

ГОСУДАРСТВЕННЫЙ КОМИТЕТ СВЯЗИ, ИНФОРМАТИЗАЦИИ И  
ТЕЛЕКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ РЕСПУБЛИКИ УЗБЕКИСТАН  
ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

*На правах рукописи*

УДК 004.421

**СЫДЫКОВ ХУСАН БАХТИЯРОВИЧ**

**Разработка алгоритмов и прикладных программ низкочастотной  
фильтрации биосигналов на основе спектрального подхода**

5A330203 – Прикладная информатика

Диссертация на соискание академической степени магистра

Научный руководитель  
к.т.н., доц. Каххаров А.А.

Ташкент – 2013

## ОГЛАВЛЕНИЕ

<b>Введение.....</b>	<b>3</b>
<b>Глава 1. Анализ компьютерных систем регистрации биосигналов....</b>	<b>7</b>
1. Системы регистрации и математической обработки биосигналов.....	7
2. Проектирование компьютерных биомедицинских систем реального времени.....	20
3. Компьютерные системы регистрации электрокардиограмм.....	24
4. Анализ погрешностей каналов измерения.....	37
<b>Выводы по главе I.....</b>	<b>42</b>
<b>Глава 2. Методы и алгоритмы фильтрации сигналов ЭКГ.....</b>	<b>44</b>
1. Характеристики помех при регистрации электрокардиограмм и методы их устранения.....	44
2. Методы идентификации с использованием калибровочных и синтезируемых сигналов.....	46
3. Фильтрация сигналов ЭКГ на основе спектральных функций.....	55
<b>Выводы по главе II.....</b>	<b>64</b>
<b>Глава 3. Программная реализация алгоритма фильтрации ЭКГ на основе спектральных функций.....</b>	<b>65</b>
1. Программы фильтрации сигналов ЭКГ на основе спектрального подхода.....	65
2. Порядок функционирования программ.....	67
3. Инструкции для пользователей программ.....	72
<b>Выводы по главе III.....</b>	<b>76</b>
<b>Заключение.....</b>	<b>77</b>
<b>Список литературы.....</b>	<b>78</b>
<b>Приложение.....</b>	<b>82</b>

## Введение

Сбор, обработка и автоматизированный анализ физиологической информации человека являются важнейшей составной частью многих диагностических методов современной медицины. Компьютерные системы сбора и математической обработки электрофизиологических сигналов являются сложными аппаратно-программными комплексами, состоящими из множества программных компонент, выполняющих функции регистрации биомедицинской информации, ее анализа и обработки, а также диагностики.

Роль электрофизиологических сигналов в современной диагностике трудно переоценить. Однако имеющиеся в клинике аппаратные комплексы записи и анализа электрофизиологических сигналов обладают рядом погрешностей различного происхождения. Это связано с тем, что электрокардиографы, электроэнцефалографы, эхокардиографы, ангиографические комплексы, томографическая аппаратура работают с малыми величинами токов и напряжений на уровне милливольт и микроампер. В этом случае несовершенство аппаратуры и наличие помех приводит к тому, что контроль одного и того же физиологического процесса, проводимый различными общепризнанными приборами, дает различные значения контролируемых параметров.

Качество оценки состояния больного зависит от того, насколько эти погрешности будут скомпенсированы. Учитывая, что автоматические способы мониторинга состояния больного постоянно развиваются, очевидно, что точность записи и обработки биологического сигнала приобретает принципиальное значение.

Для повышения качества диагностики заболеваний сердца на ранней стадии необходимо, в частности, увеличивать точность измерения электрокардиосигнала. Для ряда кардиологических обследований недостаточно интерпретации высоковольтных зубцов ЭКГ (P,Q,R,S,T). Получить информацию о низковольтных зубцах (например, о расщепленном зубце P) на

стандартных кардиографах невозможно. Широко распространенные в настоящее время компьютерные регистраторы ЭКГ позволяют достаточно просто усилить исходный сигнал (в 10÷100 раз по отношению к обычному). При этом особенно важной оказывается проблема подавления помех близких к 50 Гц.

Преобладающая помеха, как правило, имеет частоту от 40 до 55 Гц, при этом может изменяться как по частоте, так и по амплитуде и фазе даже в течение нескольких секунд на одном отведении. Для сигналов, амплитудные и частотные параметры которых существенно отличаются от основного сигнала, наиболее эффективным способом подавления помех является низкочастотная фильтрация. Такой фильтр реализуется не аппаратно, а программным способом, что чрезвычайно важно, поскольку позволяет не усложнять средства регистрации, а использовать широчайшие возможности компьютера, входящего, как правило, в состав компьютерных комплексов.

В последнее время ведутся активные работы по разработке более точных и быстрых алгоритмов классификации электрокардиограмм. Широко используется спектральный анализ электрокардиограмм с последующим табличным преобразованием для нахождения QRS-комплексов. Также другие работы используют частотно-временной анализ для выявления QRS-комплексов, а также дополнительной информации об относительном смещении QRS-комплексов в отведениях.

**Актуальность данной работы.** Таким образом, проблема разработки алгоритмов фильтрации электрокардиосигналов для точного распознавания элементов электрокардиограмм (ЭКГ) весьма актуальна, особенно в связи с необходимостью постановки точного и быстрого диагноза.

В соответствии с вышеизложенным актуальность представленной проблемы не вызывает сомнений.

**Целью диссертационной работы** является анализ архитектуры компьютерных систем регистрации и обработки ЭКГ, используемых методов и средств обработки, разработка метода и алгоритма фильтрации медико-

биологических сигналов на примере ЭКГ на основе спектрального анализа. При этом решаются следующие задачи.

1. Исследование аппаратно-программных средств регистрации кардиосигналов и анализ основных проблем по повышению качества обработки и постановке диагноза.
2. Исследование методов и алгоритмов программной фильтрации сигналов ЭКГ, разработка алгоритма низкочастотной фильтрации на основе спектральных методов.
3. Разработка и реализация прикладных программ обработки для фильтрации сигналов ЭКГ в системе 1-ого стандартного отведения с поддержкой измерения и анализа одноканальной ЭКГ на основе разработанного алгоритма.

**Научная новизна** диссертационной работы заключается в следующем:

- исследованы современные средства обработки кардиографической информации, выявлены основные проблемы повышения качества обработки, исследованы методы и алгоритмы цифровой фильтрации ЭКГ (программная реализация);
- разработан метод и соответствующий алгоритм низкочастотной фильтрации, отличающийся высоким быстродействием (при сохранении заданной точности) и относительной простотой программной реализации.

**Объект и предмет исследования.** Объектом исследования в данной диссертационной работе являются сигналы регистрации кардиограмм в медицинских исследовательских комплексах. Предметами являются устройство обработки и демонстрации ЭКГ сигналов.

**Методы исследования.** Теоретическую основу проведенных исследований составляют теория функционального анализа, методы спектрального анализа в Фурье-базисах, численные методы, а также теория рядов и матриц.

**Практическая значимость** – алгоритм и программа фильтрации имеют большую диагностическую значимость при расшифровке сигналов ЭКГ с целью постановки диагноза, поскольку устраняется зашумленность

характерных участков ЭКГ. Разработанные прикладные программы и соответствующие инструкции пользователя могут быть использованы в медицинских учреждениях кардиологического профиля.

**Объем и структура работы.** Магистерская диссертация состоит из введения, четырёх глав и заключения, изложена на 82 страницах машинописного текста, содержит 9 таблиц и 24 рисунка. Список литературы содержит 40 публикаций.

# ГЛАВА 1. АНАЛИЗ КОМПЬЮТЕРНЫХ СИСТЕМ РЕГИСТРАЦИИ БИОСИГНАЛОВ

## 1. Системы регистрации и математической обработки биосигналов

В настоящее время одним из наиболее важных инструментов при проведении исследований по изучению биологических объектов и особенностей организма человека служат компьютерные биомедицинские системы (БМС) регистрации и обработки электрофизиологических сигналов, обеспечивающие функции записи, обработки и анализа биомедицинских сигналов различной природы.

Биологические сигналы допускают несколько возможных классификаций, наиболее важными из которых являются разделение по физической природе источника сигнала, по предметной области применения и, наконец, по характеристикам самого сигнала.

По физической природе источника регистрируемые биомедицинские сигналы можно разделить на несколько групп [6]:

- *биоэлектрические* — связаны с активностью нервных и мышечных клеток;
- *импедансные* — состоят в изменении импеданса тканей при пропускании переменных токов, с частотами до 1 МГц;
- *биоакустические* - создаются различными физиологическими процессами человека: дыхание, движение и т.д.;
- *биомагнитные* - связаны с излучением человеком чрезвычайно слабых магнитных полей;
- *биомеханические* — включают все виды механических функций человека;
- *биохимические* — являются результатом измерений химических процессов, происходящих в тканях;
- *оптические* - связаны с регистрацией электромагнитных излучений в оптической области спектра, обусловленных функциональной деятельностью биологических систем или наведенных извне.

Основные биологические сигналы, с которыми наиболее часто работают компьютерные БМС, приведены в таблице 1.1.

Табл. 1.1. Характеристики основных физиологических сигналов человека.

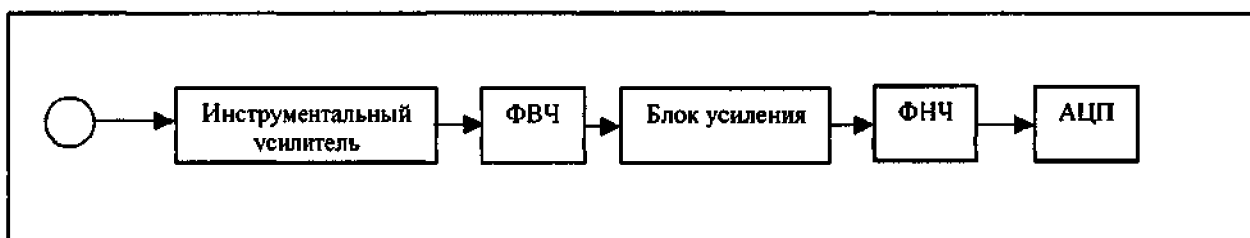
Предметная область	Сигнал	Частотный диапазон	Динамический диапазон
Нейрофизиологическая	Электроэнцефалограмма (ЭЭГ)	0.5 – 100 Гц	2-100 мкВ
	Электроокулограмма (ЭОГ)	0 – 100 Гц	10мкВ – 5мВ
	Электромиограмма (ЭМГ)	500 – 10 кГц	1-10мВ
	Электроретинограмма (ЭРГ)	0.2 – 200 Гц	0.5мкВ – 1мВ
	Электrokортикограмма	100 – 5 кГц	0.5мкВ–10мкВ
Сердечно-сосудистая	Электрокардиограмма	0.05 – 100 Гц	1-10мВ
	Артериальное давление	до 100 Гц	40-200 мм.рт.ст
	Частота сердечных сокращений	до 3 Гц	0-200 уд.
Респираторная	Степень насыщения кислородом	0-1 Гц	
	PO <sub>2</sub> , PCO <sub>2</sub>	0-1 Гц	
	Шум дыхания	до 5 кГц	
	Температура тела	до 1 Гц	0-42С

Классификация сигналов по предметной области основывается на рассмотрении спектра биомедицинских задач, таких как диагностика, мониторинг, решение которых строится путем регистрации, математической обработки и анализа тех или иных электрофизиологических сигналов.

Классификация по физическим характеристикам предусматривает разделение множества сигналов на группы, которые в свою очередь, также делятся на подгруппы в соответствии с такими свойствами, как периодичность, стационарность, детерминированность и т.д.

Различные по своему назначению, компьютерные биомедицинские системы имеют одну общую особенность - источником данных является сам человек. Немалую роль в развитии БМС играет прогресс в области

вычислительных средств и численных методов анализа. Возможности комплексов увеличиваются, совершенствуются методики и алгоритмы анализа данных, вычислительная нагрузка растет и перекладывается полностью с аппаратного обеспечения на программные средства. Ядром таких программных комплексов выступает подсистема цифровой обработки сигналов (ЦОС), от эффективности работы которой и зависит общая производительность системы, ее надежность и потребительские свойства комплекса в целом, в то время как качество биомедицинской информации в значительной мере определяется реализацией аналого-цифрового преобразования и схемотехникой аналоговых входных схем [5, 7, 8] (Рис. 1.1).



*Рис. 1.1. Схема аналоговой части системы сбора электрофизиологической информации.*

Одним из наиболее распространенных подходов при разработке биомедицинских систем является проектирование комплекса на базе аппаратной части персонального компьютера (ПК) или одноплатных ЭВМ, содержащих необходимый набор периферийных устройств со встроенным программным обеспечением. Это дает возможность использовать стандартные операционные системы, обслуживающие ресурсы системы, и сосредоточиться на реализации специфичной для биомедицинской системы функциональности. При этом большое значение получает выбор типа интерфейса ПК или одноплатной ЭВМ с проектируемыми периферийными устройствами. Подобные системы становятся автоматизированными или компьютерными [8].

В процессе проектирования компьютерных БМС часто выделяется два типа основных работ:

- 1) проектирование и реализация аппаратной части устройств регистрации биомедицинской информации (БМИ);
- 2) проектирование и разработка программного обеспечения, в том числе встраиваемого.

При этом, разработка программного обеспечения (ПО) может быть разделена на несколько функциональных этапов:

- 1) выбор программной платформы, под которой понимается операционная система (ОС) и набор предоставляемых ею инструментальных средств;
- 2) разработка архитектуры программного обеспечения системы на основании технического задания и исследование ее свойств на основе методов моделирования;
- 3) реализация отдельных функциональных компонент системы;
- 4) интеграция аппаратных средств регистрации БМИ;
- 5) выбор и реализация коммуникационных каналов для интеграции системы в информационное пространство.

Одним из старейших методов обработки и анализа электрофизиологической информации человека является электрографический метод. Этот метод регистрации и анализа биоэлектрических процессов человека и животных, нашел весьма широкое применение в клинической практике, физиологическом эксперименте, авиационной и космической медицине, исследованиях по физиологии труда и спорта. Столь широкое применение электрографического метода объясняется тем, что он позволяет получить ценную информацию о деятельности тканей, органов и систем. Электрическое возбуждение распространяется в определенных направлениях и последовательности, создавая на поверхности тела электрическое поле. Поскольку электропроводные свойства тканей неоднородны, электрическое поле асимметрично с разностью потенциалов между отдельными участками тела. Это свойство положено в основу метода электрокардиографии, который регистрирует разность потенциалов путем

различных отведений от поверхности тела, что достигается с помощью приборов электрокардиографов [22, 23].

Компьютерные системы ЭКГ-диагностики берут свое начало в начале 70-х годов, когда появились достаточно дешевые и компактные аппаратные средства для их реализации. Первые программные комплексы являлись попыткой автоматизировать процесс получения из ЭКГ диагностической информации с помощью компьютера [8, 27]. Ожидалось, что кроме тех преимуществ, которые дает автоматизация, удастся также повысить диагностическую точность анализа, благодаря применению статистических методов, которые не могут быть использованы при обработке вручную. Результатом развития различных компьютерных систем явился ЭКГ-анализ, позволивший существенно повысить эффективность диагностики, уменьшить относительную частоту диагностических ошибок, сократить время электрокардиографического обследования, освободить квалифицированных врачей от выполнения трудоемких и рутинных процедур измерения и анализа электрокардиограмм. До начала 70-х годов ЭКГ-отведения записывались последовательно одно за другим. Лишь позже появились трехканальные электрокардиографы, позволявшие вести запись сразу трех отведений. Современная регистрирующая аппаратура может одновременно фиксировать до 12 отведений, а также больше, если это необходимо.

Развитие современных методов медицинской диагностики заключается в увеличении объема БМИ, необходимой в клинической практике. Поэтому для современных компьютерных БМС регистрации электрофизиологических сигналов характерна поддержка записи и обработки нескольких различных биоэлектрических сигналов. После ЭКГ наиболее значимыми клинически являются ЭЭГ и ЭМГ-сигналы, также велико значение таких физиологических параметров, как артериальное давление, степень насыщения крови кислородом, температура тела. Приведенная обобщенная схема на рис.1.2. является характерной для реализаций систем на основе компьютеров.

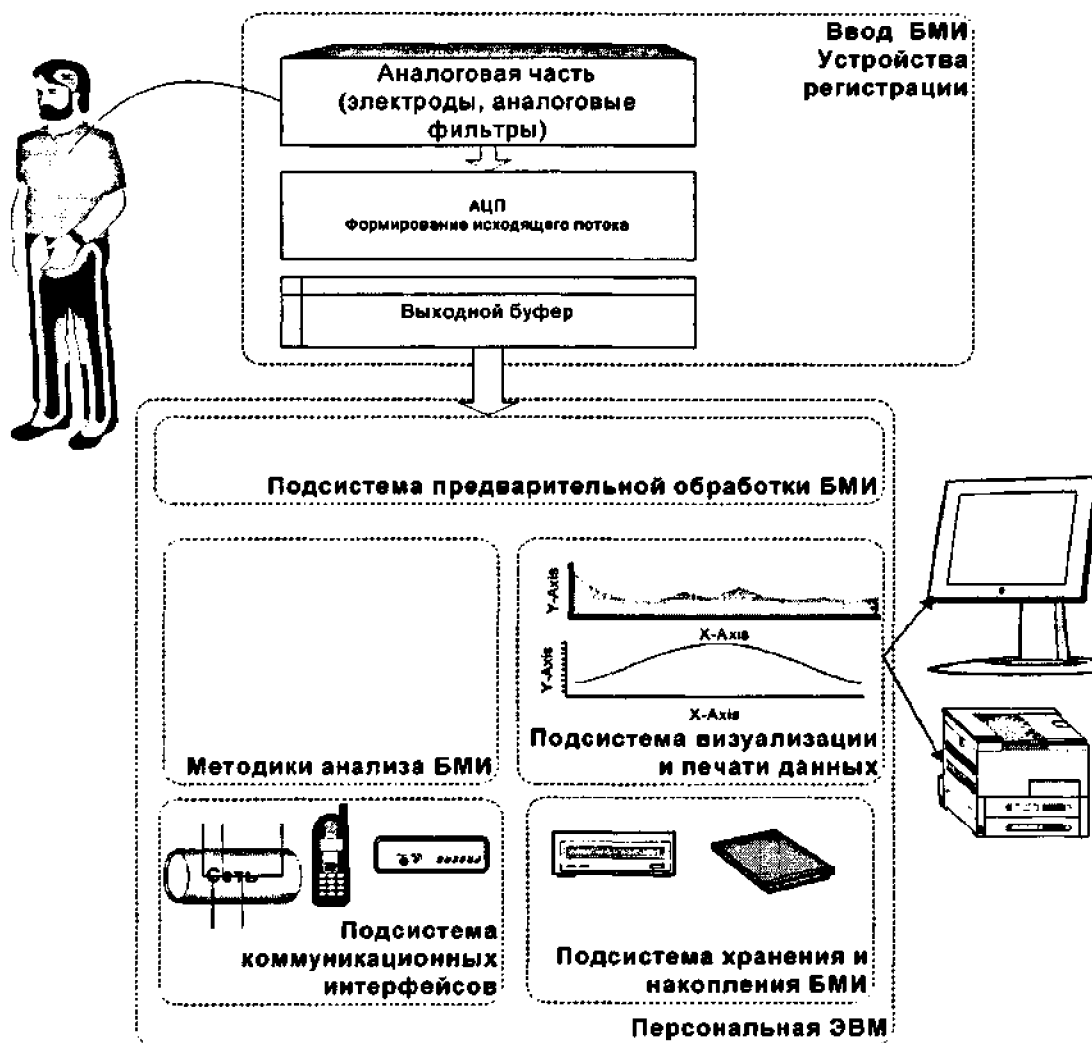


Рис. 1.2. Обобщенная схема компьютерной БМС.

