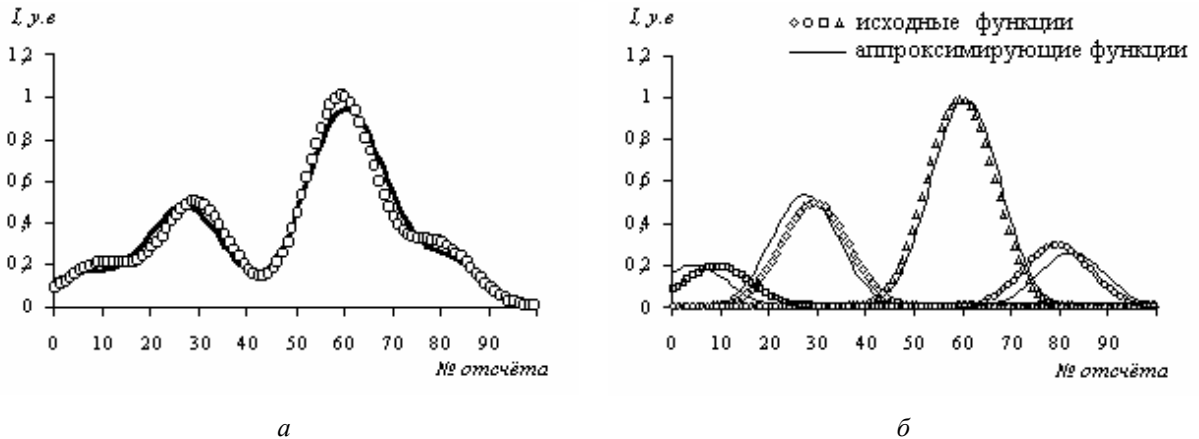


Рис. 4.6. Результат аппроксимации четырьмя радиальными функциями;
а) общий вид; б) функциональный состав.

Видно, что теперь радиальные функции с отрицательными амплитудами отсутствуют, зато и аппроксимация не вполне точная. Кроме того, ширина пиков остаётся постоянной.

4.3. Двухступенчатая аппроксимация

Для более точной и физически оправданной аппроксимации был предложен алгоритм, сочетающий в себе как МНК, так и аппроксимацию РБФ-сетями. Из рис. 4.4б видно, что аппроксимирующие РБФ и исходные пики весьма близки. Можно предположить, что если задать параметры радиальных функций в качестве начальных приближений при МНК, то процесс минимизации ошибки попадёт в глобальный минимум, а не в один из локальных. Для проверки этой гипотезы был поставлен ряд вычислительных экспериментов. Результаты двух из них представлены на рис. 4.7.

Метод великолепно справился как с аппроксимацией, так и с выявлением исходных пиков. Благодаря использованию РБФ-сетей, он в достаточной степени устойчив к шумам.

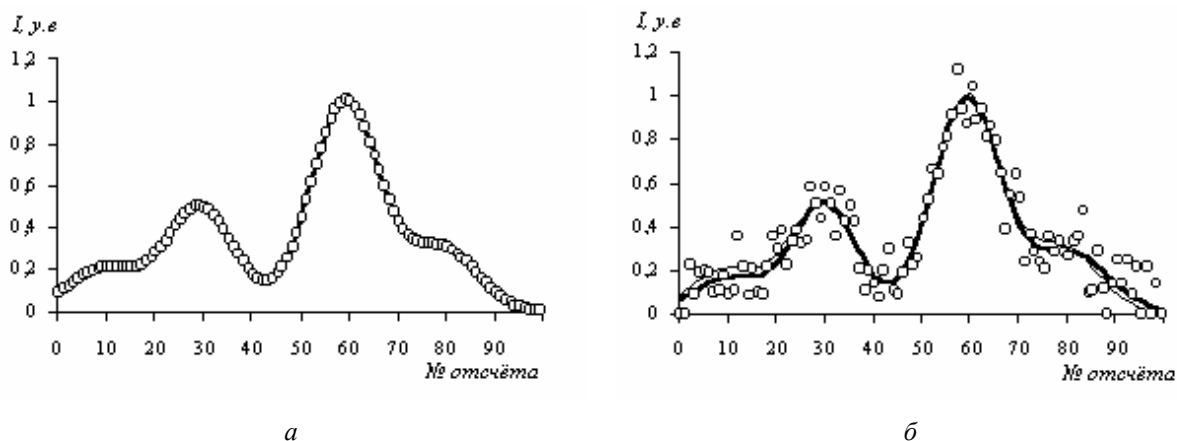


Рис. 4.7. Результат работы синтезированного метода;

а) аппроксимация незашумлённых данных; б) аппроксимация данных с шумом..

Синтезированный метод был применён к линии свинца (220.353 нм). Результаты аппроксимации представлены на рис. 4.8.

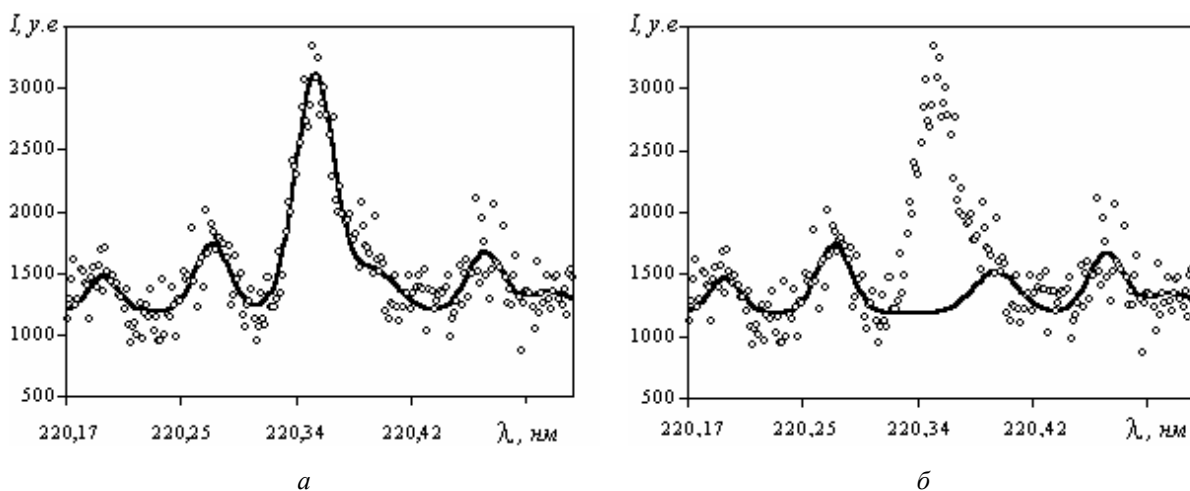


Рис. 4.8. Результаты работы синтезированного метода на экспериментальном скане;

а) аппроксимация участка спектра; б) выделенный фон плазмы.

Физически оправданная аппроксимация спектров дала возможность сравнительно легко устранять маскирующее влияние фоновых пиков, без длительного и досконального изучения спектра в окрестности пика. Снижение пределов обнаружения по свинцу в 2 раза было достигнуто с минимальными затратами времени.

ЗАКЛЮЧЕНИЕ

Итогом проведённых исследований явилась программа обработки эмиссионных спектров, реализованная в среде разработки DELPHI, а так же библиотека процедур среды MATLAB. Эти программные средства используются при коррекции эмиссионных спектров в лаборатории лазерной диагностики плазмы ИМАФ НАН РБ. Созданная программа позволяет:

- снизить предел обнаружения анализируемых элементов в среднем в два раза;
- уменьшить относительное с.к.о. при определении концентрации в среднем в 1.8 раз по интенсивности и в 1.4 раза по площади.

Следует отметить, что данное снижение пределов обнаружения достигается сравнительно дешево, без переделки материальной части прибора. Значительно дороже стоит модернизация ИСП-спектрометра, направленная на стабилизацию подачи газа, подачи раствора пробы, мощности ВЧ-генератора, на увеличение точности настройки монохроматора, на уменьшение шумов электронных устройств. Модернизация же ИСП-спектрометра совместно с применением современных методов обработки данных позволит ощутимо снизить пределы обнаружения элементов и приблизить их к показателям ИСП-спектрометров лучших зарубежных фирм.