Аппаратная часть представляет собой устройства регистрации и предварительной обработки сигналов, и эта аппаратная часть обычно осуществляет аналого-цифровое преобразование. Буферизация данных на выходе является обычным для систем реального времени с целью синхронизации или упаковки передаваемых данных. Интерфейсной частью со стороны программной системы выступает системное приложение - драйвер, обеспечивающий взаимодействие с аппаратной частью на низком уровне. Средства визуализации и печати обеспечивают представление данных в удобном для анализа формате. Все больший вес приобретает подсистема коммуникационных интерфейсов, поэтому она в данном представлении выделена в отдельный блок.

## **Стандарты в области аппаратно-программных систем регистрации и обработки биомедицинских сигналов.**

Стандартизация в области медицинской информатики в целом, биомедицинских аппаратов и систем в частности, наталкивается на ряд трудностей, вызванных, прежде всего, сложностью в формализации задач и построении единых моделей подобных систем. Однако, несмотря на то, что развитие отрасли биомедицинских систем характеризуется, в значительной мере, противоречивостью и непоследовательностью, стандарты обмена медицинской информацией успешно развиваются и внедряются на уровне крупных и средних информационных систем. Задачей разработки информационных норм является построение на основе моделирования разнообразных биомедицинских данных единой номенклатуры, универсальных методов представления и кодирования данных, а также системы сообщений, позволяющей интегрировать подсистемы различных производителей в рамках единой информационной среды медицинских учреждений. Рост стоимости организации медицинского обслуживания привел разработчиков медицинских систем к необходимости применения нового подхода, главным принципом которого является ориентация на рядового пациента, вовлеченного в информационную среду здравоохранения.

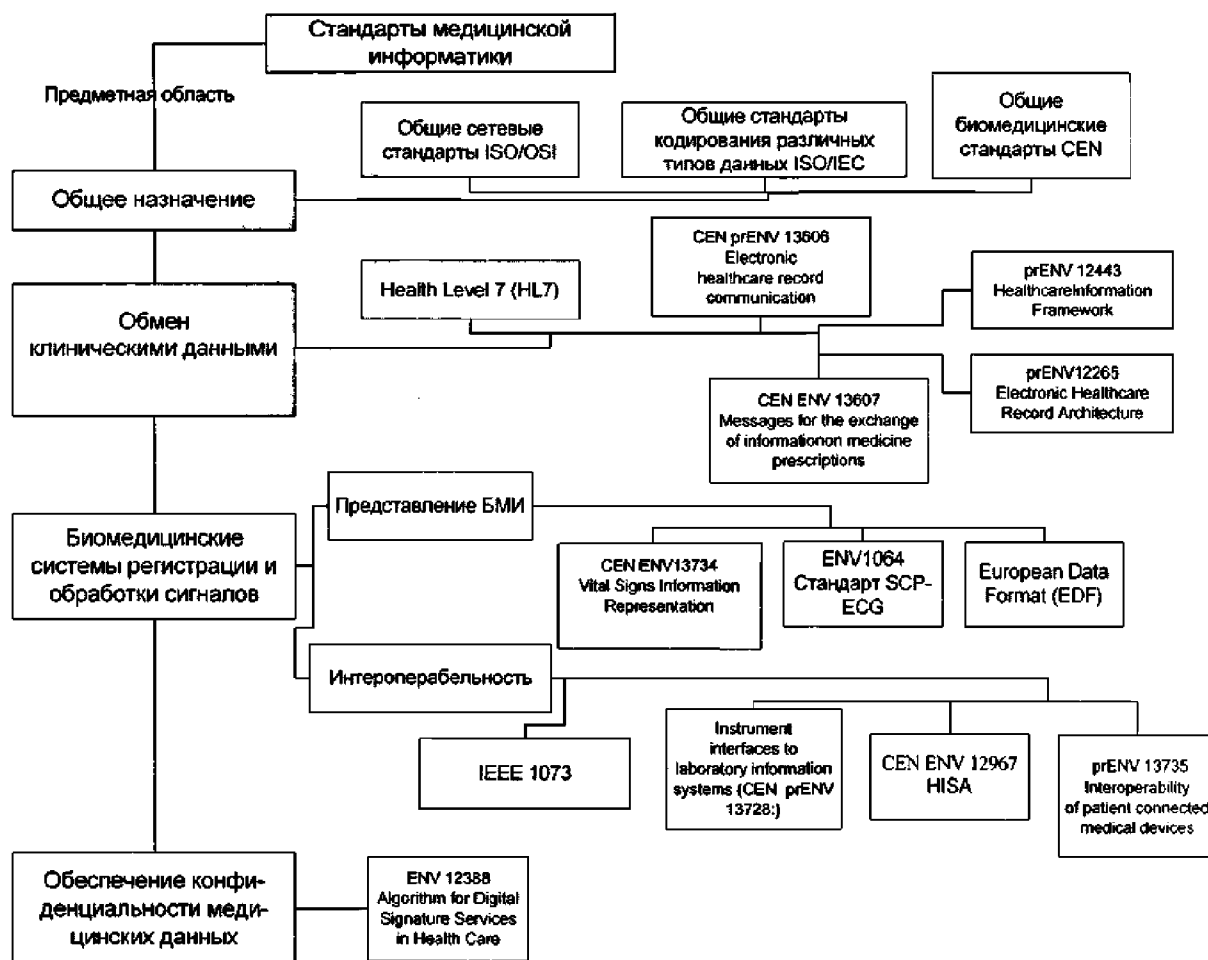
Для настоящего времени характерно разделение вопросов стандартизации медицинских информационных систем масштаба крупных организаций с одной стороны, и биомедицинских программных систем регистрации и обработки данных - с другой. Данный факт в какой-то степени является логичным, поскольку подходы, применяемые в этих двух случаях, фактически противоположны. В первом случае, большое значение уделяется структурам данных, полноте представления и универсальности номенклатуры, а во втором - производительности, универсальности интерфейса и скорости взаимодействия между отдельными компонентами системы.

Создание огромного количества близких по своим функциям, но совершенно несовместимых друг с другом систем, стало серьезным препятствием на пути развития отрасли в целом. Наиболее сложной задачей, стоящей перед разработчиками любой программной биомедицинской системы, является обеспечение высокого уровня совместимости между биомедицинскими приложениями, разработанными различными производителями и работающими на различных платформах. Одной из наиболее важных целей быстро развивающихся процессов стандартизации биомедицинских систем является преодоление имеющихся препятствий на пути интеграции систем и достижения, так называемой *интероперабельности* (от англ. *interoperability*), под которой понимается способность систем эффективно взаимодействовать друг с другом. Данное понятие имеет широкое значение: от возможности различным системам иметь доступ к единым информационным ресурсам, до способности систем эффективно взаимодействовать друг с другом для решения прикладных задач в масштабе реального времени.

Под способностью разнообразных систем к взаимодействию по принципу *plug-and-play* понимают совокупность их свойств, благодаря которым подобные системы могут эффективно быть интегрированы в рамках единого аппаратно-программного комплекса на основании установленных общих принципов и разработанных протоколов. Данное свойство систем подразумевает взаимную совместимость на основании сертификации их соответствия установленным нормам и стандартам. Примерами подобных комплексов могут выступать системы интенсивной терапии, состоящие из набора разнообразных устройств, управляемых единой информационной системой [23]. Укрупненная схема с основными направлениями стандартизации в области медицинской информатики представлена на рис. 1.3.

Различные по своей направленности стандарты разнесены по разным предметным областям: нормы общего назначения, стандарты обмена клиническими данными, обеспечения конфиденциальности, а также

спецификации в области разработки биомедицинских систем регистрации и обработки БМИ.



*Рис. 1.3. Основные направления стандартизации современной медицинской информатики.*

Нормы общего назначения во многом не являются специфичными для области биомедицинской информатики и относятся к информационной науке в целом. Эту категорию формируют нормы общего использования, в частности, основные стандарты построения открытых сетей [9], широко используемые в разработке аппаратно-программных систем, общие стандарты кодирования простых типов данных, а также биомедицинские спецификации общего характера.

Стандарты группы обмена клиническими данными сосредотачивают свое внимание на информационных системах масштаба крупных учреждений. Среди подобных групп стандартов следует выделить хорошо зарекомендовавший себя стандарт HL7 (Health Level 7) [8], официально получивший статус нормы ANSI (American National Standards Institute). HL7 является спецификацией клинического интерфейса, соответствующего седьмому уровню в стеке протоколов OSI (Open System Interconnection-международная программа стандартизации), и решает задачи связанные с организацией электронного обмена разнообразной клинической, финансовой и административной информацией. Основой стандарта выступает так называемая реферативная информационная модель, которая служит основой для методологии построения клинических сообщений, служащих основой протокола обмена. Это позволяет придать процессу формирования сообщений и прочих элементов протокола необходимую логичность. Дополнительно в стандарте приведена модель предметной области с соответствующей номенклатурой и моделями использования. Аналогом HL7 выступает Европейский стандарт Electronic healthcare record communication [9], рассматривающий проблему взаимодействия информационных систем со сходных позиций. В дальнейшем возможно объединение этих двух стандартов в единую спецификацию: в настоящее время подобные работы ведутся в рамках совместной рабочей группы.

Таким образом, в настоящее время можно выделить несколько подходов к решению задач информационного взаимодействия систем:

- прямое использование стандартов представления и обмена медицинской информацией, таких как HL7 или EHCRC;
- применение стандартов представления данных на уровне файлов общего доступа и обмена;
- комбинированное применение форматов представления данных на уровне файловой структуры и стандартов обмена медицинскими записями;

- разработка и реализация частных протоколов обмена.

Долгое время разработка специфичного для производителя внутреннего представления данных и собственных закрытых протоколов обмена являлась обычной практикой, поскольку разработка каких-либо интегрирующих стандартов в области биомедицинских систем является гораздо более сложной задачей в силу широты предметной области и разной специфики. Биомедицинские программные системы регистрации биологических сигналов имеют широкую функциональную направленность, и их стандартизация предусматривает построение абстрактных информационных моделей всех этих систем, жестко регламентирующих механизмы работы всех элементов. Комплекс подобных моделей составляет метамодель предметной области, в рамках которой определена система связей всех элементов, разработана соответствующая номенклатура и формы кодирования событий и данных. В рамках такой метамодели, любой формат представления данных является некоторым логическим следствием системных построений и не привязан к конкретному транспортному уровню.

Группа стандартов в области биомедицинских систем нацелена на решение подобных задач и концентрирует свое внимание на регламентации коммуникационного интерфейса для интеграции данных систем. Существующие интерфейсные стандарты в вычислительной технике (например, IEEE 802.x, USB и т.д.) не способны решить в полной мере задачи по обеспечению интерактивного взаимодействия по принципу plug-and-play разнообразных устройств, осуществляющих регистрацию и обработку физиологических сигналов в режиме реального времени. Универсальный коммуникационный стандарт должен учитывать особенности функционального назначения разнообразных подключаемых устройств - систем регистрации сигналов, систем мониторинга т.д.

Таким образом, можно выделить несколько подходов к разработке интегрируемых систем регистрации и обработки физиологических сигналов:

- 1) применение регламентированных форматов, решающих вопросы

- совместимости на уровне использования общих файлов хранения и обмена;
- 2) создание специализированного промежуточного ПО, реализующего необходимый открытый набор интерфейсов;
  - 3) применение универсальных стандартов проектирования биомедицинских систем регистрации и обработки разнообразных физиологических данных таких, как Medical Information Bus (Также известен как семейство стандартов IEEE 1073) (MIB) [15];
  - 4) комбинированной применение трех представленных выше методов.

Форматы записей медицинских сигналов содержат спецификации представления биомедицинских данных: физиологических сигналов, информации о пациенте, аннотаций и прочей информации в виде бинарной последовательности данных. Данное представление может быть использовано для сохранения полученной информации на различных носителях, централизованной БД или для организации обмена данными между различными программными биомедицинскими системами. В настоящий момент насчитывается большое количество разнообразных открытых форматов представления электрофизиологических данных, многие производители БМС имеют также свой закрытый формат данных.

#### **Требования, предъявляемые к компьютерным биомедицинским системам.**

Современные программные биомедицинские системы являются сложными комплексами, состоящими из множества разнородных компонент. Основным подходом к их построению является создание монолитного приложения, обладающего заранее определенной функциональностью, которая обеспечивается жестко связанной группой компонент-модулей. В качестве примера можно привести типичную систему ЭКГ-диагностики, работающую под управлением операционной системы DOS. Она строится из законченных исполняемых модулей, взаимодействие между которыми осуществляется

набором командных файлов: обмен данными обычно происходит через временные файлы. Обладая набором определенных характеристик, такая система способна обеспечить решение лишь тех задач, которые ставились перед разработчиками на этапе ее создания. С точки зрения пользователя система предоставляет законченный интерфейс, обладающий минимальными возможностями адаптации, а не редко и вовсе лишенный таковых. Одновременно с этим, такой интерфейс предполагает полную однозначность дерева реализуемых функций, прост в документировании и использовании. Однако по завершению работ над проектом развитие подобной системы застывает, или же она устаревает еще во время разработки. Причины этого кроются в проблемах, связанных с расширяемостью, надежностью и невозможностью повторного использования кода, свойственных такой архитектуре. Таким образом, подобная система не оправдывает как ресурсов, потраченных на ее создание, так и не оставляет конечному пользователю возможности модернизации существующего оборудования с развитием новых методов компьютерной диагностики.

На основе приведенных фактов можно сформулировать требования, предъявляемые к современным комплексам сбора и математической обработки ЭКГ-данных, которые можно разделить на две группы.

Архитектурные:

- модульная организация системы;
- гибкая модель взаимодействия и взаимозаменяемости модулей (динамическое связывание);
- открытая архитектура управления взаимодействием модулей (возможность создания новых модулей и моделей их взаимодействия сторонними разработчиками);
- надежность комплекса, которая не должна уступать надежности систем, построенных на базе жестких моделей взаимодействия.

Функциональные:

Не исключая требования, характерные для большинства систем этого

класса, отметим наиболее важные для представленной реализации:

- обеспечение легкости в обучении пользовательскому интерфейсу;
- возможность адаптации комплекса как для различных применений (клинические, лабораторные, научно-исследовательские работы), так и для конкретного пользователя;
- документированный интерфейс взаимодействия с оборудованием, обеспечивающий возможность работы с приборами разных производителей;
- общее хранилище данных, доступное каждому подмодулю системы и инкапсулирующее методы взаимодействия с различными внешними источниками данных (СУБД, системы резервирования, сетевые ресурсы);
- интеграция комплекса в современные медицинские системы посредством реализации требований стандарта IEEE 1073.

Нетрудно заметить, что перечисленные требования обладают определенными противоречиями, которые в первую очередь продиктованы проблемой выбора между жестким и гибким архитектурными решениями.

## **2. Проектирование компьютерных биомедицинских систем реального времени**

Потребность в прикладном программном обеспечении за последние годы растет значительными темпами. Это связано с проникновением вычислительных систем в новые области применения, характеризующиеся постоянным повышением сложности программных систем, что приводит к увеличению времени реализации и падению надежности. Производительность же процесса разработки за последнее двадцатилетие выросла незначительно, в основном, за счет совершенствования вычислительных систем и средств разработки. Ранее, когда аппаратные ресурсы вычислительных систем были незначительны, основной задачей теории программирования являлось

совершенствование алгоритмических методов решения конкретных задач. В настоящее время принята обобщенная модель представления программной архитектуры: "Программная архитектура - структурная модель, представляющая программную систему множеством компонентов, соединением этих компонентов и различными способами реализации". Разработка теории проектирования программных систем началась сравнительно недавно и в настоящий момент предметная область не является полностью и однозначно определенной. Можно выделить несколько категорий предметной области программной архитектуры (ПА), иными словами, терминологии.

К первой из таких категорий относится ПА потоков данных, которая основывается на рассмотрении модели программной системы, состоящей из двух разновидностей компонент: каналов передачи и фильтров. В рамках данной модели каналы обеспечивают доставку данных от одного компонента-фильтра к другому. Фильтр, обладающий входом и выходом, осуществляет некоторые операции с данными, поступившими на его вход, и выдающий результат на выход, при этом к одному выходу можно подключить несколько каналов. Данная архитектура является прямой реализацией модели потоков данных, известной достаточно давно.

Вторая категория - ПА параметрического описания функций, основанная на соответствующих классической парадигме программирования компонентах - подпрограмм, образующих иерархическую структуру. На взаимодействие компонент не налагаются какие-либо ограничения в зависимости от уровня их реализации. Механизм удаленного вызова подпрограмм, обеспечиваемый сервисами аналогичными сервисам удаленного вызова процедур, или ПО промежуточного слоя, является предметной областью данной категории.

Третьей категорией является объектно-ориентированная ПА, которая основывается на иерархической структуре объектов, инкапсулирующих данные и соответствующие функции работы с ними. Взаимодействие между компонентами сводится к механизмам межобъектного взаимодействия.

К четвертой категории можно отнести многослойные, или многозвенные, ПА. Компоненты такой ПА являются слоями, реализующими дополнительную функциональность на основе функций, лежащих на нижнем уровне слоев. Реализация слоя не является существенной, он может являться, например, программной библиотекой, "надстроенной" над другой библиотекой, или же транслятором из одного языка в другой, для которого уже есть интерпретатор. Примерами подобных ПА могут служить виртуальные машины языков.

Пятой категорией являются ПА независимых компонент, представляющие собой исполняемые сущности, обменивающиеся сообщениями и реагирующие на сообщения на основе так называемой "модели событий". Последняя фактически представляет собой аналог уникальной для каждого компонента таблицы, ставящей в соответствие сообщению инициацию выполнения компонентом некоторого действия. Подобные таблицы получили название интерфейсов. Данная модель обеспечивает 100%-ное повторное использование кода, однако достаточно трудоемка при реализации.

Очевидно, что архитектура системы не может быть исчерпывающе описана на основании одной из пяти вышеописанных ПА. Структура любой системы может быть рассмотрена с различных позиций, например, ПА независимых компонент и объектно-ориентированной ПА.

Основным средством поддержки разработки значительных по объему проектов в настоящее время является применение объектно-ориентированной методологии проектирования (ООП). ООП фактически единственный в настоящее время подход, позволяющий эффективно решать вопросы проектирования сложных программных систем. Любая зрелая методология должна включать в себя возможность качественного и количественного анализа архитектурных решений, выбранных при проектировании той или иной системы, оценивать ее количественные характеристики - сложность, структуру, ожидаемое количество дефектов, длительность процесса отладки. Объединение различных идеологий и методов дало унифицированный объектно-ориентированный подход, получивший название UML - Unified modeling

language. UML - это единый метод объектно-ориентированного анализа, включающий методы визуализации, описания и документирования ООП.

Широко используемым понятием в теории программирования является программная платформа, которая тесно связана с интерфейсом прикладного программирования (ИПП). Под ИПП понимается набор базовых сервисных функций, предоставляемых операционной системой (ОС) или дополнительными компонентами ОС приложениям. Разнообразие используемых ИПП и выбранную ОС принято называть программной платформой (ПП). Оптимальный выбор ПП, инвестиции на приобретение качественных программных и аппаратных средств является одним из стратегических решений, определяющих успех разработки в целом [13, 14, 15].

Важной стороной поставленной задачи является учет требований реального времени, соблюдение которых для систем регистрации и обработки сигналов необходимо.

В настоящий момент наметилась тенденция отказа от частных решений в пользу концентрации усилий на реализациях, опирающихся на общепринятые стандарты. Это, прежде всего, объясняется тем, что стандартизация в области подобных систем подошла к уровню, достаточному для начала работ по коммерческой реализации.

Для организации взаимодействия между различными устройствами регистрации медицинских сигналов и клиническими информационными системами необходимо использовать единый коммуникационный стандарт, удовлетворяющий ряду требований, наиболее важными из которых являются следующие:

- *уникальность имени*. Устройство должно иметь идентификатор, позволяющий однозначно выделить его в сети.
- *надежность*. Стандарт должен обеспечивать надежную работу в случае отказа одного или нескольких устройств.
- *совместимость с производимым оборудованием*. Технология реализации стандарта должна позволять использовать имеющуюся

аппаратно-программную реализацию настолько, насколько это возможно.