Проведённое в работе исследование применимости нейросетевых методов обработки спектров показало их методические преимущества перед классическими методами. Однако практическое применение ИНС для прямого определения концентраций элементов на настоящем этапе затруднительно и требует дальнейших исследований.

Часть работы была выполнена в рамках гранта БГУ для студентов и аспирантов "Разработка методов повышения чувствительности приёмников оптического, ультрафиолетового и инфракрасного излучения", № 610/18.

Результаты работы докладывались на 2-х конференциях:

1. 56-я конференция студентов и аспирантов БГУ (1999 г.).
2. V Республиканская научная конференция студентов и аспирантов РБ (НИРС - 2000).

В печати находятся 2 статьи:

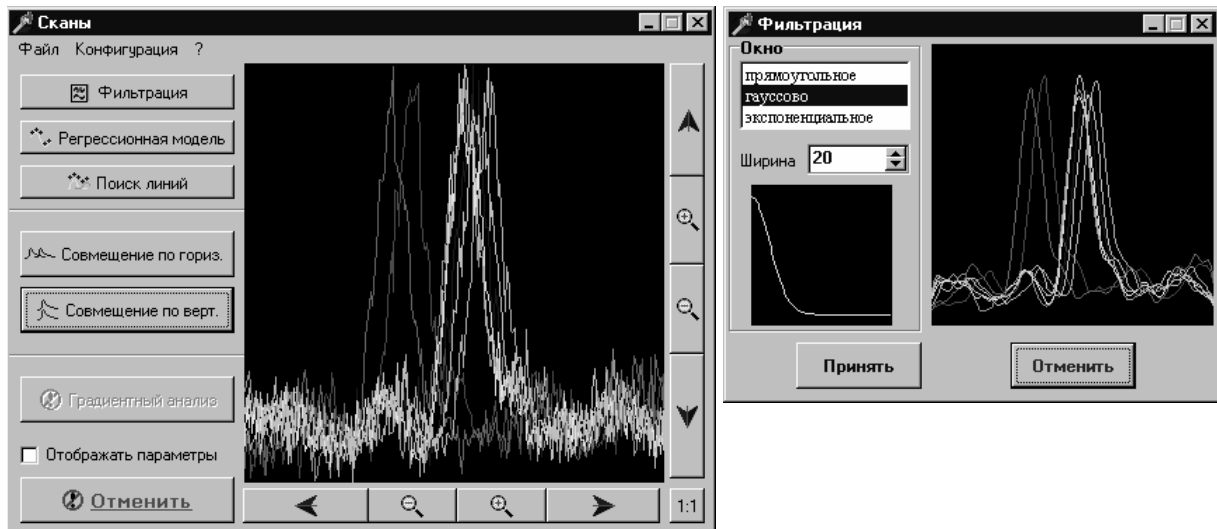
1. "Применение нейронных сетей для обработки атомно-эмиссионных спектров", — Сборник работ факультета РФЭ за 1999.
2. "Снижение погрешности определения элементов в атомно-эмиссионном спектральном анализе", — Вести НАН РБ.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. “Атомно-эмиссионный анализ с индуктивной плазмой” под ред. П.В. Нестерова. “Итоги науки и техники”, серия “Аналитическая химия”, том 2. М.:ВИНИТИ, 1990, с. 4-10
2. Исаевич А. В., Козловский А. С., Мисаков П. Я. Исследование аналитических и метрологических характеристик атомно-эмиссионного спектрометра на основе индукционно-связной плазмы. Препринт №6. — ИМАФ НАН РБ, 1997, — 23 с.
3. Фирш С. Э. и др. Спектроскопия газоразрядной плазмы. Под ред. Фриша С. Э. — Л.: Наука, 1970, — с. 7–16.
4. Рабинер Л., Гоулд. Б. Теория и применения цифровой обработки сигналов. — М.: Мир, 1978, — 848 с.
5. Уоссерман Ф. Нейрокомпьютерная техника: теория и практика. — М.: Мир, 1992.
6. Змитрович А.И. Интеллектуальные информационные системы. — Мн.: НТООО "ТетраСистемс", 1997.
7. Минский М.Л., Пайперт С. Перцептроны. — М.: Мир, 1971.
8. Амосов Н.М. и др. Нейрокомпьютеры и интеллектуальные роботы. Под ред. Амосова Н. М. — Киев: Наук. думка, 1991.
9. Bishop M. Neural Networks for Pattern Recognition, — Oxford: Clarendon Press, 1997.