- *поддержка множества физических интерфейсов.* Обеспечение возможности устанавливать соединение, используя множество доступных протоколов.

*поддержка множества сетевых топологий.* Поддержка прозрачности доступа к компонентам системы (устройствам, системе контроля) из внешних сетей.

*синхронизация по времени.* Обеспечение синхронизации передаваемых данных по времени.

- *совместимость с клиническими информационными системами.*

Должна быть обеспечена возможность передачи данных в информационную среду соответствующего стандарта обмена медицинскими записями.

- *защищенность.* Интерфейс должен обеспечивать защищенность и конфиденциальность передаваемых данных.

- *масштабируемость.* Архитектура должна позволять изменять количество элементов системы.

Усилия по координации работ групп IEEE по стандарту MIB и комитета TC251 Европейского института по стандартизации говорит в пользу единого семейства стандартов, которые уже в ближайшем будущем, возможно, будут приняты ISO в качестве полноценных норм.

### **3. Компьютерные системы регистрации электрокардиограмм**

Проектирование любой программной системы включает в себя целый комплекс вопросов, связанных с выбором методологии проектирования, разработки программной архитектуры и выборе программной платформы, применения методов оценки качества программных приложений, гибкости реализации и тестирования функционального содержания, формирования

требований рекомендательного характера, ориентированных на повышение качества продукта. Важным этапом проектирования является исследование структуры (структуризации) подобного класса систем с целью выявления общих элементов структуры, выработки единых требований к реализации. Это дает возможность подойти к разработке с позиций универсального подхода, дающего возможность использовать единые методы для построения целого класса систем. Поэтапная структура исследования, включая первый этап, состоящий в изучении проблемы, формулирования целей и задач исследования, рассмотренный в предыдущих главах, представлена на рис. 2.1.

Основным методом структуризации в данном исследовании явился системный метод, являющийся основой теории системного анализа.

В системно-структурном методе акцент преимущественно делается на исследование внутренней структуры системы, внутренних связей, тогда как системный подход, кроме этого, подразумевает исследование поведения системы и ее элементов в зависимости от связей с внешней средой, от происхождения ситуации, в которую попадает система. В данном случае, в понятие структура вкладывается смысл отображения определенных взаимосвязей, взаиморасположение составных частей системы, ее устройства (строения).

Методы системного анализа являются основой разработки технических средств, программного обеспечения [23], при этом регламентируется процесс проектирования ПО от этапа создания и сопровождения моделей системы до фазы тестирования и внедрения (этап 2).

В качестве метода формализованного представления систем в рамках данной методологии используются лингвистические, графические представления, языки моделирования.

Предварительный, поверхностный анализ систем требует их представления в виде схем, в которых фиксируются наиболее существенные параметры и связи между ними. При этом элементы данных схем не подвергаются какой-либо детализации до уровня конкретных используемых

технологий. Такие схемы получили название концептуальных моделей, или структурных моделей [28], и являются отправной точкой исследования (этап 3).

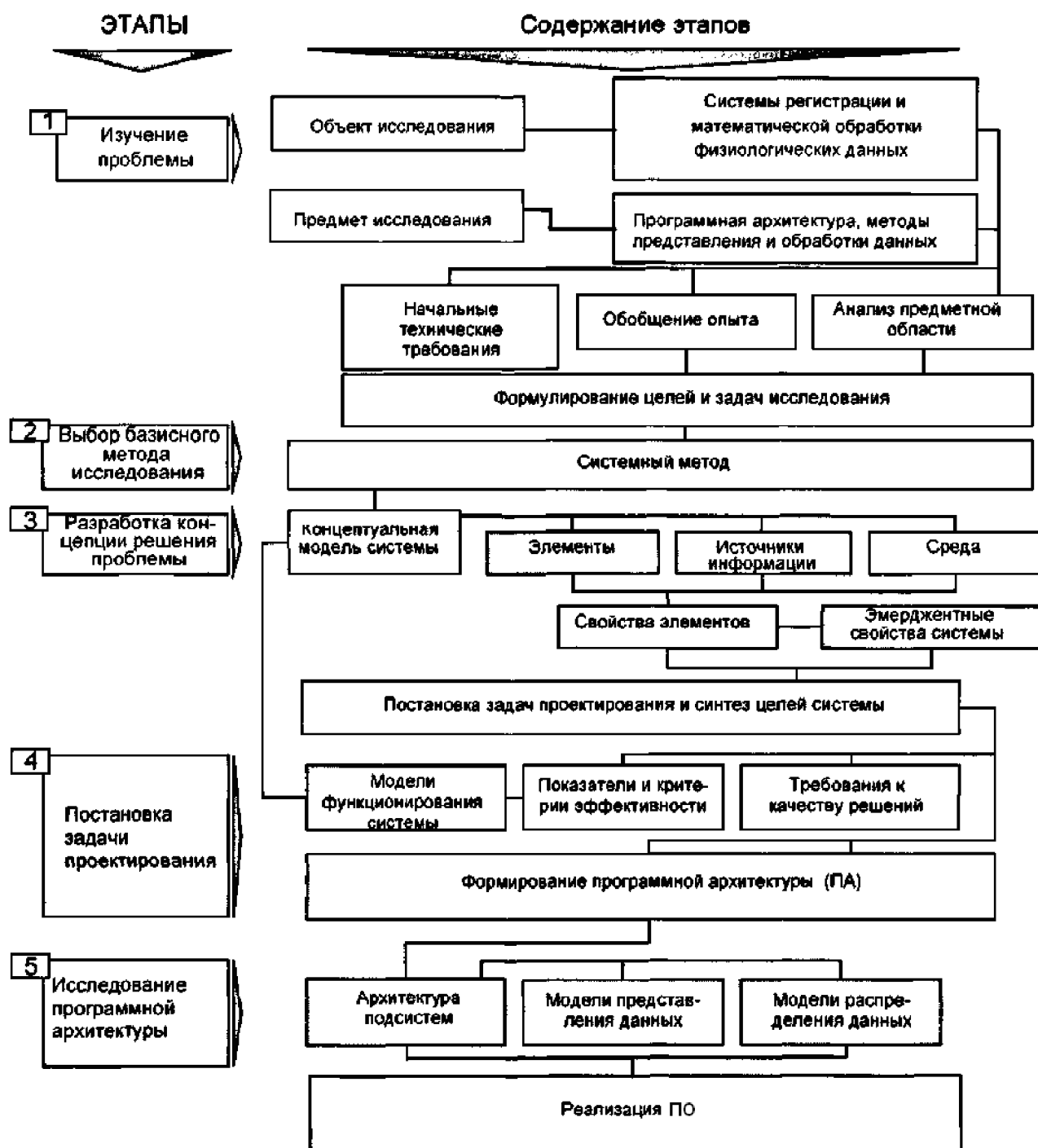


Рис. 2.1. Структурная схема проведения исследований.

Основными параметрами концептуальных моделей считаются элементы системы, источники информации, внешняя среда, каждый из которых характеризуется своими свойствами, а система - свойствами целостности. Эти

свойства формируют совокупность факторов, влияющих на поведение системы. Важнейшим этапом исследования является переход от разработанных структурных представлений о системе, к программной архитектуре (этап 4) и ее дальнейшей реализации. Архитектурные решения должны приниматься только тогда, когда стала ясна структура системы в целом, т.е. полностью сформулированы требования, предъявляемые к системе.

Для решения поставленных задач необходимо обладать определенным набором приемов решения, среди которых наиболее результативны модели функционирования системы, среди которых стоит выделить модели процессов, формализующих механизмы параллелизма и синхронизации, представления и накопления данных, а также модель развертывания, отражающую топологию используемых аппаратных средств. Модель реализации служит заключительным представлением, реализующим покомпонентную конфигурацию системы (этап 4).

Завершающим этапом на основании полученных результатов выносятся общее суждение о функционировании исследуемой программной системы, выявляются закономерности, присущие системе, которое приводит к формулированию заключения по теме.

Особенностью цифровых компьютерных комплексов регистрации электрофизиологической информации, осуществляющих запись и обработку данных непосредственно во время записи сигнала, является принадлежность к системам реального времени, так как функциональная часть этих систем должна непрерывно обрабатывать и анализировать поступающую информацию и оперативно реагировать на изменившееся состояние внешних параметров. Так, например, для полиграфических систем регистрации электрофизиологической информации такими сигналами могут являться ЭКГ, ЭЭГ, ЭМГ и ЭОГ, а также целый спектр физиологических сигналов, приведенный в табл.1.1. Исследовать структуру подобных аппаратно-программных систем можно с использованием двух подходов: информационного и функционального.

Информационный подход подразумевает анализ на основе рассмотрения транспортных потоков данных, циркулирующих между элементами системы, в то время как функциональный основывается на рассмотрении разнообразных связей внутри комплексной системы, выделяя в отдельные единицы функционально законченные компоненты. С точки зрения формализации наиболее проработанным является информационный подход, поскольку он предоставляет возможность количественно и качественно проанализировать и представить структуру связей на основе математических методов. Функциональная схема отражает структуру в виде множества законченных элементов, реализующих определенные функции в рамках требований, предъявляемых к комплексам ЭКГ-диагностики, а также предоставляющих различные дополнительные диагностические или сервисные возможности. Детализация функциональной схемы подразумевает учет конкретной программной платформы, на базе которой строится реализация, накладывающей свой отпечаток на структуру связей и набор функциональных компонент, присутствующих в структурной схеме. Более того, цели и задачи самой биомедицинской системы также играют важную роль. Например, функциональная схема компьютерных систем ЭКГ-диагностики может существенно различаться по структуре (рутинная ЭКГ, ЭКГ физических нагрузок, системы прикроватного мониторинга) и иметь четко выраженную близость с используемым аппаратным обеспечением. Все дальнейшее рассмотрение будет касаться систем, ориентированных на получение и обработку БМИ в режиме реального времени.

На Рис.2.2 представлена структура многокомпонентной, полиграфической биомедицинской системы регистрации и обработки электрофизиологической информации, включающая основные элементы, отвечающие общим требованиям, предъявляемым к системам такого рода. Данная схема дает только общие представления о структуре систем и не отражает функциональную логику связей между отдельными его элементами в рамках каждой подсистемы. Выделение независимых подсистем заключается в

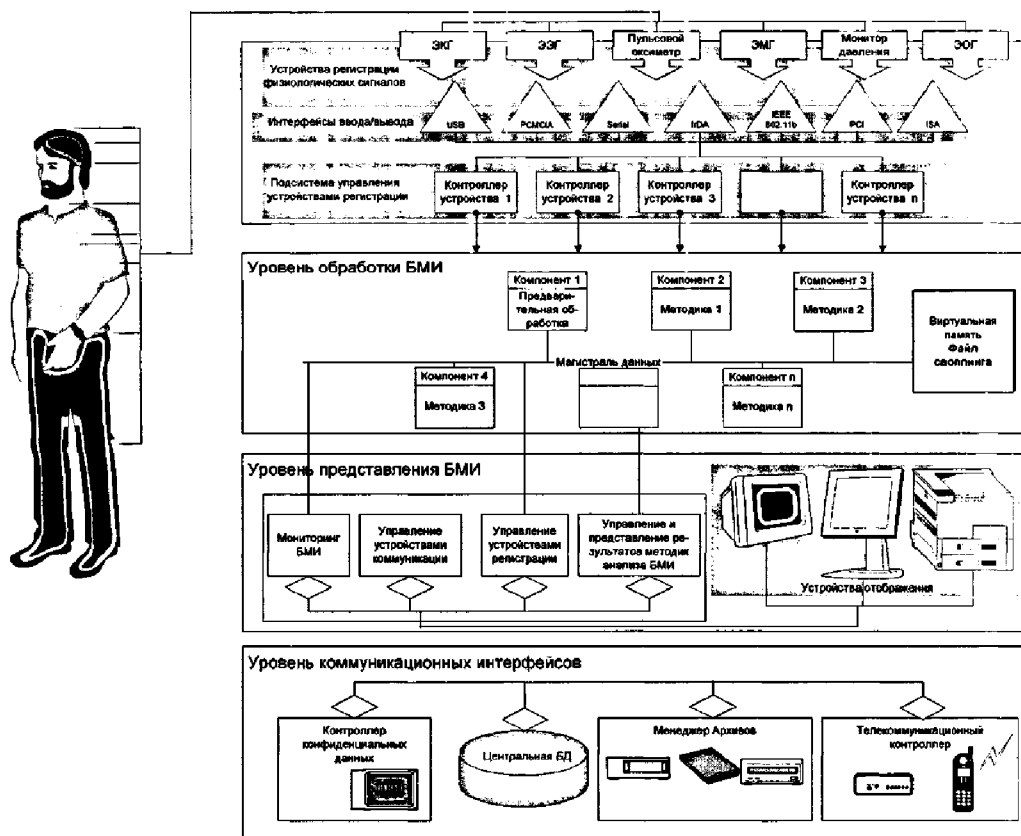
выделении отдельных элементов с сильными функциональными взаимными связями. Для анализа функциональных связей отдельных подсистем необходим детальный анализ структуры подобных систем.

Структурно, произвольная компьютерная БМС может быть разделена на 4 уровня: уровень сопряжения с устройствами регистрации, уровень обработки БМИ, представления и коммуникационных интерфейсов.

Для обеспечения доступа к устройствам в большинстве современных ОС, в том числе и семейства Windows, необходимо наличие драйвера, чтобы обеспечить полный доступ к аппаратному обеспечению на уровне ядра ОС. Множество поддерживаемых аппаратных интерфейсов для подключения внешних устройств в настоящее время является достаточно широким, поэтому использование шин ISA/PCI для обеспечения высокоскоростного обмена не является необходимым. Среди поддерживаемых производителями портативных ПК интерфейсов, необходимо выделить USB, IrDA, Wi-Fi (стандарт IEEE 802.11b), Bluetooth, как наиболее перспективные для сопряжения с устройствами регистрации БМИ, поддерживающие скорости передачи от нескольких Мбит/с до десятков Мбит/с.

Таким образом, программный слой работы с оборудованием составляют компоненты-контроллеры, обеспечивающие прозрачный интерфейс доступа к соответствующим драйверам устройств, которые могут быть стандартными в случае использования поддерживаемых ОС интерфейсов, или специально разработанными для данного устройства при использовании стандартов ISA/PCI/PCMCIA. В предметную область стандарта IEEE 1073 контроллеры устройств внесены под наименованием виртуального медицинского устройства (ВМУ), так как с точки зрения БМС данный компонент является абстрактным представлением реального аппаратного устройства. Для обеспечения расширенной функциональности в подсистему управления устройствами регистрации может быть добавлен менеджер устройств, который позволяет пользовательскому процессу динамически подключаться к различным устройствам, сохраняя их параметры и обеспечивая стандартный интерфейс

доступа. В качестве подобных устройств, например, в комплексе полнофункциональной диагностики ЭКГ могут выступать, собственно, регистратор ЭКГ, а также пульсовый оксиметр и монитор артериального давления (АД).



*Рис. 2.2. Концептуальная модель полиграфической системы регистрации и анализа БМИ.*

В целом, архитектура системы должна обеспечивать возможность подключения различных устройств в зависимости от назначения и требований диагностических методов, закладываемых в систему на этапе разработки или в процессе ее совершенствования.

Наиболее важным уровнем является уровень обработки данных, который составляют компоненты, выполняющие обработку данных, как в режиме реального времени, так и в автономном режиме. Здесь наибольшую важность приобретает возможность динамического изменения конфигурации

обработчиков данных, добавление и удаление компонент во время работы. Это позволяет эффективно изменять заложенную на этапе проектирования системы функциональность в соответствии с новыми требованиями.

Уровень представления БМИ представлен компонентами, осуществляющими визуализацию БМИ и результатов анализа, а также пользовательский интерфейс управления различными подсистемами.

Выделение коммуникационных модулей в отдельный уровень обеспечивает режимы удаленного взаимодействия и организацию видеосвязи. Коммуникационная подсистема может обеспечивать интеграцию БМС с централизованной БД медицинского учреждения, сохранение и ведение архивов медицинских данных, поддержку телекоммуникационных функций, что особенно важно для портативных систем. При этом возможно широкое применение разнообразных доступных средств передачи данных. Поддержка смарт-карт позволяет обеспечивать конфиденциальность доступа к БД и защищенное управление данными пациентов.

**Характеристики электрокардиограммы.** Метод исследования биоэлектрической активности сердца, получивший название электрокардиографии, является сегодня незаменимым в диагностике нарушений ритма и проводимости, гипертрофии миокарда предсердий и желудочков, ишемической болезни сердца и других заболеваний. Метод основан на регистрации электрических потенциалов, возникающих в сердце.

Последовательность возбуждения мышечных масс сердца, обусловленная автоматизмом синусового узла и проведением импульса по проводящей системе сердца к отдельным кардиомиоцитам, формирует мощный электрический потенциал сердечной мышцы, регистрируемой с помощью накладываемых на поверхность тела электродов. Регистрация этих потенциалов во времени и графическое их отображение и есть электрокардиограмма. Изменение разности потенциалов, возникающих на поверхности тела при возбуждении сердца, записываются с помощью различных систем отведений. Каждое отведение, состоящее из пары электродов подсоединенных: один к

положительному, другой к отрицательному полюсу входа усилителя электрокардиографа, регистрируют разность потенциалов, существующую между двумя определенными точками электрического поля сердца, в которых установлены электроды. В связи с тем, что многочисленные способы расстановки электродов могут привести к сложностям взаимопонимания врачей в трактовке ЭКГ, в международную клиническую практику была введена наиболее часто используемая система 12 отведений, состоящая из трех стандартных отведений, трех усиленных однополюсных от конечностей и шести грудных отведений (рис.2.3).

Стандартные двухполюсные отведения, предложенные в 1913 году открывателем ЭКГ Эйнтховеном, регистрируют разность потенциалов между двумя точками, удаленными от сердца и расположенными во фронтальной плоскости тела - на конечностях. Для записи этих отведений принята разноцветная маркировка электродов:

правая рука - электрод красного цвета;

левая рука - электрод желтого цвета;

левая нога - электрод зеленого цвета;

правая нога - электрод черного цвета

Попарное включение перечисленных электродов (кроме черного, подключенного к заземляющему проводу) формирует систему трех стандартных отведений.