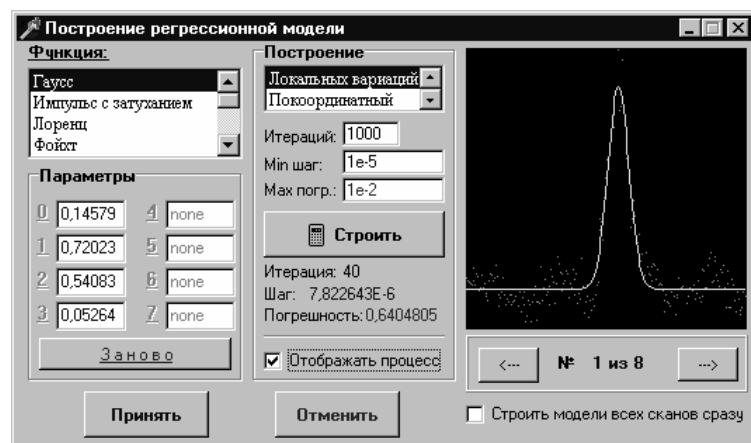
ПРИЛОЖЕНИЕ 1. ПРОГРАММА ОБРАБОТКИ АЭМ СПЕКТРОВ

По результатам работы было создано приложение, позволяющее обрабатывать АЭМ спектры. Из рассмотренных алгоритмов в него вошли: фильтрация, корреляционный сдвиг, коррекция фона и аппроксимация. Программа позволяет работать данными как в специальном формате, так и в формате ASCII.



а

б



в

Рис. 1. Внешний вид разработанного приложения;
а) главное окно, б) окно фильтрации; в) окно аппроксимации.

Рассмотрим основные режимы работы приложения.

Режим фильтрации (рис. 1б) позволяет производить фильтрацию сканов прямоугольным, гауссовым и экспоненциальным окнами с варьируемыми коэффициентами.

Режим аппроксимации (рис. 1в) позволяет аппроксимировать сканы различными аналитическими функциями. Эти функции находятся в DDL библиотеках и могут быть дополнены пользователем по желанию. Единственным ограничением является количество параметров функции. В программе не предусмотрена работа с функциями, имеющими более 8 параметров.

Все параметры сканов вычисляются автоматически при каждом действии, и их результат отображается в отдельном окне.