I отведение - правая рука (минус) и левая рука (плюс)

II отведение - правая рука (минус) и левая нога (плюс)

III отведение - левая рука (минус) и левая нога (плюс)

В 1942 году Гольдбергер, используя упомянутую систему наложений электродов предложил регистрировать разность потенциалов между одной из конечностей, на которой установлен положительный электрод и средним потенциалом двух других конечностей. Этот метод получил название усиленных однополюсных отведений от конечностей и имеет стандартную маркировку:

**avR** - усиленное однополюсное отведение от правой руки;

**avL** - усиленное однополюсное отведение от левой руки;

**avF** - усиленное однополюсное отведение от левой ноги.

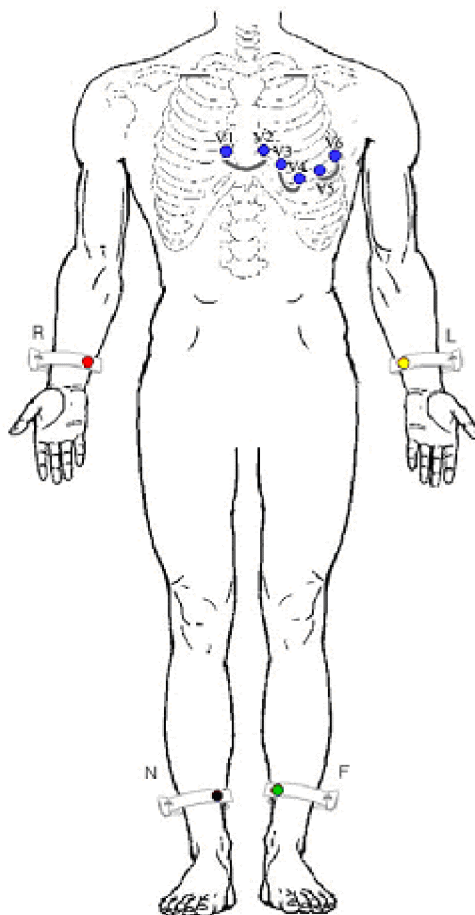


Рис. 2.3. Пример расположения внешних электродов при регистрации 12 отведений ЭКГ с кабеля для 6 электродов

Систему стандартных и усиленных от конечностей отведений дополняют однополюсные отведения от грудной клетки, предложенные в 1934 г. Wilson. Они регистрируют разность потенциалов между положительным грудным электродом, расположенным в разных точках и объединенным электродом Вильсона, объединяющим три электрода от конечностей. Грудные электроды маркируются буквой V и расположены в следующих позициях:

**V1** - четвертое межреберье справа от грудины;

**V2** - четвертое межреберье слева от грудины;

**V3** - промежуточная позиция V2 и V4;

**V4** - пятое межреберье по левой срединно-ключичной линии;

**V5** - пятое межреберье по левой передней подмышечной линии;

**V6** - пятое межреберье по левой средней подмышечной линии;

Иногда в целях расширения диагностических возможностей 12 общепринятых отведений используют дополнительную расстановку электродов отведений Вильсона:

**V7** - пятое межреберье по левой задней подмышечной линии;

**V8** - пятое межреберье по левой лопаточной линии;

**V9** - пятое межреберье по левой околопозвоночной линии.

Нормальные значения длительности интервалов сердечного цикла, в зависимости от частоты сердечных сокращений, часто обозначаемых как ЧСС, приведены в трех отдельных таблицах 2.1, 2.2 и 2.3. с указанием длительности интервала и частоты сердечных сокращений в минутах.

Таблица 2.1. Зависимость ЧСС от длительности интервала R-R

Длительность интервала R-R, с	ЧСС в мин.	Длительность интервала R-R, с	ЧСС в мин.
1.50	40	0.85	70
1.40	43	0.80	75
1.30	46	0.75	80
1,25	48	0.70	86
1.20	50	0.65	82
1.15	52	0.60	100
1.10	54	0.55	109
1.05	57	0.50	120
1.00	60	0.45	133
0.95	63	0.40	150
0.90	66	0.35	172

Таблица 1.2. Минимальная и максимальная длительность интервала Q-T

ЧСС, уд/мин	Длительность интервала Q-T, с	ЧСС, уд/мин	Длительность интервала Q-T, с	ЧСС, уд/мин	Длительность интервала Q-T, с
40-41	0,42-0,51	66-67	0,33-0,40	101-104	0,27-0,32
42-44	0,41-0,50	68-69	0,33-0,39	105-106	0,26-0,32
45-46	0,40-0,48	70-71	0,32-0,39	107-113	0,26-0,31
47-48	0,39-0,47	72-75	0,32-0,38	114-121	0,25-0,30
49-51	0,38-0,46	76-79	0,31-0,37	122-130	0,24-0,29
52-53	0,37-0,45	80-83	0,30-0,36	131-133	0,24-0,28
54-55	0,37-0,44	84-88	0,30-0,35	134-139	0,23-0,28
56-58	0,36-0,43	89-90	0,29-0,34	140-145	0,23-0,27
59-61	0,35-0,42	91-94	0,28-0,34	146-150	0,22-0,27
62-63	0,34-0,41	95-97	0,28-0,33	151-160	0,22-0,26
64-65	0,34-0,40	98-100	0,27-0,33		

Таблица 2.3. Максимальная нормальная продолжительность интервала P-Q в зависимости от ЧСС

ЧСС, уд/мин	Продолжительность интервала	ЧСС, уд/мин	Продолжительность интервала
40	0,20	90	0,145
50	0,19	100	0,135
60	0,175	110	0,13
70	0,16	120	0,125
80	0,15	130-160	0,12

Система отведений Нэба позволяет уточнить изменения ЭКГ, наблюдаемые при повреждениях боковой и задней стенки левого желудочка.

**Элементы ЭКГ.** Зарегистрированный на ЭКГ потенциал сердца содержит ряд последовательных зубцов, обозначаемых буквами P, Q, R, S, T (рис.2.4). Форма и величина этих зубцов различна в разных электрокардиографических отведениях. Зубец P отражает деполяризацию

предсердий. Его максимальная величина наблюдается во II-ом стандартном отведении и составляет 1,5 - 2,5 мм, а длительность не превышает 0,1 с.

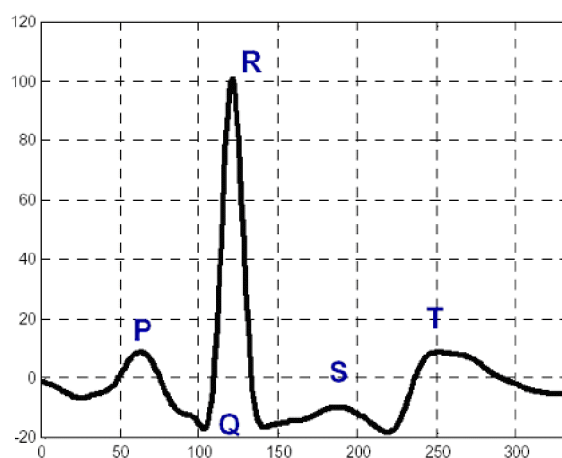


Рис. 2.4. Ряд последовательных зубцов ЭКГ.

У здоровых людей в отведениях I, II и  $avF$  зубец P всегда положительный, в отведениях III и  $avL$  - может быть положительным, двухфазным или отрицательным, а в отведении  $avR$  - всегда отрицательный, в отведении V1 - часто двухфазный. Интервал P-Q отражает продолжительность атриовентрикулярного проведения и измеряется от начала зубца P до начала комплекса QRS. Длительность интервала P-Q составляет от 0,12 до 0,20 сек и прямо зависит от частоты сердечного ритма. Зубец Q обусловлен вектором деполяризации межжелудочковой перегородки, регистрируется отрицательным в отведениях I, II, III,  $avL$ ,  $avF$ . Его величина не должна превышать  $1/4$  зубца R и продолжительность - 0,03 с. Зубец R отражает суммарный вектор деполяризации правого и левого желудочков. Регистрируется положительным в I, II, III,  $avL$  и  $avF$  отведениях, а также отведениях V1-6. Его максимальная величина во II-ом стандартном отведении, в грудных - зубец R постепенно увеличивается от отведения V1 к V4, а затем несколько уменьшается к V6. В отведениях от конечностей его величина не превышает 20 мм, а в грудных отведениях - 25 мм.

Продолжительность зубца R составляет примерно 0,04 сек. Зубец S обусловлен проекцией на оси отведений конечного вектора распространения

волны возбуждения в базальных отделах межжелудочковой перегородки - правого и левого желудочка. Амплитуда зубца S подвержена значительным колебаниям даже у здоровых людей, но как правило не превышает 20 мм. Часто в стандартных отведениях зубец S может отсутствовать, а в грудных его амплитуда постоянно уменьшается от отведения V1 к V6. Минимальная продолжительность всего желудочкового комплекса QRS находится в пределах 0,07-0,1 сек. Сегмент S-T соответствует периоду полного охвата возбуждением обоих желудочков, когда разность потенциалов между участками сердечной мышцы мала. Смещение сегмента S-T вверх и вниз от изолинии не должно превышать 0,5 мм в отведениях V 4, 5, 6, а в отведениях V 1 - V 3 - вверх от изолинии не более 2 мм.

Зубец T отражает процесс быстрой реполяризации миокарда желудочков. В норме зубец положительный в отведениях I, II, avF, V2 - V6, а в отведениях III, avF, V1 - может быть положительным, двухфазным и отрицательным. На ЭКГ отведениях максимальная амплитуда зубца T соответствует максимальной амплитуде зубца R. Одним из важных показателей ЭКГ является длительность интервала Q-T, измеряемая от начала зубца Q до окончания зубца T. Его величина определяется по известной формуле Базетта, где:  $QT = 0,39 (R - R)$ , где 0,39 - коэффициент, R -R - расстояние между зубцами R на ЭКГ. Увеличение интервала QT может обуславливать появление опасных расстройств ритма сердца.

#### **4. Анализ погрешностей каналов измерения**

Для повышения качества диагностики заболеваний сердца на ранней стадии необходимо, в частности, увеличивать точность измерения сигнала кардиограммы [20, 21, 22, 23], Обычные аналоговые электрокардиографы представляют собой электронно-механические системы. Напряжение, наведённое на электроды в результате сердечной деятельности, усиливается

электронным усилителем и регистрируется на бумажной ленте при помощи механического самописца.

К усилителю предъявляются жёсткие требования по линейности, отношению сигнал/шум, дрейфу нуля, равномерности амплитудно-частотной характеристики и т.д. В то же время самописцу, как и всем механическим системам присущ ряд недостатков, связанных с подверженностью износу, необходимостью периодического технического обслуживания, погрешностью изготовления, т.е., в процессе эксплуатации характеристики самописца меняются, и запись ЭКГ становится менее достоверной.

Сам самописец обладает постоянной времени. Учитывая всё это, в технической документации прилагаемой к кардиографу, заводом-изготовителем указывается реальная погрешность измерения порядка 10-20%. Для диагностики, например, некоторых нарушений ритма сердца (НРС) нужна длительная запись потенциалов сердца.

Однако при длительном мониторинге длина ленты может выражаться десятками метров, и обработка ЭКГ становится неудобной. Имеется ещё ряд задач в кардиологии, когда аналоговая регистрация становится тормозом. Учитывая изложенное, целесообразно отказаться от механического звена в цепи обработки ЭКГ и записывать потенциалы сердца в память компьютера.

При этом открываются большие возможности по автоматизации анализа кардиограмм. Точность записи и разрешающая способность метода компьютерной ЭКГ позволяют получить большую информацию, и тем самым увеличить достоверность диагноза на ранней стадии заболевания.

Подсчет ошибки, возникающей при аналого-цифровом преобразовании. Квантование, при котором сигнал представляется ближайшим уровнем аналого-цифрового преобразования, называется округлением (рис.2.5). Очевидно, что максимальная абсолютная ошибка округления не превышает половину ступени квантования: чем больше входной сигнал, тем меньше погрешность, естественно, при условии  $U < U_{\max}$ .

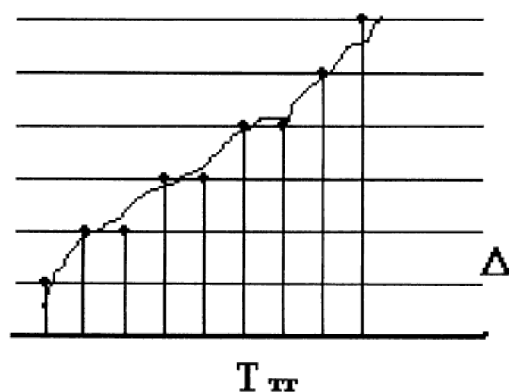


Рис. 2.5 . Представление аналогового сигнала дискретной последовательностью.

Единицы измерения значения не имеют. Это может быть напряжение на электродах в мВ или эквивалентное отклонение пера самописца, выраженное в миллиметрах. Диапазон регистрируемых сигналов серийного кардиографа колеблется от 0.03 до 5 мВ, эффективная ширина записи канала - 40 мм. При наиболее часто устанавливаемой чувствительности 10 мм/мВ максимальный регистрируемый сигнал составляет 4 мВ, что соответствует 40 мм на бумажной ленте. Эти величины логично принять в качестве  $U_{max}$ .

На графике (рис. 2.6), иллюстрирующем зависимость  $e(U)$  отмечены некоторые характерные точки. Видно, что во всём рабочем диапазоне входных сигналов относительная ошибка квантования, вносимая АЦП, не превосходит 6.5%, в то время как паспортная погрешность измерения напряжения аналогового ЭКГ-сигнала (табл. 2.4) составляет не более 20% в диапазоне от 0.1 до 0.5 мВА и 10% в диапазоне от 0.5 до 4.0 мВ.

Генератор тактовых импульсов, управляющий устройством выборки и хранения (УВХ), стабилизирован кварцевым резонатором, поэтому его относительная нестабильность частоты оценивается как 10 в минус четвертой степени %. Относительная же погрешность скорости движения ленты аналогового электрокардиографа по паспорту составляет 5%.

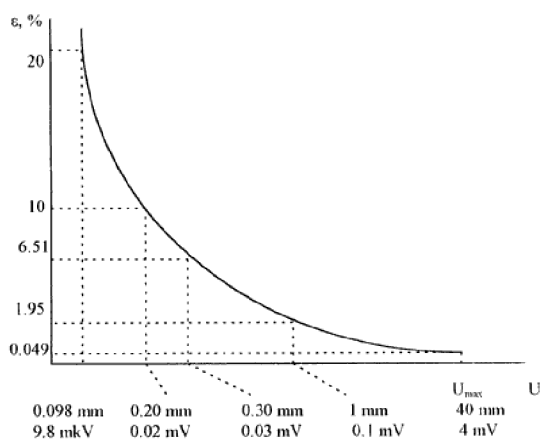


Рис. 2.6. Зависимость относительной ошибки измерения напряжения от амплитуды входного сигнала.

Таблица 2.4.

Сравнение разрешающей способности серийного и компьютерного электрокардиографов.

Относительная погрешность измерения амплитуды сигнала	Электрокардиограф	
	Серийный	Компьютерный
Рабочий диапазон от 0.03 до 4 mV	—	< ±6.5%
от 0.1 до 0.5 mV	< ±20%	< ±2%
от 0.5 до 4 mV	< ±10%	< ±0.4%
Разрешающая способность по времени	8 ms	60 mks
Относительная погрешность измерения временных интервалов в диапазоне от 0.1 до 1 s	< ±10%	< ±0.1%

Оценка разрешающей способности прибора по времени, если принять следующие допущения: разрешающая способность человеческого глаза при анализе ЭКГ, записанной на бумаге, равна толщине линии самописца; разрешающая способность при анализе ЭКГ с экрана монитора равна одному пикселю (один дискретный отсчёт). Толщина линии самописца равна 0.2 мм.

При скорости протяжки ленты равной 50 мм/с на бумаге записывается  $5 \cdot 10 / 0.2 = 250$  "толщин", в то же самое время в память компьютера записывается

1200 отсчётов ЭКГ. Таким образом, представление ЭКГ на экране монитора эквивалентно увеличению скорости протяжки ленты в 4.8 раза.

Таким образом, погрешность измерения амплитуды при помощи компьютерного кардиографа по сравнению с аналоговым меньше в 10-25 раз, в зависимости от уровня входного сигнала; разрешающая способность по времени у компьютерного варианта в 4,8 раза больше, чем у аналогового, относительная погрешность измерения временных интервалов у компьютерного меньше в 100 раз.

Однако погрешность вносится не только аппаратной частью. Как уже отмечалось, цифровой сигнал обрабатывать проще, чем аналоговый. Для обработки аналогового сигнала необходимо изготавливать реальные приборы "в железе". Каждая операция предполагает использование нового прибора. Такая "жесткая" структура оказывается громоздкой, дорогой, трудно поддающейся изменению.

Таким образом, вышеприведенный материал позволяет сделать следующие заключения:

1. Основные погрешности аналоговой регистрации ЭКГ значительно выше, чем при компьютерной регистрации.
2. Компьютерная регистрация позволяет применить современные технологии обработки ЭКГ и получить большое количество новых характеристик (спектр ЭКГ, фазовый портрет сердечных сокращений и другие данные).
3. Компьютерная регистрация ЭКГ позволяет избежать искажений, которые неизбежно возникают при сетевых наводках (50, 100 Гц) и др.
4. Компьютерная регистрация ЭКГ позволяет легко автоматизировать процесс обработки электрокардиограмм, что существенно, как для ускорения получения информации, так и при отсутствии высококвалифицированных кардиологов.
5. Разработанная при эксперименте методика легко позволяет перевести все имеющиеся в нашем здравоохранении аналоговые электрокардиографы в

компьютерные по цене в 10-25 раз ниже стоимости зарубежных компьютерных электрокардиографов.

## **Выводы по главе I**

1. Проведен анализ типов биологических сигналов в соответствии с их физической природой, систематизированы динамические характеристики основных физиологических сигналов человека.

2. Определены основные этапы проектирования программного обеспечения биомедицинских компьютерных систем, приведена обобщенная схема биомедицинской системы с устройствами регистрации, визуализации, хранения данных и выдачи результатов анализа.

3. Сформулированы требования к современным комплексам сбора и математической обработки данных электрокардиограмм, в том числе по архитектурным и функциональным признакам.

4. Рассмотрены этапы проведения исследований при проектировании программных систем, начиная от этапа изучения проблемы и выбора базисного метода исследования до формирования программной архитектуры и реализации. Дана концептуальная модель полиморфической системы регистрации и анализа биомедицинской информации, даны показатели качества при оптимизации архитектуры таких систем.

5. Освещены системы отведений, используемые в практике снятия ЭКГ-сигналов, и компоненты нормальной кардиограммы. Подробно проанализированы разработанные экспертные системы по анализу ЭКГ, основанных на определении соотношений её графических параметров, а также принципы проектирования программного обеспечения экспертных систем. Предоставлен сравнительный анализ погрешностей в аналоговой и компьютерной электрокардиографии. Сделан вывод о неоспоримых преимуществах компьютерной регистрации.

6. Детально проанализированы характеристики электрокардиограмм с учетом принятых на практике стандартных отведений внешних электродов. Даны числовые значения длительности интервалов сердечного цикла в зависимости от частоты сердечных сокращений.

7. Изучены основные компоненты электрокардиограмм и их нормальные величины с целью последующего создания алгоритмов обработки.

8. Поставлена задача частотно-временного анализа ЭКГ и фильтрации с применением фильтров нижних частот с использованием компьютерной техники.

## ГЛАВА 2. МЕТОДЫ И АЛГОРИТМЫ ФИЛЬТРАЦИИ СИГНАЛОВ ЭКГ

### 1. Характеристики помех при регистрации электрокардиограмм и методы их устранения

При анализе электрокардиограмм очень важное значение имеет обнаружение и устранение помех, чтобы получить правильный диагноз заболеваний. Получение электрокардиограмм зависит от усилительного канала электрокардиографа.

Усилительная система в электрокардиографе способствует резкому усилению не только полезных сигналов, но и тех незначительных помех, которые не всегда устранимы.

В основном, причиной помех могут быть электрическая активность тканей, через которые проводится импульс (например, скелетные мышцы), сопротивление тканей, особенно кожи, а также сопротивление на входе усилителя. Примером помех такого рода является электрическая активность скелетных мышц, поэтому при регистрации электрокардиограммы необходимо рекомендовать пациенту максимально расслабить мышцы. Мышечные токи накладываются на электрокардиограмму у больных с дрожательным параличом, хореей, тетанией, паркинсонизмом, тиреотоксикозом. Колебания, вызываемые мышечными токами, иногда трудно отличить от трепетания предсердий. Артефакты, возникающие на кривой при случайном толчке аппарата или кушетки могут имитировать желудочковые экстрасистолы. Однако, при внимательном рассмотрении артефакты легко распознаются. Неравномерная работа отметчика времени или лентопротяжного механизма может симулировать аритмию.

При сопоставлении динамических изменений нельзя придавать диагностическое значение изменениям амплитуды зубцов, если серийные электрокардиограммы у одного и того же пациента зарегистрированы при разной чувствительности электрокардиографа.

Большое значение имеет постоянство нулевой (или основной) линии, от

которой производится отсчет амплитуды зубцов. Стабильность нулевой линии зависит от наличия достаточно высокого входного сопротивления усилительной системы и минимального кожного сопротивления.