Листинг 1. Модуль обработки сигналов.

```
unit SignalProcessing;
interface
uses SysUtils;

Type TSingleArray=array[0..0] of single;
      PSingleArray=^TSingleArray;
      TWordArray=array[0..0] of Word;
      PWordArray=^TWordArray;

procedure FFT (var sre,sim:PSingleArray; n:longword; code:byte );
procedure Spectrum(var X,S:PSingleArray; n:longword; CENT,NORM,LOG:byte);
procedure FFTFilter(var X,W:PSingleArray; n:longword);
procedure Correlat(var X:PSingleArray; n:longword);
function RandN(M,STD:double):double;
function Rand(X1,X2:double):double;
function GetMin(var Data:PSingleArray;n:longword):single;
function GetMax(var Data:PSingleArray;n:longword):single;
procedure SaveAsText(FName:string; var
Data:PSingleArray;n:longword;atr:byte);

implementation

Procedure FFT (var sre,sim:PSingleArray; n:longword; code:byte );
var
  re,sr,si,im,arg:real;
  jp,jk,jq,ih,ix,n2:integer;
  i,ml:integer;
  wre,wim:PSingleArray;
label label_1;
Begin
  GetMem(Wre,n*SizeOf(TSingleArray));
  GetMem(Wim,n*SizeOf(TSingleArray));
  { Calculation W-ik for FFT }
  n2:=n shr 1;
  arg:=Pi/n2;
  for i:=0 to n2-1 do
  begin
    re:=arg*i;
```

```
        wre^[i]:=cos(re);
        wim^[i]:=-sin(re);
    end;
    ih:=n2;
    ix:=1;
    label_1 :i:=0;
    while i<n2 do
    begin
        ml:=i mod ih;
        jk:=ml*ix;
        jp:=(i shl 1)-ml;
        jq:=jp+ih;
        re:=sre^[jp];
        sre^[jp]:=re+sre^[jq];
        re:=re-sre^[jq]; im:=sim^[jp];
        sim^[jp]:=im+sim^[jq];
        im:=im-sim^[jq];
        sre^[jq]:=re*wre^[jk]-im*wim^[jk];
        sim^[jq]:=re*wim^[jk]+im*wre^[jk];
        i:=i+1;
    end;
    ih:=ih shr 1;
    ix:=ix shl 1;
    if ih>0 then goto label_1;
        {Building double inversion}
    jq:=0;
    for jp:=1 to n-2 do
    begin
        ml:=n2;
        while jq >= ml do
        begin
            jq:=jq-ml;
            ml:=ml shr 1
        end;
        jq:=jq+ml;
        if jq>jp then
        begin
            sr:=sre^[jp];
            si:=sim^[jp];
            sre^[jp]:=sre^[jq];
            sim^[jp]:=sim^[jq];
            sre^[jq]:=sr;
            sim^[jq]:=si
        end
    end;
    if code=1 then
    begin
        for i:=0 to n-1 do
        begin
            Sre^[i]:=Sre^[i]/n;
            Sim^[i]:=Sim^[i]/n
        end
    end;
    FreeMem(Wre,n*SizeOf(TSingleArray));
    FreeMem(Wim,n*SizeOf(TSingleArray));
end;

procedure Spectrum(var X,S:PSingleArray; n:longword; CENT,NORM,LOG:byte);
var Sre,Sim:PSingleArray;
    i:longword;
    m:double;
Begin
```

```
i:=1;
while i<n do
begin
  if n mod i <> 0 then halt;
  i:=i*2;
end;
GetMem(Sre,N*SizeOf(single));
GetMem(Sim,N*SizeOf(single));
fillchar(Sim^,N*SizeOf(single),0);
m:=0;
if CENT>0 then
  for i:=0 to N-1 do m:=m+X^[i];
m:=m/N;
for i:=0 to N-1 do Sre^[i]:=X^[i]-m;
fft(Sre,Sim,N,NORM);
for i:=0 to (N div 2)-1 do
  case LOG of
    0:S^[i]:=sqr(Sre^[i])+sqr(Sim^[i]);
    1:begin
      S^[i]:=sqr(Sre^[i])+sqr(Sim^[i]);
      if S^[i]<1e-15 then S^[i]:=1e-15;
      S^[i]:=ln(S^[i]);
    end;
    2:S^[i]:=ln(sqr(Sre^[i])+sqr(Sim^[i]))/ln(10);
  end;
FreeMem(Sre,N*SizeOf(single));
FreeMem(Sim,N*SizeOf(single));
End;
```

```
procedure FFTFilter(var X,W:PSingleArray; n:longword);
var Sre,Sim:PSingleArray;
    i:longword;
Begin
  i:=1;
  while i<n do
  begin
    if n mod i <> 0 then halt;
    i:=i*2;
  end;
  GetMem(Sre,n*SizeOf(single));
  GetMem(Sim,n*SizeOf(single));
  fillchar(Sim^,n*SizeOf(single),0);
  for i:=0 to n-1 do Sre^[i]:=X^[i];
  fft(Sre,Sim,n,0);
  for i:=0 to (n div 2)-1 do
  Begin
    Sre^[i]:=Sre^[i]*W^[i];
    Sim^[i]:=-Sim^[i]*W^[i];
    Sre^[n div 2 + i]:=Sre^[n div 2 + i]*W^[n div 2 - 1 - i];
    Sim^[n div 2 + i]:=-Sim^[n div 2 + i]*W^[n div 2 - 1 - i];
  End;
  fft(Sre,Sim,n,0);
  for i:=0 to n-1 do
    X^[i]:=Sre^[i]/n;
  FreeMem(Sre,n*SizeOf(single));
  FreeMem(Sim,n*SizeOf(single));
End;
```

```
Procedure Correlat(var X:PSingleArray; n:longword);
Var il,Tst:integer;
    vl:real;
    Sre,Sim:PSingleArray;
```

```
    i:longword;
  Begin
    GetMem(Sre,SizeOf(TSingleArray)*2*n);
    GetMem(Sim,SizeOf(TSingleArray)*2*n);
    for i1:=0 to Nc-1 do
      begin
        Sre^[i1]:=Data^[i1]/MaxVolume;{-M1}
        Sim^[i1]:=0;
      end;
    for i:=Nc to 2*Nc-1 do
      begin
        Sre^[i]:=0;
        Sim^[i]:=0
      end;
    FFT(Sre,Sim,2*n,1);
    for i:=0 to 2*n-1 do
      begin
        Sre^[i]:=Sre^[i]*Sre^[i]+Sim^[i]*Sim^[i];
        Sim^[i]:=0
      end;
    FFT(Sre,Sim,2*n,0);
    v1:=Sre^[0];
    for i:=0 to n-1 do
      begin
        C^[i]:=Sre^[i]/v1;
      end;
    FreeMem(Sre,SizeOf(TSingleArray)*2*n);
    FreeMem(Sim,SizeOf(TSingleArray)*2*n);
  End;

function RandN(M,STD:double):double;
var
  x:real;
  v1,v2:real;
Begin
  repeat
    v1:=2*random-1;
    v2:=2*random-1;
    x:=v1*v1+v2*v2;
  until x<1;
  Result:=STD*v1*sqrt(-2*ln(x)/x)+M;
End;

function Rand(X1,X2:double):double;
Begin
  Result:=X1+(X2-X1)*random;
End;

function GetMin(var Data:PSingleArray;n:longword):single;
var i:longword;
Begin
  Result:=Data^[0];
  for i:=1 to n-1 do
    if Result>Data^[i] then Result:=Data^[i];
  End;

function GetMax(var Data:PSingleArray;n:longword):single;
var i:longword;
Begin
  Result:=Data^[0];
  for i:=1 to n-1 do
    if Result<Data^[i] then Result:=Data^[i];
```

```
End;

Procedure SaveAsText (FName:string; var
Data:PSingleArray;n:longword;atr:byte);
var T:Text;
    i:integer;
Begin
    system.Assign(T,FName);
    if atr=0 then rewrite(T)
        else append(T);
    for i:=0 to n-1 do
        writeln(T,FloatToStr(Data^[i]),'  ',i);
    writeln(T);
    system.Close(T);
End;
end.
```

Листинг 2. Модуль построения аппроксимации.

```
unit Functions;
INTERFACE
USES SignalProcessing,Global,
    StdCtrls,ExtCtrls,Windows,SysUtils,Dialogs;

CONST MaxError:real=1e-2;           // Допустимая ошибка регрессии
        MinStep:real=1e-5;
        MaxItr:longint=1000;        // Максимальное число итераций
        nParam:byte=5;

TYPE TParameters=array[0..7] of double;
        TBorder=array[0..7,0..1] of double;

VAR P0:TParameters;
        P:array[0..255] of TParameters;
        Border:TBorder;

//Описание функций
TYPE
    TF=function (x:double; P:TParameters):double;stdcall;
    TdF=function (n:byte; x:double; P:TParameters):double;stdcall;
    TGet_F_nParams = function :byte; stdcall;
    TGetParamBorder = function (nP:byte;Bkind:byte):double; stdcall;
    TGetParam0 = function (nP:byte):double; stdcall;

VAR
    FLibHandle:HModule;
    F:TF;
    dF:TdF;
    Get_F_nParams:TGet_F_nParams;
    GetParamBorder:TGetParamBorder;
    GetParam0:TGetParam0;

function E(X,Y:PSingleArray; P:TParameters):double;
function dE(n:byte; X,Y:PSingleArray; P:TParameters):double;
Function GetMin_LocVar(X,Y:PSingleArray; var P:TParameters;
    Max_Itr:longword; var a:double):double;
Function GetMin_Grad(X,Y:PSingleArray; var P:TParameters;
    Max_Itr:longword; var a:double):double;
Function GetMin_MonteKarlo(X,Y:PSingleArray; var P:TParameters;
```

```
Max_ITR:longword; var a:double):double;  
procedure ProceedDLLOpen(FileName:string);  
procedure GetNewFunctionDLL(Name:PChar);
```