Нередко основная линия электрокардиограммы колеблется вместе с элементами кривой. Подобную электрокардиограмму не следует считать патологической, так как причиной могут быть нарушения режима питания аппарата, форсированное дыхание пациента, кашель, икота, чиханье, перистальтика кишечника. В грудных отведениях подобные изменения нередко проявляются при трении электрода о выступающие ребра.

Низкий вольтаж зубцов иногда обуславливается плохим контактом электродов с кожей. Значительные помехи вызывают наводные токи («фон»), распознаваемые по правильности колебаний 50 Гц (от осветительной сети). Подобные помехи могут появиться при плохом контакте электродов с кожей, особенно при ее волосатости. Нетрудно распознать локализацию возникновения помех. Например, если «наводка» вида во II и III отведении, а в I отведении ее нет, то провод от левой ноги имеет плохой контакт с электродом, или последний неплотно прилегает к коже. Если «наводка» видна в I и II отведении, то плохой контакт на правой руке. Если контакты достаточные, а «наводка» наблюдается во всех отведениях, то рекомендуется заземлить правую ногу, присоединив ее специальным кабелем к отопительным и водопроводным трубам. Иногда полезно изменить положение койки пациента, поскольку тело человека иногда можно уподобить антенне, которая в одних положениях подвержена влиянию атмосферных токов, а в других — свободна от них. Кроме того, нельзя укладывать пациента так, чтобы осветительная сеть была параллельна оси отведения. Для устранения «наводки» часто прибегают к различным фильтрам. Такого способа устранения помех следует избегать, так как при этом наряду с освобождением кривой от наводных токов нередко выбрасываются и частоты сигнала сердца. Для анализа и обработки сигналов ЭКГ имеется множество методов и алгоритмов. Некоторые из них исследуются в следующем параграфе.

## **2. Методы идентификации с использованием калибровочных и синтезируемых сигналов**

При возбуждении и реполяризации сердца возникает электрическое поле, которое можно зарегистрировать на поверхности тела. При этом между различными точками создается разность потенциалов, изменяющаяся в соответствии с колебаниями величины и направления этого электрического поля. Эти разности потенциалов, изменяющиеся во времени, являются электрокардиосигналами

Под идентификацией сигналов электрокардиограмм понимается сравнительный анализ «идеальных» сигналов здоровых пациентов и искаженных сигналов больных. Для качественной оценки результатов сравнительного анализа и подтверждения правильности идентификации используют разные методы и приемы. Это может быть генерация калибровочного сигнала с идеальными характеристиками (имитатора), с которым сравнивается исследуемый неизвестный сигнал ЭКГ. Это также может быть синтезируемый с помощью численных методов сигнал, математическая модель которого наиболее полно удовлетворяет поставленной цели.

Для реализации генерирования кардиосигналов имитаторами наиболее подходит алгоритм, основанный на линейной и квадратичной интерполяции участков кардиограмм. В рамках этого алгоритма сигнал задается в виде последовательности отрезков и участков парабол, называемые далее элементами ЭКГ. С целью описания изменения формы и очередности появления элементов введен набор систем координат (СК). Начало набора СК (рис. 2.1) совпадает с началом текущего кардиоцикла и последовательно перемещается при возникновении следующих кардиоциклов. Момент времени первого отклонения какого-либо сигнала от собственной изоэлектрической линии является началом набора СК в текущем кардиоцикле. Изоэлектрическая линия ЭКГ является осью абсцисс, соответствующей этому сигналу СК. Таким

способом формируется перемещаемый набор СК циклов кардиосигналов, относительно которых определяются координаты исходных и конечных точек элементов ЭКГ.

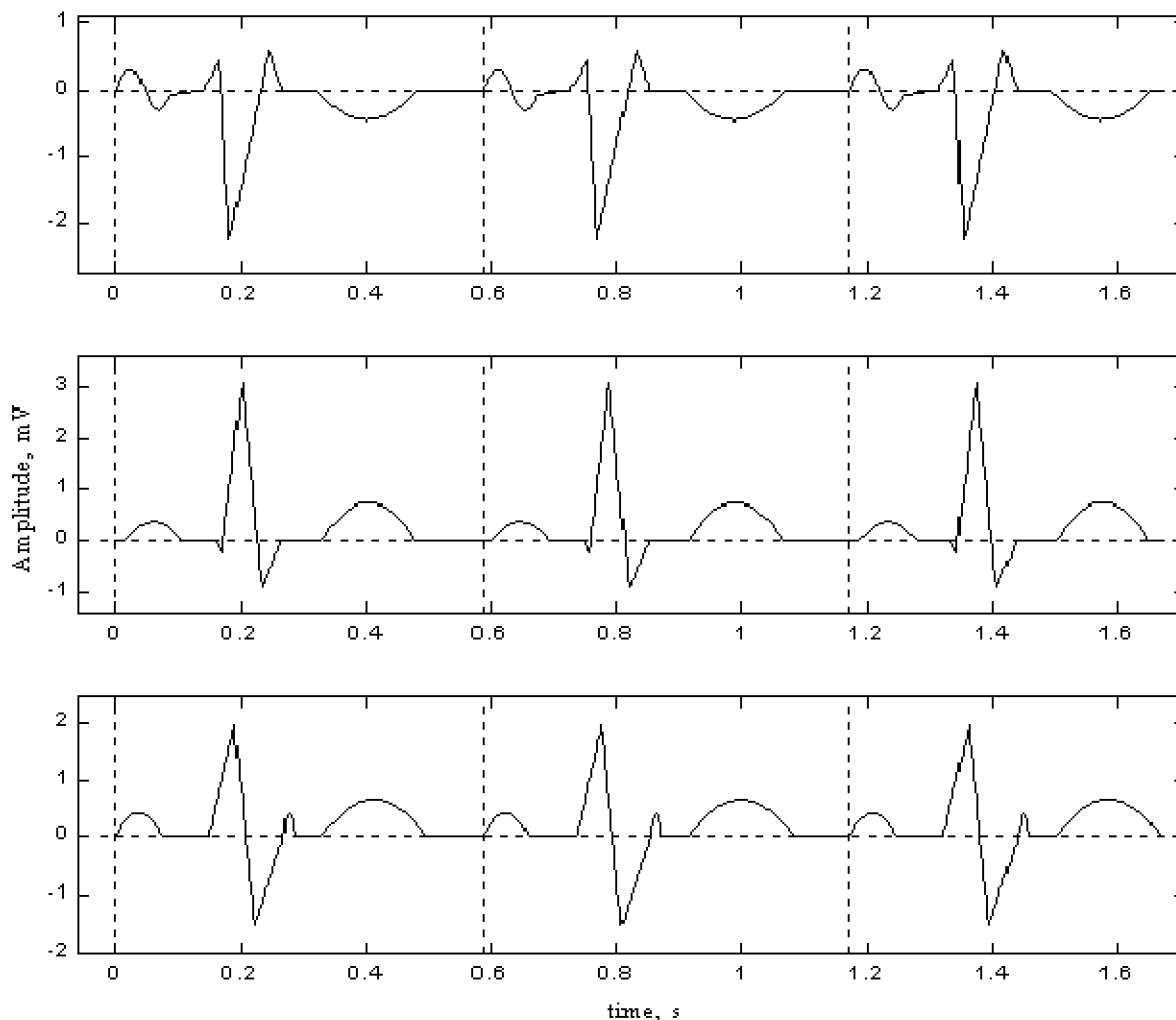


Рис.2.1. Синтезированные кардиосигналы с набором координатных систем (пунктирные линии).

Каждый  $i$ -й элемент сигнала кодируется шестью параметрами  $a_i$ ,  $b_i$ ,  $x_i$ ,  $y_i$ ,  $x_{i+1}$ ,  $y_{i+1}$ . Параметры  $x_i$ ,  $y_i$  и  $x_{i+1}$ ,  $y_{i+1}$  задают исходную и конечную координату элемента кардиосигнала. В случае элемента параболы  $a_i$  (ордината) и  $b_i$  (абсцисса) определяют её экстремум. В случае элемента отрезка  $a_i$  и  $b_i$  полагаются равными нулю. Тогда уравнения элементов кардиосигналов записываются в следующем виде:

$$y = \frac{y_i - a_i}{(x_i - b_i)^2} (x - b_i)^2 + a_i \quad (2.1)$$

для параболы и

$$y = y_i + \frac{y_{i+1} - y_i}{x_{i+1} - x_i} (x - x_i) \quad (2.2)$$

для отрезка,

где  $x_i \leq x \leq x_{i+1}$ .

Шесть параметров каждого элемента кардиосигнала можно определять двумя способами. Первый способ заключается в том, что параметры элементов кардиосигнала задаются абстрактно от реального сигнала. С целью предотвращения участков изломов между соседними элементами предлагается следующая процедура. Требуется выполнение условия

$$y_{i+1} = \frac{y_i - a_i}{(x_i - b_i)^2} (x_{i+1} - b_i)^2 + a_i \quad (2.3)$$

где  $x_i < b_i < x_{i+1}$ ,  $|a_i| > \{|y_i|, |y_{i+1}|\}$ ,  $a_i$  - заданная ордината экстремума параболы. Данные условия накладывают ограничения на форму искомого участка параболы, который должен проходить через заданную конечную координату и содержать точку экстремума. Тогда, решая уравнение (2.3) относительно  $b_i$ , получаем следующее соотношение:

$$b_i = x_i + (x_{i+1} - x_i) \frac{1 - \sqrt{H}}{1 - H},$$

где  $H = \frac{y_{i+1} - a_i}{y_i - a_i}$ . Если  $H=1$  ( $y_{i+1}=y_i$ ), то  $b_i=0.5(x_{i+1}+x_i)$ .

Исходными данными для вычисления параметров элементов кардиосигнала первым способом является последовательность триплетов, каждый из которых содержит следующие числа:

1.  $a_i$  со значением равным нулю, если элемент - отрезок, и со значением ординаты экстремума параболы, если элемент - парабола,
2.  $x_{i+1}, y_{i+1}$  конечная координата  $i$ -го элемента.

Конечная координата предыдущего ( $i-1$ )-го элемента является исходной координатой  $i$ -го, т.е.  $x_i$  и  $y_i$ . Таким образом, вычисление параметров элементов кардиосигнала данным способом включает три этапа:

1. Задание желаемых исходных и конечных координат элементов, значений ординат экстремумов парабол и формирование последовательности триплетов.
2. Вычисление для каждого элемента параболы абсциссы её экстремума по приведенной выше формуле.
3. Формирование последовательности наборов параметров  $(a_i, b_i, x_i, y_i, x_{i+1}, y_{i+1})$  элементов кардиосигнала.

На рис. 2.2. показан пример синтезированных калибровочных кардиосигналов с помощью описанного способа на основе заданных последовательностей триплетов

$$ecs=[a_1, x_1, y_1; \dots a_i, x_i, y_i; \dots a_n, x_n, y_n]:$$

$$ecs1=[2,4.5,0;-2,8,-0.5;0,13,0;0,15.5,3;0,17,-15;0;23,4;0,25,0;0;30,0;-3,45,0;0,47,0],$$

$$ecs2=[0,1.5,0;2.5,10,0;0,15,0;0,16,-1.5;0,19,20.5;0,22,-6;0,25,0;0,31,0;5,45,0;0,47,0],$$

$$ecs3=[3,7,0;0,14,0;0,18,13;0,21,-10;0,25.25,0;3,27,0;0,31,0;4.5,47,0].$$

Графики синтезированных кривых отображены с масштабными коэффициентами 0.011 и 0.15 соответственно по осям абсцисс и ординат, а также со значением интервала между кардиоциклами, равного 0.605.

Второй способ предполагает использование реальных сигналов ЭКГ. Из этих сигналов удаляют сетевую помеху, диагностически незначимые колебания с помощью полосового фильтра с частотами среза 0.05 и 65.5 Гц, а также другие артефакты. С помощью программы находят исходные и конечные координаты элементов кардиосигналов. Поскольку не все программы

автоматизированного измерения кардиограмм точно и корректно производят анализ кривых (определение моментов точек перегиба), то такие программы должны реализовывать функцию "ручного" обмера и корректировки измеренных величин. Вычислению подлежат параметры  $a_i$ ,  $b_i$  параболического элемента, поскольку исходными и конечными координатами элементов отрезка и параболы являются точки, полученные на начальном этапе определения изломов кардиосигналов.

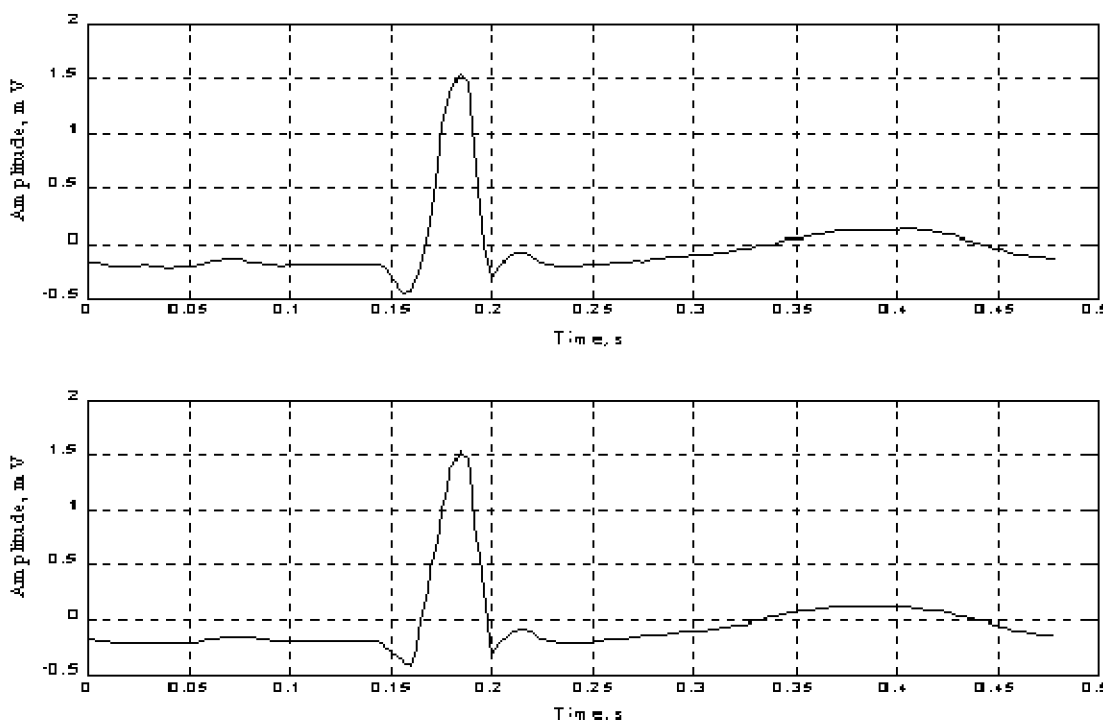


Рис. 2.2. Графики реального кардиоцикла (верхний) и его синтезированного аналога (нижний).

Пусть требуется найти уравнение параболического элемента кардиосигнала  $y = a * x^2 + b * x + c$  на заданном множестве точек  $(x_1, y_1), \dots, (x_k, y_k), \dots, (x_n, y_n)$ . Преобразуем это множество точек к виду  $(t_1, y_1), \dots, (t_k, y_k), \dots, (t_n, y_n)$ , где  $t_k = x_k - x_1$ . Для того, чтобы избежать появления участков изломов между соседними элементами и уменьшить влияние остаточного шума в исходных данных  $y_k$ , параметры параболы будем находить по методу наименьших квадратов с краевыми условиями  $y(t_1) = y_1$  и  $y(t_n) = y_n$ .

$$\begin{cases} y_1 = a^* t_1^2 + b^* t_1 + c^* & (2.4) \\ y_n = a^* t_n^2 + b^* t_n + c^* & (2.5) \\ F(a^*, b^*, c^*) = \sum_{k=1}^n (y_k - a^* t_k^2 - b^* t_k - c^*)^2 \rightarrow \min & (2.6) \end{cases}$$

Из (2.4) и (2.5) имеем,

$$c^* = y_1, \quad b^* = \frac{y_n - y_1}{t_n} - a^* t_n.$$

Подставляя  $c^*$  и  $b^*$  в (2.6), получаем

$$F(a^*) = \sum_{k=1}^n (y_k - a^* (t_k^2 - t_n t_k) - t_k \frac{y_n - y_1}{t_n} - y_1)^2.$$

Приравнявая к нулю первую производную функции  $F(a^*)$ , находим из полученного уравнения параметр  $a^*$  в следующем виде:

$$a^* = \frac{\sum_{k=1}^n (y_k - t_k \frac{y_n - y_1}{t_n} - y_1)(t_k^2 - t_n t_k)}{\sum_{k=1}^n (t_k^2 - t_n t_k)^2}. \quad (2.7)$$

Тогда остальные параметры будут выражаться следующими уравнениями:

$$b^* = \frac{y_n - y_1}{t_n} - \frac{\sum_{k=1}^n (t_n (y_k - y_1) - t_k (y_n - y_1))(t_k^2 - t_n t_k)}{\sum_{k=1}^n (t_k^2 - t_n t_k)^2}, \quad (2.8)$$

$$c^* = y_1. \quad (2.9)$$

В итоге имеем уравнение параболического элемента  $y = a^*(x-x_1)^2 + b^*(x-x_1) + c^*$ , где  $x_1 \leq x \leq x_n$ . Сопоставляя его с уравнением (2.1), находим следующие соотношения:

$$a_i = c^* - \frac{b^{*2}}{4a^*}, \quad b_i = x_i - \frac{b^*}{2a^*}. \quad (2.10)$$

Описанный выше способ представления кардиосигналов удобно использовать на предварительном этапе подготовки данных для последующего синтеза кардисигналов имитаторами. Представленные уравнения (2.1) и (2.2) можно упростить, т.е. исключить ресурсоемкую для микроконтроллеров операцию деления и умножения-деления и уменьшить число параметров элементов кардиосигналов. А именно, введем новый параметр

$$q_i = \frac{y_i - a_i}{(x_i - b_i)^2} \text{ для парабол и}$$

$$q_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \text{ для отрезков.}$$

Производя замену переменных  $t = x - x_i$ , уменьшаем число параметров элемента кардиосигналов. Итоговые уравнения записываем в следующем виде:  $y = q_i(t - p_i)^2 + r_i$ ,  $p_i = x_i - b_i$ ,  $r_i = a_i$  для парабол и  $y = q_i t + r_i$ ,  $r_i = y_i$ ,  $p_i = 0$  для отрезков, где  $0 \leq t \leq s_i = x_{i+1} - x_i$ . Таким образом, каждый элемент кардиосигнала в имитаторе определяется четырьмя параметрами  $(0, q_i, r_i, s_i)$  для отрезка и  $(p_i, q_i, r_i, s_i)$  для параболы.

Приведем результат работы второго способа нахождения параметров элементов ЭКС в виде сравнения графиков реального кардиоцикла и его синтезированного аналога, которые представлены на рис.3.2. Реальный кардиоцикл был выделен из кардиосигнала, полученного с 16-ти разрядной точностью и с частотой дискретизацией 250 Гц. Сигнал пропущен через полосовой фильтр с диапазоном от 0.05 до 65.5 Гц с предварительным удалением сетевой помехи. С помощью программы, реализующей только "ручной" обмер кардиосигнала, определялись исходные и конечные координаты предполагаемых элементов выделенного цикла. После указания типа элемента кардиосигнала, который должен находиться между выделенными соседними координатами, формировался массив триплетов  $[k_i, x_i, y_i, \dots, k_n, x_n, y_n]$ . В этом массиве  $(x_i, y_i)$  является конечной координатой  $i$ -го элемента, а  $k_i$  равно единице, если  $i$ -ый элемент - парабола, в

противном случае равно нулю. Подавая на вход расчетной программе исходный сигнал и полученный массив триплетов, эта программа автоматически находила участки интерполяции и вычисляла точки экстремумов параболических элементов по формулам (2.7) - (2.10). После формирования массива параметров  $a_i, b_i, x_i, y_i, x_{i+1}, y_{i+1}$  элементов кардиосигналов по формулам (2.1) - (2.2) строился график синтезированного кардиоцикла с частотой дискретизации 500 Гц. Общее число использованных элементов составило десять, из которых 6 участки парабол и 4 отрезка.