IMPLEMENTATION

```
function E(X,Y:PSingleArray; P:TParameters):double;  
Var i:smallint;  
Begin  
  Result:=0;  
  for i:=0 to Scans.nSamples-1 do Result:=Result+sqr(Y^[i]-F(X^[i],P));  
End;
```

```
function dE(n:byte; X,Y:PSingleArray; P:TParameters):double;  
Var i:smallint;  
Begin  
  Result:=0;  
  for i:=0 to Scans.nSamples-1 do  
    Result:=Result+(F(X^[i],P)-Y^[i])*dF(n,X^[i],P);  
  Result:=Result*2;  
End;
```

```
Function GetMin_LocVar(X,Y:PSingleArray; var P:TParameters;  
Max_ITR:longword; var a:double):double;
```

```
Var  
  err,err_min:double;  
  i,j:smallint;  
  Itr:word;  
  Jmin:array [0..7] of smallint;  
  Steps,Pnew:TParameters;  
Begin  
  Itr:=0;  
  Err:=0;  
  for i:=0 to nParam-1 do  
    Begin  
      Steps[i]:=Border[i,1]-Border[i,0];  
      Pnew[i]:=P[i];  
    End;  
  
  while (Itr<=Max_ITR) do  
    Begin  
      inc(Itr);  
      // Проход по координатам  
      for i:=0 to nParam-1 do  
        Begin  
          err_min:=1e10;  
          for j:=-1 to 1 do  
            Begin  
              Pnew[i]:=P[i]+j*a*Steps[i];  
              if ((Pnew[i]>Border[i,0])and(Pnew[i]<Border[i,1])) then  
                begin  
                  err:=E(X,Y,Pnew);  
                  if (err<err_min) then  
                    begin  
                      err_min:=err;  
                      Jmin[i]:=j;  
                    end;  
                end;  
              Pnew[i]:=P[i];  
            End;  
          P[i]:=P[i]+Jmin[i]*a*Steps[i];  
          Pnew[i]:=P[i];  
        End;  
    End;
```



```
End;
// Проверка необходимости изменения шага
j:=1;
for i:=0 to nParam-1 do
  if (Jmin[i]<>0) then
    begin
      j:=0;
      break;
    end;
  if j=1 then a:=a/1.5;
  err:=E(X,Y,P);
End;
Result:=err;
End;
```

```
Function GetMin_Grad(X,Y:PSingleArray; var P:TParameters;
                      Max_ITR:longword; var a:double):double;
```

```
Var i:word;
    P0:TParameters;
    err,dP:real;
Begin
  P0:=P;
  Result:=E(X,Y,P);
  for i:=0 to nParam-1 do
    begin
      dP:=0.001*a*dE(i,X,Y,P);
      if (P[i]-dP<Border[i,0])or(P[i]-dP>Border[i,1]) then
        a:=a/1.05
      else P[i]:=P[i]-dP;
    end;
  err:=E(X,Y,P);
  if err<=Result then Result:=err
    else begin
      P:=P0;
      a:=a/1.05;
    end;
End;
```

```
Function GetMin_MonteKarlo(X,Y:PSingleArray; var P:TParameters;
                             Max_ITR:longword; var a:double):double;
```

```
Var i:word;
    P0:TParameters;
    err,dP:real;
Begin
  P0:=P;
  Result:=E(X,Y,P);
  for i:=0 to nParam-1 do
    begin
      repeat
        dP:=a*(1-2*random)*(Border[i,1]-Border[i,0]);
      until (P[i]+dP>Border[i,0])and(P[i]+dP<Border[i,1]);
      P[i]:=P[i]+dP;
    end;
  err:=E(X,Y,P);
  if err<Result then Result:=err
    else begin
      P:=P0;
      a:=a/1.01;
    end;
End;
```

```
Procedure ProceedDLLOpen(FileName:string);
```

```
Var Nm:array[0..31] of char;
    PNm:PChar;
    i:word;
Begin
    for i:=1 to length(FileName) do
        Nm[i-1]:=FileName[i];
    Nm[length(FileName)]:=#0;
    PNm:=Nm;
    if @F<>nil then FreeLibrary(FLibHandle);
    GetNewFunctionDLL(PNm);
End;

Procedure GetNewFunctionDLL(Name:PChar);
Var i:byte;
Begin
    FLibHandle:=LoadLibrary(Name);
    if FLibHandle<32 then
        begin
            ShowMessage('Could not load DLL. ');
            halt;
        end;
    @F:=GetProcAddress(FLibHandle,'F');
    @dF:=GetProcAddress(FLibHandle,'dF');
    @Get_F_nParams:=GetProcAddress(FLibHandle,'Get_F_nParams');
    @GetParamBorder:=GetProcAddress(FLibHandle,'GetParamBorder');
    @GetParam0:=GetProcAddress(FLibHandle,'GetParam0');
    if (@F=nil)or(@dF=nil) then
        begin
            FreeLibrary(FLibHandle);
            ShowMessage('Could not link the procedure in DLL. ');
            halt;
        end;
    nParam:=Get_F_nParams;
    for i:=0 to 7 do P0[i]:=0;
    for i:=0 to nParam-1 do
        begin
            Border[i,0]:=GetParamBorder(i,0);
            Border[i,1]:=GetParamBorder(i,1);
            P0[i]:=GetParam0(i);
        end;
End;

end.
```