Описанный выше способ представления сигналов кардиограмм удобно использовать на предварительном этапе подготовки данных для последующего синтеза кардиосигналов имитаторами. Представленные уравнения (2.1) и (2.2) можно упростить. А именно, введем новый параметр

$$q_i = \frac{y_i - a_i}{(x_i - b_i)^2} \text{ для парабол и}$$

$$q_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \text{ для отрезков.}$$

Производя замену переменных  $t=x-x_i$ , уменьшаем число параметров элемента ЭКС. Итоговые уравнения записываем в следующем виде:  $y=q_i(t-p_i)^2$ ,  $p_i=x_i-b_i$ ,  $r_i=a_i$  для парабол и  $y=q_i t+r_i$ ,  $r_i=y_i$ ,  $p_i=0$  для отрезков, где  $0 \leq t \leq s_i = x_{i+1} - x_i$ . Таким образом, каждый элемент кардиосигнала в имитаторе определяется четырьмя параметрами  $(0, q_i, r_i, s_i)$  для отрезка и  $(p_i, q_i, r_i, s_i)$  для параболы.

В этом алгоритме полуволны ЭКС описываются как участки парабол. В связи с этим возникает вопрос. Почему использованы параболы, а не синусоиды? Отвечая на этот вопрос, можно привести два довода в пользу выбора квадратичного описания полуволн кардиосигналов. Во-первых, существующие алгоритмы расшифровки электрокардиограмм строятся на концепции пиков кривых. В этой концепции отсутствует предположение о природе пика полуволны, а именно, принадлежность его к гипотетической параболе, синусоиде или другой кривой. Поэтому выбор уравнения полуволн в данном контексте может быть произвольным. Во-вторых, вычисление значений

синусоиды более ресурсоемкая операция в сравнении с вычислением значений парабол.

Приведем оценку эффективности описанного алгоритма синтеза кардиосигнала в сравнении с методом непосредственного воспроизведения дискретных значений реальных кардиосигналов. Средняя длительность кардиоцикла нормального ритма составляет 0.5 с, который состоит из 8 элементов. Международная Электротехническая Комиссия рекомендует использовать 16-ти разрядные значения тестовых сигналов, дискретизированные с частотой 1000 Гц. В этом алгоритме каждое значение элемента кардиосигнала представляется четырьмя байтами действительного числа. Следовательно, для хранения одного кардиоцикла в алгоритме синтеза требуется  $8 \times 4 \times 4 = 128$  байт, для хранения того же кардиоцикла в методе непосредственного воспроизведения требуется  $0.5 \times 1000 \times 2 = 1000$  байт. Эффективность алгоритма синтеза кардиосигнала в сравнении с методом непосредственного воспроизведения по требуемому объему памяти имитатора очевидна. Отметим также дополнительное преимущество алгоритма синтеза кардиосигнала. В имитаторе после цифро-аналогового преобразования тестовые сигналы должны быть пропущены через фильтр нижних частот. Порядок фильтра зависит от разрядности цифро-аналогового преобразователя и отношения частотного диапазона сигнала к его частоте дискретизации. Чем больше эти величины, тем больший порядок фильтра требуется, что приводит к повышению стоимости прибора и трудности его настройки. С одной стороны, метод непосредственного воспроизведения оцифрованных значений кардиосигнала не позволяет задавать произвольные разрядность и частоту дискретизации сигнала, не прибегая к начальной интерполяции этих сигналов. С другой стороны, посредством оптимального выбора разрядности и частоты дискретизации приведенных уравнений элементов ЭКС возможно использовать в имитаторах НЧ фильтры не более 3-го порядка при сохранении приемлемого качества синтезированных кардиосигналов. Следовательно, эффективность алгоритма синтеза в сравнении с методом непосредственного воспроизведения

значений кардиосигналов проявляется в снижении требований в реализации выходных каскадов имитаторов.

### 3. Фильтрация сигналов ЭКГ на основе спектральных функций

Предварительная обработка данных является одной из актуальных задач, возникающих при построении информационных технологий обработки сигналов различной физической природы [31]. В первую очередь эта задача связана с выделением полезного сигнала на фоне помех: требуется по сигналу  $y[n]=f(x[n],\varepsilon[n])$ , наблюдаемому в дискретные моменты времени  $n=1,\dots,N$ , отделить полезный сигнал  $x[n]$  от искажающей его помехи  $\varepsilon[n]$ .

Понятно, что такая задача имеет решение только в том случае, когда априори известна либо сделаны обоснованные допущения о функции  $f(\cdot)$ . Довольно часто предполагается, что помеха аддитивна, т.е.  $y[n]=x[n]+\varepsilon[n]$  [32]. Но даже при таких предположениях, возможно, отделить полезный сигнал от помехи, если имеются различия в их характеристиках (частотных, вероятностных и т.п.).

В тех случаях, когда модель полезного сигнала  $x[n]$  может быть описана с точностью до небольшого числа параметров, решение задачи сводится к оценке этих параметров по наблюдаемой реализации. Такой подход возможен при обработке зашумленных сигналов, порождаемых в процессе функционирования достаточно простых технических систем.

В то же время часто модель полезного сигнала либо неизвестна, либо достаточно сложна, чтобы можно было свести задачу отделения полезного сигнала от помехи к оценке небольшого числа параметров. Такая ситуация характерна при обработке электрокардиограмм и других сигналов сложной структуры, порожденных живой природой, а не технической системой.

В таких ситуациях не остается ничего иного, как ввести адекватные предположения о модели помехи и строить различные фильтры,

обеспечивающие максимально возможное удаление помехи при минимальных искажениях полезного сигнала.

В этом разделе исследуется один из возможных подходов к решению задачи фильтрации частотных помех, основанный на прямом и обратном дискретном преобразовании Фурье (ДПФ).

**Постановка задачи.** Пусть на вход системы предварительной обработки данных поступает сигнал  $y[n]$ , который измерен в дискретные моменты времени  $n=1,2,\dots,N$  и содержит информацию о полезном сигнале  $x[n]$ , зашумленном аддитивной помехой  $\varepsilon[n]$

$$y[n]=x[n]+\varepsilon [n] \quad (2.11)$$

Будем полагать, что помеха  $\varepsilon$  имеет частотную природу и может быть представлена суммой стационарных гармонических колебаний

$$\varepsilon[n]=\sum_{i=1}^M a_i \sin(2\pi f_i n + \varphi_i) \quad (2.12)$$

с априори известными частотами  $f_i$  и неизвестными амплитудами  $a_i$  и фазами  $\varphi_i$  [47].

Традиционно для цифровой фильтрации таких сигналов используют различные фильтры с бесконечной или конечной импульсной характеристикой, в частности, широко известный фильтр Баттерворта [33,34]. Основным их преимуществом является возможность обработки сигнала в реальном масштабе времени по мере поступления очередного дискретного отсчета  $y[n]$ .

В то же время всем таким фильтрам присущи следующие недостатки. Во-первых, они позволяют устойчиво фильтровать сигнал лишь после определенного промежутка времени, зависящего от параметров настройки фильтра, в частности, от его порядка. Во-вторых, даже при оптимальной настройке таких фильтров не удастся реализовать “идеальную” амплитудно-частотную характеристику (АЧХ), обеспечивающую полное подавление частот в полосе заграждения без искажения сигнала в полосе пропускания. Реально эти фильтры позволяют лишь аппроксимировать “идеальную” АЧХ прямоугольной формы с некоторой степенью точности. Особые трудности

синтеза таких фильтров вызывают практически важные случаи, когда требуется обеспечить узкополосную режекцию сигнала (меньше 0.1% частоты дискретизации).

Именно поэтому в данной работе исследуется альтернативный подход к решению задачи цифровой фильтрации, основанный на прямом и обратном дискретном преобразовании Фурье. Такой подход позволяет не только заграждать частоты в области 0-0.6 Гц, а реализовать частотный фильтр с заданными полосами пропускания и заграждения во всем диапазоне частот от 0 до частоты Найквиста с практически “идеальной” АЧХ прямоугольной формы. Более того, при определенных модификациях обеспечивается возможность подавления отдельных фиксированных гармоник, что особенно важно при решении ряда практических задач, когда гармоническая помеха находится в области информативных частот полезного сигнала.

#### **Алгоритм фильтрации на основе спектральных функций.**

Рассматриваемый подход основан на использовании алгоритмов прямого *FFT* обратного *IFFT* дискретного преобразования Фурье (ДПФ), описанных в работах [35,36]. Известно [37], что эти алгоритмы взаимно обратимы (с точностью до малых погрешностей округления), т.е. для любого вектора  $Z$  справедливо равенство

$$IFFT(FFT(Z))=Z \quad (2.13)$$

Отсюда, если в векторе, полученном после прямого ДПФ провести обнуление элементов, соответствующих требуемым полосам режекции, то после обратного ДПФ будет получен отфильтрованный сигнал (рис. 2.3).

Например, как видно из рис.2.3,а и 2.4,а в спектре ЭКГ присутствует сетевая наводка, которая проявляется как дребезжание линии ЭКГ с частотой около 50 и около 100 Гц.

При компьютерной обработке эта помеха легко устраняется, что облегчает интерпретацию и анализ кардиограмм (рис.2.4б). Под обработкой сигнала подразумевается фильтрация. Для обработки кардиографических

сигналов нами применялись программы - цифровые фильтры нижних частот и дифференциаторы. Попробуем оценить ошибку, вносимую этими фильтрами.

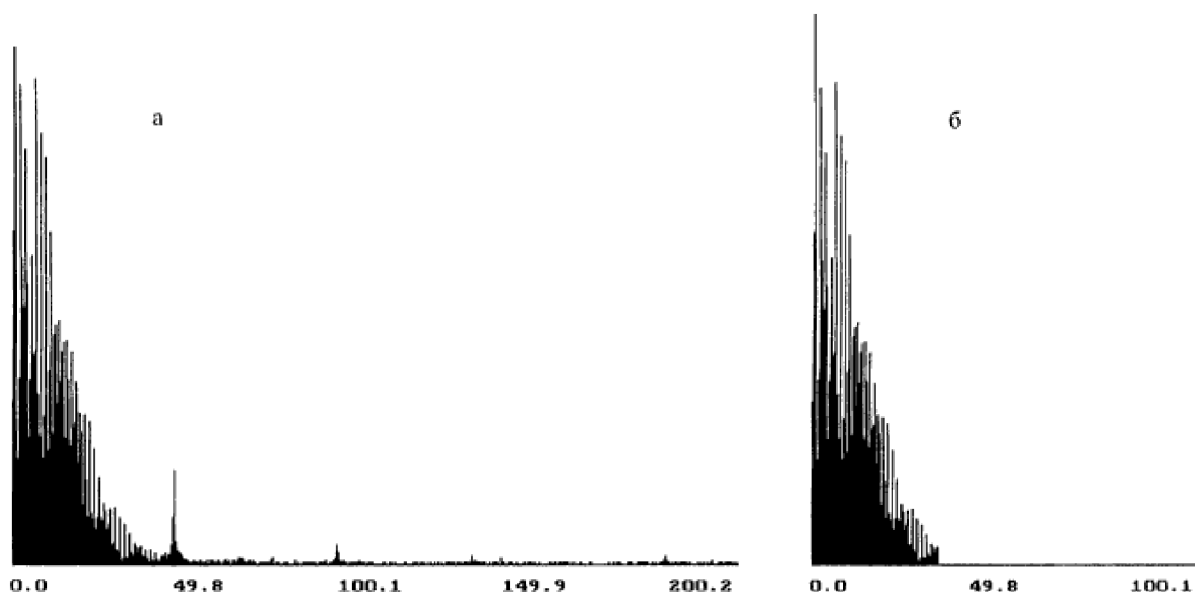


Рис.2.3. Спектр ЭКГ: ДПФ по 1024 точкам, где а - полный спектр; б - искусственно усечённый спектр.

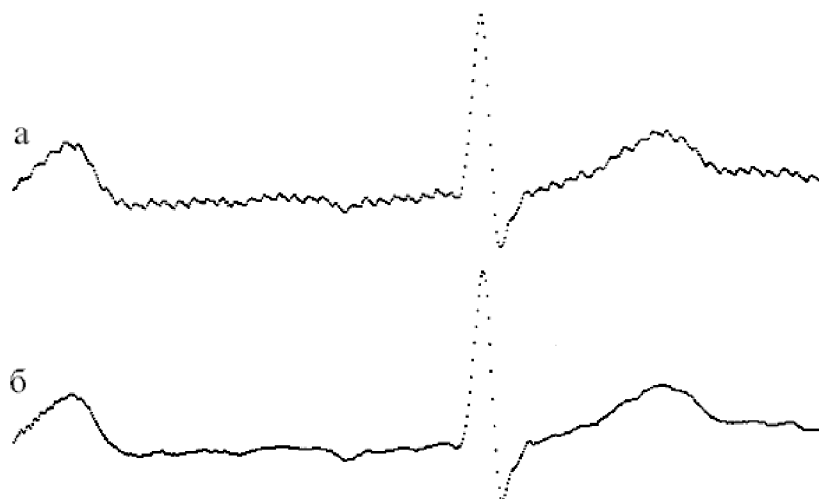


Рис.2.4. Фрагмент компьютерной электрокардиограммы (пример); а - неочищенная ЭКГ; б - очищенная ЭКГ.

Фильтрация - это уже само по себе преднамеренное изменение исходного сигнала с целью выделить из него полезную информацию. Цифровой сигнал можно подвергнуть дискретному преобразованию Фурье, то есть перевести его

из временной области в частотную. В этом представлении сигнал будет характеризоваться набором кратных частот - гармоник. Эти гармоники образуют спектр сигнала.

Спектр цифрового сигнала ограничен: первая гармоника зависит от длины цифрового ряда, ограничение на верхнюю гармонику накладывается частотой дискретизации в соответствии с теоремой Котельникова. Полезный спектр реального кардиосигнала ограничивается примерно частотой 50 Гц, тогда как при частоте дискретизации 1200 Гц теоретически частота сигнала может достигать 600 Гц, реально эта цифра должна быть уменьшена в 3-5 раз.

3-5-кратный запас берётся для уменьшения искажений, связанных с дискретизацией. Зная спектр сигнала можно восстановить сам сигнал. При восстановлении сигнала можно искусственно ограничить спектр, отбросив, занулив все гармоники выше некоей выбранной (рис.2.3). Теперь его граничная частота совпадёт с верхней используемой гармоникой. Описанный алгоритм эквивалентен низкочастотной фильтрации.

Причём, искусственно зануляя верхние гармоники мы не вносим искажений в низкочастотную часть, которая несёт основную часть информации. То есть получается как бы фильтр нижних частот с идеальной, прямоугольной частотной характеристикой.

Амплитудно-частотная характеристика идеального дифференциатора представляет собой наклонную прямую с началом в нуле координат, конец линии уходит на бесконечность, но при работе с сигналами с ограниченным спектром логично ограничить полосу пропускания фильтра частотой верхней гармоники (рис.2.5).

Шаг арифметической прогрессии или наклон АЧХ дифференциатора не играет роли, поскольку влияет лишь на амплитуду выходного сигнала, но не на его форму.

Однако на практике такой случай является скорее редким исключением, чем правилом. Отсюда следует, что если в общем случае для фильтрации гармонической помехи обнулять все гармоники, на которые распространилась

нагрузка помехи, то это может затронуть гармоники, несущие информацию о полезном сигнале, и в результате после обратного преобразования Фурье полезный сигнал может быть существенно искажен.

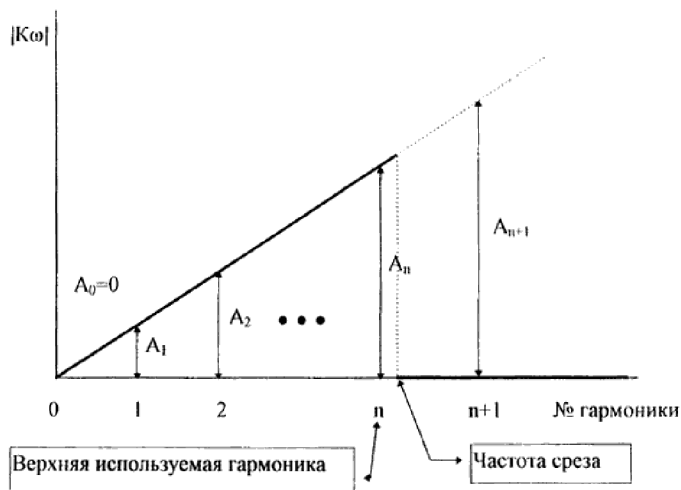


Рис.2.5. АЧХ идеального дифференциатора. Коэффициенты  $\{A_i\}$  образуют арифметическую прогрессию.

Для устранения отмеченного недостатка предложен модифицированный алгоритм частотно-избирательной фильтрации, блок схема которого представлена на рис.2.6.

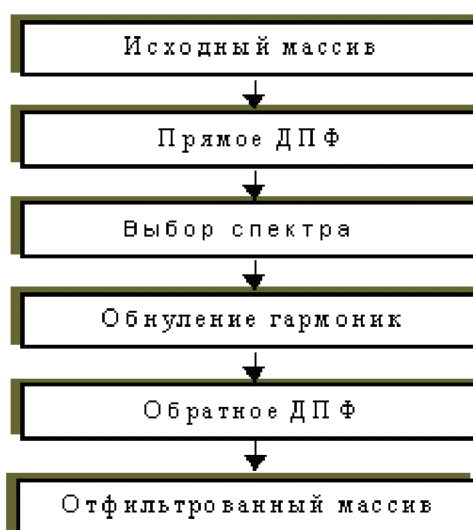


Рис.2.6. Блок схема алгоритма

В отличие от тривиального, алгоритм предусматривает процедуру модификации данных, поступающих на вход прямого ДПФ, обеспечивая эффект “выбор спектра” гармонической помехи. В результате после применения такой процедуры обнуление (либо снижение амплитуды) гармоник в области режекции и последующее обратное преобразование Фурье приводит к эффективному удалению помехи без заметных искажений полезного сигнала.

Для выбора спектра будем использовать специальный алгоритм, который представлен на рис.2.7.

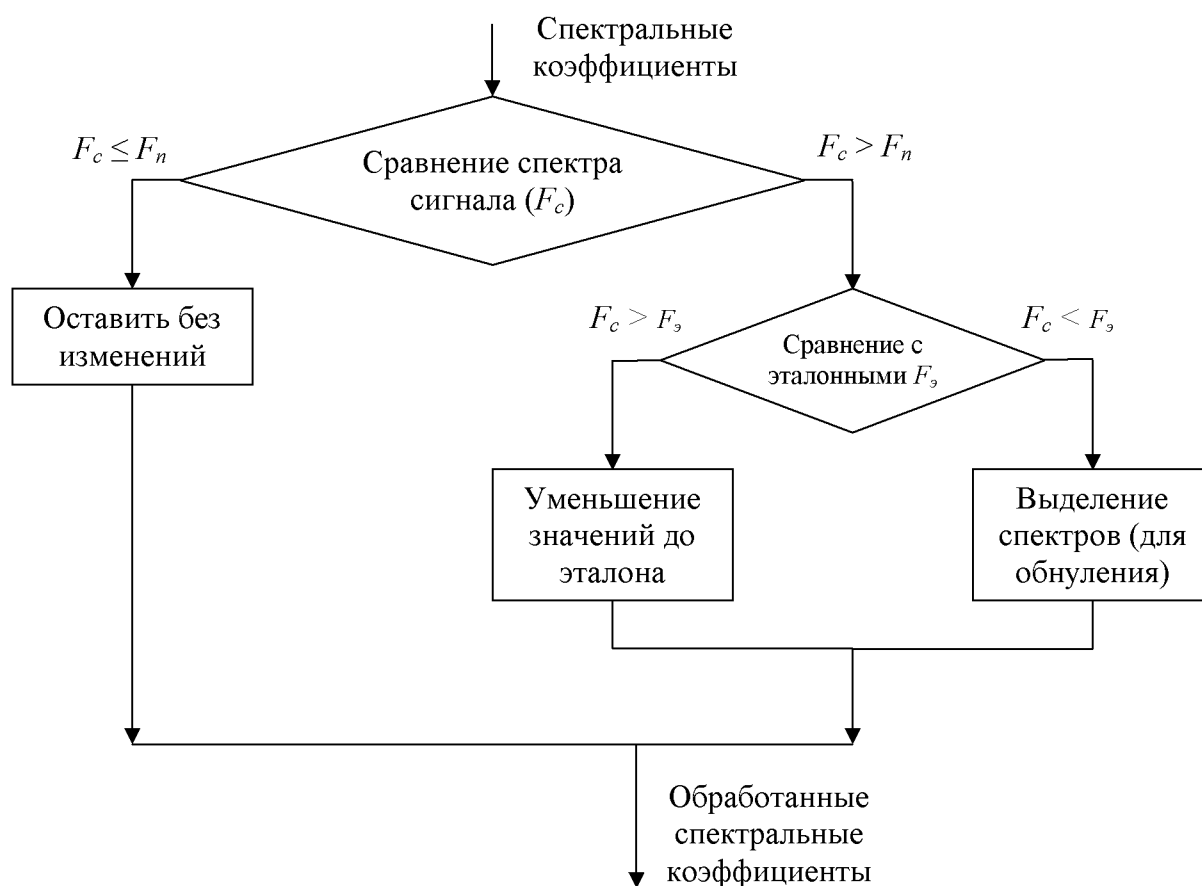


Рис.2.7. Блок схема алгоритма “выбор спектра”:  $F_c$  – спектральные коэффициенты входного сигнала,  $F_n$  – пороговые значение спектральных коэффициентов,  $F_s$  – эталонные значение спектральных коэффициентов ЭКГ сигнала.

Для оценки эффективности разработанных алгоритмов проводились эксперименты на модельных и реальных сигналах. Такая оценка выполнялась согласно схеме, показанной на рис.2.8.

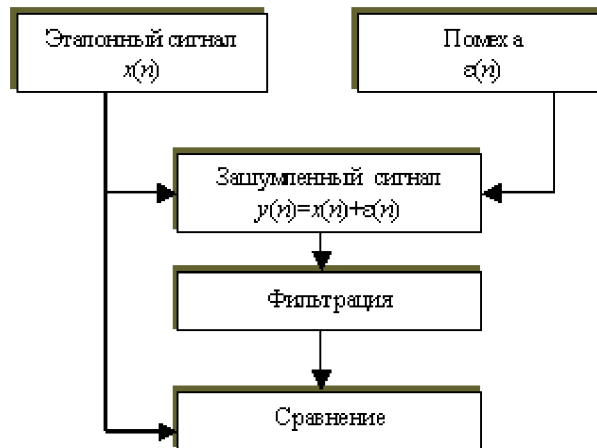


Рис.2.8. Схема эксперимента

**Фильтрация электрокардиограммы.** В данном эксперименте в качестве эталонного сигнала была использована реальная ЭКГ, зашумленная гармонической помехой в области 16.7 Гц. Такая частота характерна для ряда условий эксплуатации кардиографических систем, но, в отличие от 50 Гц, непосредственно расположена в области информативных частот полезного сигнала.

При эксперименте амплитуда помехи составляла 100% размаха эталонного сигнала. Сумма эталонной ЭКГ и помехи генерировалась в 30000 точек с частотой дискретизации  $F=1000$  Гц.

Сравнивались два способа фильтрации – без выбора спектра помехи и с выбором спектра помехи. Результаты тестирования (фрагменты сигналов) приведены на рис.2.9.

Как видно из рис.2.9, выбора спектра помехи в процессе фильтрации позволило избавиться от существенных искажений ЭКГ.

Дальнейшее улучшение качества обработки ЭКГ в реальной системе требуются применить специального алгоритма адаптивного сглаживания [38].

Проведенные эксперименты подтвердили эффективность фильтрации гармонических помех на основе прямого и обратного ДПФ на реальных и модельных данных. Метод позволяет реализовать узкополосный

заградительный фильтр в диапазоне от частот 0 до частоты Найквиста. При этом полоса режекции может составлять величину меньшую 0.1% частоты Найквиста, что существенно при обработке биомедицинских сигналов (например ЭКГ), когда гармоническая помеха располагается в области информативных частот полезного сигнала.

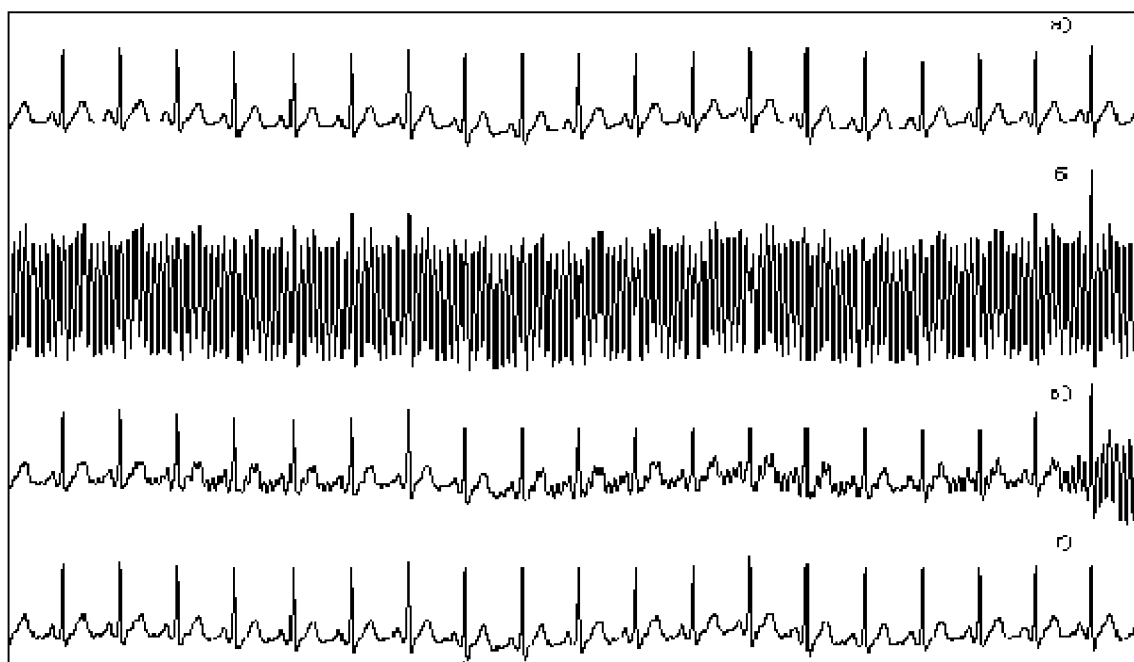


Рис.2.9. Результаты эксперимента: эталонная ЭКГ (а);  
зашумленная ЭКГ (б); результат фильтрации без выбора спектра (в) и с  
выбором спектра (г)

Эксперименты проводились с использованием компьютера Pentium IV с тактовой частотой 1.7 ГГц и оперативной памятью 512 Мб. При этом время фильтрации реальных ЭКГ, зарегистрированных в течение времени 30 сек с частотой дискретизации 1000 Гц, не превышало 2.1 миллисек., что вполне допустимо для практического применения.

## Выводы по главе II

В главе 2 рассматриваются вопросы фильтрации ЭКГ с точки зрения используемых методов и разработанных алгоритмов.

1. Анализированы виды помех, возникающих при регистрации ЭКГ, и методы их устранения, связанные с несовершенством измерительной и усилительной аппаратуры, несовершенстве самой методики проводимой регистрации сигналов.

2. Рассмотрен метод сравнительного анализа неизвестного входного сигнала ЭКГ с эталонным сигналом («идеальным» сигналом). Изучены два варианта генерации эталонного сигнала и его сравнения со входным неизвестным сигналом ЭКГ: вариант имитатора на базе сочетания линейных и квадратичных (параболических) участков записи ЭКГ и вариант формирования синтезированного анализа кардиосигнала на базе математической модели.

Первый вариант удобнее реализовать на предварительном этапе подготовки данных для получения грубой модели сигнала ЭКГ. По результатам предварительного этапа уточнена более приближенная математическая модель для синтеза аналога ЭКГ.

3. Даны оценки эффективности данного подхода идентификации в сравнении с методом, когда в качестве эталона выбирается сигнал ЭКГ здорового пациента.

4. Предложен метод и алгоритм низкочастотной фильтрации сигналов ЭКГ на основе ДПФ, дан алгоритм выбора полезных гармоник ЭКГ сигнала.

5. Проведенные эксперименты подтвердили эффективность фильтрации гармонических помех на основе прямого и обратного ДПФ на реальных и модельных данных. Метод позволяет реализовать узкополосный заградительный фильтр в диапазоне от частот 0 до частоты Найквиста. Описаны основные процедуры данного метода фильтрации.

## ГЛАВА 3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА ФИЛЬТРАЦИИ ЭКГ НА ОСНОВЕ СПЕКТРАЛЬНЫХ ФУНКЦИЙ

### 1. Программы фильтрации сигналов ЭКГ на основе спектрального подхода

Для создания программного пакета для фильтрации ЭКГ сигналов с помощью спектральных функций существует много объектно-ориентированных языков программирования, например: C, C++, Visual Basic, Delphi и Matlab. Была выбрана среда язык программирования Matlab, так как он является наиболее удобным для обработки сигналов, тем более, что в нем имеются встроенные процедуры спектрального анализа в виде готовых программных модулей.

Программный пакет предназначен для фильтрации зашумлённого ЭКГ на основе спектральных функций. Основа этого программного пакета, математические модели и алгоритмы, описаны в третьей главе. Основные характеристики программы приведены в таблице 3.1.

Таблица 3.1.

Основные характеристики

Наименование	Характеристика
ЭКГ сигнал	2 вида ЭКГ - сигналов: - зашумлённый; - фильтрованный
Помеха (шум)	возможности ручного управления помехами в ЭКГ
Фильтрация	возможности фильтрации на разном уровне помех в ЭКГ
Печать результатов	обеспечивается печать результатов фильтрации на разном уровне помех.

## Область применения

Данная программа может применяться для обучения: как помощник в практических занятиях для дисциплины «Цифровая обработка сигналов» и «Цифровая обработка сигналов и изображений» в ТУИТ и в медицинских ВУЗах.

## Общая характеристика входной и выходной информации

Основной входной информацией для нашей программы является зашумлённая ЭКГ. Наряду с полезным сигналом в ЭКГ присутствуют помехи разного уровня.

Выходной информацией для данной программы являются фильтрованные ЭКГ, в которых помехи исключены.

## Структура программного пакета

Основной программный пакет состоит из двух частей:

**nfilter.fig** – оболочка;

**nfilter.m** – исходной код, который создан на языке Matlab 7.

Для получения nfilter.exe файла на Matlab есть компилятор MATLAB Compiler. MATLAB Compiler компилирует \*.m файлы на \*.c (файлы, программированные на языке C) и автоматически создаёт \*.exe. Чтобы выполнить такую операцию, на командной строке Matlab даётся команда:

```
mcc -m -B sgl nfilter.m
```

После выполнения этой команды структура программного пакета будет в следующем виде:

**nfilter.exe** – оболочка, запускаемая пользователем при загрузке ПО, выполняет следующие функции:

- сохранение информации о сеансе работы (конфигурации);
- поддержка работы с внутренними базами данных;
- определение значений измеренных ЭКГ-параметров;
- печать данных и результатов;

- работа со справочной информацией;
- информация об авторах.

**nfilter.c** – исходные коды программы на программированном языке C;

<b>nfilter.h</b>	}	библиотека;
<b>libmatlbm.h</b>		
<b>libmmfile.h</b>		
<b>libmwsglm.h</b>		
<b>sgolayfilt.h</b>		
<b>subplot.h</b>		
<b>title.h</b>		

## 2. Порядок функционирования программы

### Термины и определения

Для работы с программой необходимо ознакомиться с терминами, используемыми в программе.

### Экран

*Экран* - совокупность данных, полностью занимающих экран компьютера. Данные относятся к одной теме или к одному виду действий.

### Окно

*Окно* - прямоугольная область экрана, содержащая функционально связанные данные, относящиеся к одной теме или одному виду действий.

## Поле ввода

*Поле ввода* - место на экране или в окне, предназначенное для ввода или корректировки данных. Каждое поле ввода имеет имя, показывающее, к какому параметру относится значение в поле ввода. Само поле ввода обычно ограничено квадратными скобками [ ]. Поля ввода бывают следующих видов:

- поля, обязательные для ввода;
- поля, необязательные для ввода;
- связанные поля ввода;
- поля ввода с прокруткой;
- поля ввода со связанным списком.

Если поле является обязательным для ввода, то для того чтобы перейти на следующее поле надо произвести ввод корректной информации в текущее поле. При попытке перехода на следующее поле без ввода данных или при вводе некорректных данных программа сигнализирует звуком об ошибке ввода и не выполняет перехода.

Поле, необязательное для ввода, позволяет перейти на следующее поле, не вводя никакой информации, однако при вводе некорректных данных программа выдаст звуковой сигнал и очистит поле ввода.

Для присвоения параметру значения, введенному в поле ввода, ввод должен закончиться одной из клавиш завершения ввода - это или клавиша Enter, или клавиша ↓, или комбинация клавиш Ctrl-Enter. Если ввод данных в поле ввода не заканчивался клавишей завершения, то присвоение введенных данных соответствующему параметру не произойдет, и после перехода на другое поле ввода клавишами отказа - Esc, ↑ либо Ctrl-Esc значение параметра примет прежнее, до его изменения, значение.

## Список

*Список* - способ представления данных. Список имеет обязательную часть - Окно просмотра, и может иметь несколько необязательных полей ввода

и полей информации. Перемещение по списку осуществляется с помощью следующих клавиш:

Действие	Клавиша
На следующую запись	↓
На предыдущую запись	↑
На первую запись окна	Home
На последнюю запись окна	End
Следующая страница списка	<u>PgDn</u>
Предыдущая страница списка	<u>PgUp</u>
На первую строку списка	Ctrl-PgUp
На последнюю строку списка	Ctrl-PgDn

*Страницей списка* называется совокупность записей, помещающихся в окне просмотра. Текущей строкой или записью называется строка, выделенная в списке маркером.

### Меню

*Меню* представляет собой фиксированный список каких-либо значений или действий. Служит для выбора пользователем дальнейших действий или для внесения данных, имеющих ограниченное число вариантов (например, меню используется масштабирование ЭКГ сигналы, где появляются на мониторы). Перемещение по пунктам меню осуществляется клавишами:

Действие	Клавиша
на следующий пункт	↓ или →
на предыдущий пункт	↑ или ←
на первый пункт меню	Home
на последний пункт меню	End

Выбрать пункт меню (выбрать действие, определяемое этим пунктом) - выделить этот пункт меню маркером и нажать клавишу Enter. Выход из меню без выбора - Esc.

### Функциональные клавиши

Функциональные клавиши - клавиши F1...F12, а также комбинации их или алфавитно-цифровых клавиш с модифицирующими клавишами Shift, Ctrl и Alt. Нажатие функциональных клавиш приводит к выполнению специфичных, связанных с данной клавишей (комбинацией клавиш), функций.

### База данных

*База данных* (БД) - совокупность данных одного вида. Например, создается внутренняя база данных с помощью математических алгоритмов. База данных содержит цифры, означающие амплитуды и частоты.

### Назначение клавиш

Действие	Клавиша
курсор влево	←
курсор вправо	→
в конец поля ввода	End
в начало поля ввода	Home
отказ от действия; переход к предыдущему полю ввода; прекращение выполнения текущего действия	Esc
Завершение ввода в поле ввода; переход к следующему полю ввода; активизация действия (в меню)	Enter
переключение блока цифровой клавиатуры	Num Lock
переключение блока цифровой клавиатуры	Caps Lock

Действие	Клавиша
активизация режима добавления; переключение между режимами ВСТАВКА- ЗАМЕНА	Ins
Активизация режима удаления; удаление символа в позиции курсора	Del
удаление символа перед курсором	Backspace
удаление символов от позиции курсора	Ctrl-End
удаление символов во всей строке	Ctrl-Home

Режим ВСТАВКА - вводимые символы вставляются в позиции курсора. Текст справа от курсора сдвигается.

Режим ЗАМЕНА - вводимые символы вставляются в позиции курсора. Имеющиеся символы затираются.

Если названия клавиш написаны через тире, то их необходимо нажимать одновременно. Например, сочетание Ctrl-Enter означает, что необходимо нажать клавишу Ctrl, и, не отпуская ее, однократно нажать клавишу **Enter** (но не наоборот).

При выполнении любых действий по изменению данных следует помнить, что для сохранения результата действия необходимо закончить одной клавишей завершения Enter, ↓, Ctrl-Enter. Для отказа от сделанных изменений необходимо завершить действие клавишами отказа - Esc, ↑ или Ctrl-Esc. Еще одно существенное отличие клавиш Esc и Enter от Ctrl-Esc и Ctrl-Enter заключается в том, что при выполнении некоторых действий, которые могут привести к необратимым последствиям (например, потере введенных данных), при нажатии комбинации клавиш Ctrl-Esc или Ctrl-Enter программа выполняет заданные действия без запросов. При выполнении действий через клавиши Esc или Enter программа запрашивает подтверждение на выполнение действия.

### 3. Инструкция для пользователей программы

Сначала загружается основной файл nfilter.exe, где расположена оболочка в памяти компьютера. Оболочка состоит из несколько частей:

**Зашумлённая ЭКГ**, которую мы получили с помощью математических функций (рис. 3.1).

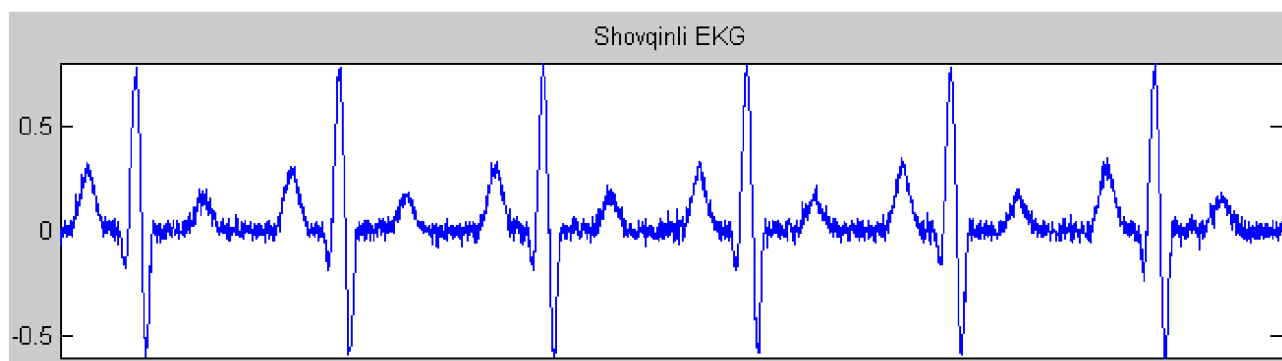


Рис.3.1. Зашумлённая ЭКГ

**Фильтрованная ЭКГ**, которую отфильтровали на основе спектральных функций. Эта часть считается выходной частью (рис.3.2.).