Листинг 3. DLL, отвечающая за функцию Гаусса при аппроксимации.

```
library Gaussian;
uses
    ShareMem,
    Types in 'Types.pas';
Const
    NUMBER_OF_PARAMETERS=4;
    P0:TParameters = (0.5, 0.5, 0.5, 0.15, 0, 0, 0, 0);
    Border:TBorder=
        ((0 ,1),
         (0 ,1),
         (0 ,1),
         (1e-4,0.5),
         (0 ,0),
```

```
(0 ,0),
(0 ,0),
(0 ,0));

function seed_function(x:double; P:TParameters):double;
Begin
  Result:=exp(-sqr((x-P[2])/P[3])) ;
End;
// Функция
function F(x:double; P:TParameters):double; stdcall;
Begin
  Result:=0;
  if ((P[1]<0)or(P[2]<0)or(P[3]<0)) then exit;
  Result:=P[0]+P[1]*seed_function(x,P);
End;
// Частные производные по параметру n
function dF(n:byte; x:double; P:TParameters):double; stdcall;
Begin
  case n of
    0 : Result:= 1;
    1 : Result:=seed_function(x,P);
    2 : Result:=-2*P[1]*(x-P[2])*seed_function(x,P)/(P[3]*P[3]);
    3 : Result:=2*P[1]*(x-P[2])*seed_function(x,P)/(P[3]*P[3]*P[3]);
    else Result:=0;
  end;
End;
//Возвращает количество параметров
function Get_F_nParams:byte; stdcall;
Begin
  Result:=NUMBER_OF_PARAMETERS;
End;
//Возвращает границы параметров (по одной)
function GetParamBorder(nP:byte;Bkind:byte):double; stdcall;
Begin
  Result:=Border[nP,Bkind];
End;
function GetParam0(nP:byte):double; stdcall;
Begin
  Result:=P0[nP];
End;

exports
  F,
  dF,
  Get_F_nParams,
  GetParamBorder,
  GetParam0;

begin
end.
```