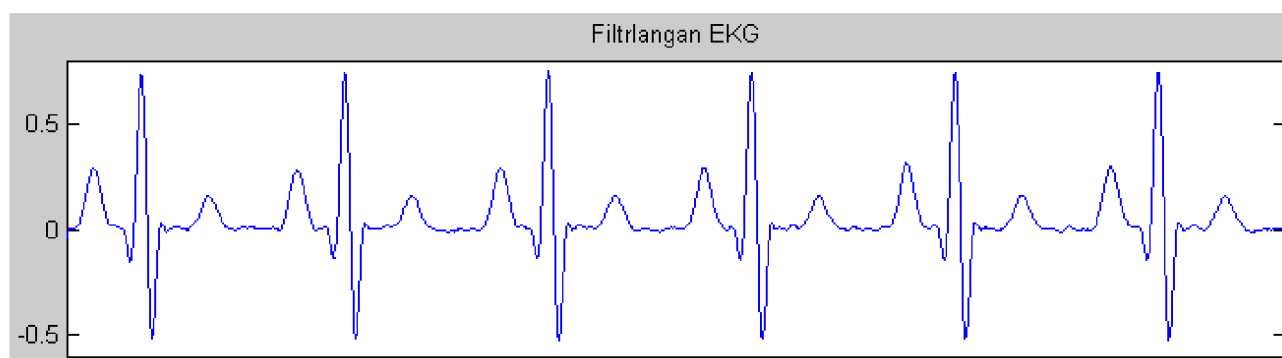


Рис.3.2. Фильтрованная ЭКГ

**Качество фильтрации.** В этом окне даются результаты фильтрации (рис.3.3).



Рис.3.3. Среднеквадратические ошибки.

**Шумовой уровень.** С помощью этой управляющей части можно повышать или понижать шумовой уровень Рис.3.4. Управляющая часть регулируется с помощью мышки или клавиатурными кнопками ←, →, ↑, ↓.

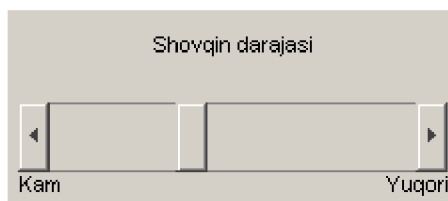


Рис.3.4. Шумовой уровень

**Параметры фильтра.** Параметры фильтра состоят из границ спектральных коэффициентов (Рис.3.5.). Здесь: Граница спектральных коэффициентов – состоит из трех цифр: 55,75,85.

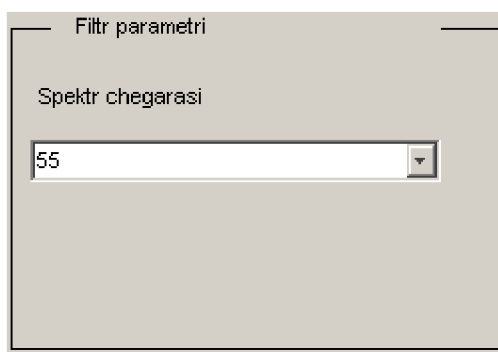


Рис.3.5. Параметры фильтра.

**Управляющие кнопки.** С помощью этих кнопок можно получить дополнительную информацию о программе, об авторах или выйти из программы (рис.3.6).

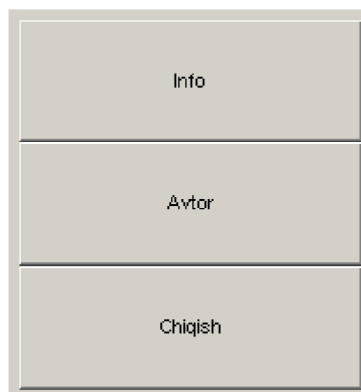


Рис.3.6. Управляющие кнопки

В итоге мы получаем общий вид программы (рис.3.7).



Рис.3.7. Оболочка программного пакета

Программой предоставляется дополнительная возможность, с помощью которой можно увидеть зубец ЭКГ (P, Q, R, S, T) в выделенном большом формате чтобы оценить качество фильтрации (рис.3.8).

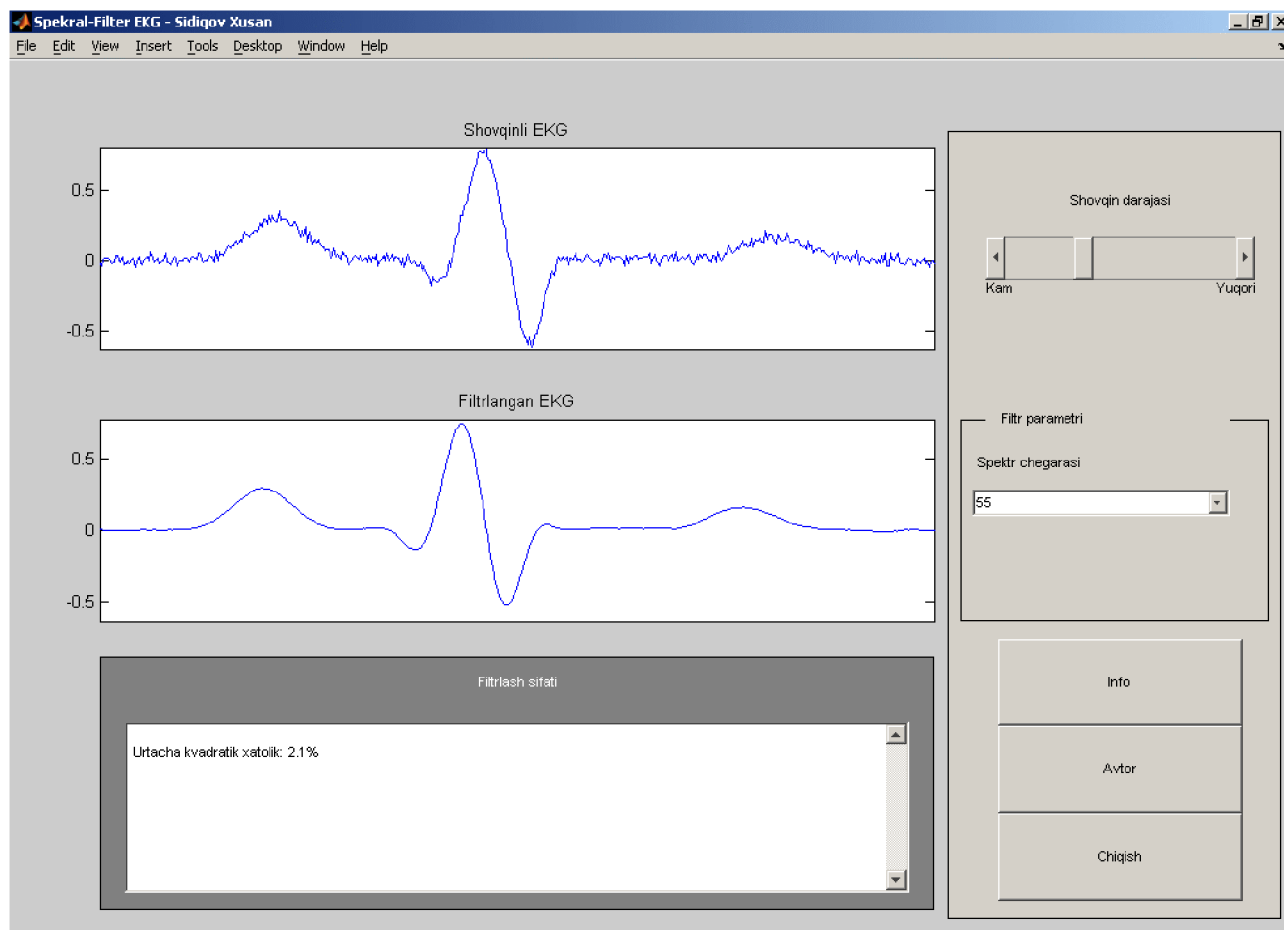


Рис.3.8. Качественное сопоставление фильтрации

При работе с программой можно пользоваться следующими кнопками клавиатуры:

←, ↓ - кнопки понижения параметров;

→, ↑ - кнопки повышения параметров;

F10 – переход на строку меню;

Enter – подтверждения выбранных параметров;

Tab – переход с одной управляющей кнопки на другую.

### **Выводы по главе III**

1. В III главе дано описание и функциональное назначение программного пакета для анализа и обработки ЭКГ-данных, его структура и область применения, краткое описание основных терминов, понятий и элементов программы, что облегчает задачу пользователя в процессе ознакомления с программой и ее практического применения.

2. Даны основные характеристики сигналов ЭКГ, области применения разработанного программного пакета, характеристика входной и выходной информации, описание структуры программного пакета.

3. Для пользователей – медиков, не имеющих специальной компьютерной подготовки, дан порядок функционирования программ и обозначены клавиши взаимодействия для меню, списка, функций, расчетов.

4. Разработана специальная инструкция для пользователей программы с описанием загружаемых файлов, получением сигналов, даны параметры фильтра и отфильтрованных сигналов, даны параметры фильтра и уровня шума. Программа предоставляет возможность качественной оценки работы фильтра.

## Заключение

В ходе выполнения диссертационной работы были достигнуты следующие результаты.

1. Исследованы существующие в настоящее время системы автоматизированного анализа ЭКГ и принципы разработки программного обеспечения для анализа ЭКГ, основанные на определении соотношений её графических параметров. На основании обширного фактического материала делается вывод об основных преимуществах компьютерной регистрации сигналов ЭКГ по сравнению с аналоговой точки зрения фильтрации шумов различного происхождения.

2. Анализ видов помех, возникающих при регистрации сигналов ЭКГ, методов их устранения и совершенствования аппаратуры и методик регистрации дает возможность совершенствовать используемые методологии фильтрации сигналов ЭКГ, повысить качество обработки, увеличить точность диагноза.

3. Предлагаемый собственный подход к решению вопроса фильтрации сигналов ЭКГ, осуществляемый на основе спектральных функций имеет очевидные алгоритмические преимущества: высокую скорость реализации, простые численные алгоритмы, требуемую точность обработки.

4. Разработанный программный пакет и инструкции пользователя имеют большую практическую значимость при расшифровке сигналов ЭКГ с целью постановки диагноза, так как обеспечивает работу в реальном масштабе времени, устраняет наиболее характерные виды шумов, может быть реализован на обычном персональном компьютере и использоваться обычным медиком без специальной подготовки.

## Литература

1. Каримов И.А. Мировой финансово-экономический кризис, пути и меры по его преодолению в условиях Узбекистана. – Т.: Узбекистан, 2009. – 31 с.
2. Каримов И.А. Узбекский народ никогда и ни от кого не будет зависеть. Т.13. – Т.: «Узбекистан», 2005. – 264 с.
3. Залманзон Л.А. Преобразования Фурье, Уолша, Хаара и их применение в управлении, связи и других областях. – М.: Наука, 1989. – 496с.
4. Зудбинов Ю.И. Азбука ЭКГ. – Издание 3. – Ростов-на-Дону: «Феникс», 2003. – 160 с.
5. Е.В. Аксенов и др. Системы сбора и обработки электрофизиологической информации на основе сигма-дельта аналого-цифрового преобразования //Биомедицинские технологии и радиоэлектроника. - №12-2001. - стр. 56-65.
6. Водолазский Л. А. Основы техники клинической электрографии.- Москва: Медицина.-1966.-270 с.
7. Шакин В. В. Вычислительная электрокардиография.- М.: Наука.-1981.-166 с.
8. Дроздов Д. В. Персональный компьютер в качестве электрокардиографа // М: Компьютерные технологии в медицине.-1996.-№1.-С. 18-23.
9. Селищев В.А Автоматизированное проектирование биомедицинских электронных систем // Биомедицинские технологии и радиоэлектроника..- М.: №12,2001. - с. 5-17.
- 10.Сергиенко А. Б. Цифровая обработка сигналов. – 2-е. – СПб.: Питер, 2007. – С. 751.
- 11.Оппенгейм А., Шафер Р. Цифровая обработка сигналов. Изд. 2-е, испр. – М.: Техносфера, 2007. – 856 с.
- 12.Блейхут Р. Быстрые алгоритмы цифровой обработки сигналов. - М.: Мир 1989.
- 13.Дьяконов В. П. MATLAB 6.5 SP1/7.0 + Simulink 5/6/ Обработка сигналов и проектирование фильтров. – М.: СОЛОН-Пресс, 2005. – 676 с.

14. Дьяконов В. П. MATLAB и SIMULINK для радиоинженеров. – Москва.: «ДМК-Пресс», 2011. – С. 976.
15. Латфуллин И. А., Тептин Г. М. Возможности электрокардиографии высокого разрешения. - Казанский медицинский журнал, 1998, 79, 2.
16. Рабинер Л., Гоулд Б. Теория и применение цифровой обработки сигналов: Пер с англ./ Под ред. Ю. И. Александрова. - М.: Мир, 1978.
17. Степура О. Б., Остроумова О. Д., Курильченко И. Т., Мартынов А. И. Клиническая значимость сигналусреднённой электрокардиографии. - Российские Медицинские Вести N1, 1997.
18. М. С. Куприянов, Б. Д. Матюшкин. Цифровая Обработка Сигналов – СПб: Политехника, 2000, С. 304-325.
19. В. В. Мурашко, А. В. Струтынский. Электрокардиография – Москва: Медицина, 1991
20. А. З. Чернов, М. И. Кечкер. Электрокардиографический атлас – Москва: Медицина, 1979
21. В. Н. Орлов. Руководство по электрокардиографии – Москва: Медицина, 1984.
22. Р. М. Баевский, Г. Г. Иванов, Вариабельность сердечного ритма: теоретические аспекты и возможности клинического применения – Москва, 2000, Гл. 3.2.
23. Г. Г. Иванов. Электрокардиография высокого разрешения – Москва, 1999.
24. К. Грегори. Использование Visual C++ 6.0. Вильямс, Москва, 2000.
25. Аввад Насер, С.В. Солошенко, С.А.Филист. Спектральное разложение электрокардиосигналов на нестационарных отрезках собственных частот // Конференция Биомедприбор, 2000.
26. А.П.Иванов, И.А.Эльгардт, Н.С.Сдобнякова, В.А.Леонтьев. Некоторые особенности спектрального анализа сердечного ритма в оценке вегетативного баланса у больных инфарктом миокарда. // Конференция Биомедприбор, 2000.
27. Т.В.Истомина, Л.Ю.Кривоногов. Перспективы применения ранговых

- методов для обнаружения границ информативных участков электрокардиосигнала. // Конференция Биомедприбор, 2000.
28. С.А. Гаспарян, Т.В. Зарубина, С.Л. Швырев. Развитие интегральной автоматизированной системы постоянного интенсивного наблюдения за состоянием пациентов реанимационных отделений. // Конференция Биомедприбор, 2000.
29. Шитов А. Б., Разработка численных методов и программ, связанных с применением вейвлет-анализа для моделирования и обработки экспериментальных данных, Иваново — 2001.
30. И.М. Соболев. Многомерные квадратурные формулы и функции Хаар. – М.: «Наука», 1969.
31. Файнзильберг Л.С. Синтез информационных технологий обработки сигналов//Управляющие системы и машины. –1998. - № 2. - С.40-47.
32. Цыпкин Я.З. Адаптация и обучение в автоматических системах. -М.: Наука, 1968. - 400 с.
33. Мизин И.А., Матвеев А.А. Цифровые фильтры.-М. : Радио и связь.- 1979, 386 с.
34. Антонью А. Цифровые фильтры: Анализ и проектирование. - М.: Радио и связь.- 1983, 320 с.
35. Oppenheim A. V., R.W. Schaffer. Discrete-Time Signal Processing.- Englewood Cliffs, NJ: Prentice-Hall.-1989.-P. 311-312.
36. Gustafsson F. Determining the initial states in forward-backward filtering // IEEE Transactions on Signal Processing.- April 1996, Volume 44, Issue 4.- P. 988—992.
37. Задирака В.К., Мельникова С.С. Цифровая обработка сигналов. - К.: Наук.думка, 1993. - 294 с.
38. Файнзильберг Л.С. Адаптивное сглаживание шумов в информационных технологиях обработки физиологических сигналов. - Математические машины и системы.-2002, № 3.- С. 96-104.
39. Сидиков Х.Б. Способы сохранения качества сигнала при цифровой

обработке. // Конференция «Фан ва техника тараккиётида интеллектуал ёшларнинг ўрни», ТГТУ - 2013.

40.<http://ru.wikipedia.org/wiki/Электрокардиография> (Электрокардиография)

41.<http://www.biosignal.ru> (ООО Биосигнал - разработка программного обеспечения)

42.<http://www.sciteclibrary.ru/rus/catalog/pages/5757.html> (SciTecLibrary - Статьи и Публикации)

## Приложение

Исходный код программы:

```
#include "nfilter.h"
#include "libmatlbn.h"
#include "libmmfile.h"
#include "libmwsglm.h"
#include "sgolayfilt.h"
#include "subplot.h"
#include "title.h"

static mxChar _array1_[10] = { 'i', 'n', 'i', 't', 'i',
                              'a', 'l', 'i', 'z', 'e' };

static mxArray * _mxarray0_;
static mxArray * _mxarray2_;
static mxArray * _mxarray3_;
static mxArray * _mxarray4_;

static mxChar _array6_[11] = { 'N', 'u', 'm', 'b', 'e', 'r',
                              'T', 'i', 't', 'l', 'e' };

static mxArray * _mxarray5_;

static mxChar _array8_[3] = { 'o', 'f', 'f' };
static mxArray * _mxarray7_;

static mxChar _array98_[15] = { 0x00d8, 0x00f3, 0x00ec, 0x00ee, 0x00e2,
                              0x00ee, 0x00e9, ' ', 0x00f3, 0x00f0,
                              0x00ee, 0x00e2, 0x00e5, 0x00ed, 0x00fc };

static mxArray * _mxarray97_;

static double _array100_[4] = { .76, .8, .21, .05 };

void InitializeModule_nfilter(void) {
    _mxarray0_ = mclInitializeString(10, _array1_);
    _mxarray2_ = mclInitializeDouble(500.0);
    _mxarray3_ = mclInitializeDouble(0.0);
    _mxarray4_ = mclInitializeDouble(15.0);
    _mxarray5_ = mclInitializeString(11, _array6_);
    _mxarray7_ = mclInitializeString(3, _array8_);
    _mxarray230_ = mclInitializeDoubleVector(1, 15, _array231_);
    _mxarray232_ = mclInitializeDoubleVector(1, 15, _array233_);
}

void TerminateModule_nfilter(void) {
    mxDestroyArray(_mxarray232_);
    mxDestroyArray(_mxarray230_);
    mxDestroyArray(_mxarray3_);
    mxDestroyArray(_mxarray2_);
    mxDestroyArray(_mxarray0_);
}

static mxArray * mlfNfilter_update_signal(mxArray * * f,
                                          mxArray * y,
                                          mxArray * noise,
                                          mxArray * degree,
                                          mxArray * frame);

static void mlxNfilter_update_signal(int nlhs,
                                     mxArray * plhs[],
                                     int nrhs,
```

```

                                mxArray * prhs[]);
static void mlfNfilter_update_gui(mxArray * y,
                                mxArray * noise,
                                mxArray * degree,
                                mxArray * frame);
static void mlxNfilter_update_gui(int nlhs,
                                mxArray * plhs[],
                                int nrhs,
                                mxArray * prhs[]);
static mxArray * mlfNfilter_ecg(mxArray * L);
static void mlxNfilter_ecg(int nlhs,
                           mxArray * plhs[],
                           int nrhs,
                           mxArray * prhs[]);
static void Mnfilter(mxArray * action);
static mxArray * Mnfilter_update_signal(mxArray * * f,
                                       int nargout_,
                                       mxArray * y,
                                       mxArray * noise,
                                       mxArray * degree,
                                       mxArray * frame);
static void Mnfilter_update_gui(mxArray * y,
                                mxArray * noise,
                                mxArray * degree,
                                mxArray * frame);
static mxArray * Mnfilter_ecg(int nargout_, mxArray * L);

static mexFunctionTableEntry local_function_table_[3]
= { { "update_signal", mlxNfilter_update_signal, 4, 2, NULL },
    { "update_gui", mlxNfilter_update_gui, 4, 0, NULL },
    { "ecg", mlxNfilter_ecg, 1, 1, NULL } };

_mexLocalFunctionTable _local_function_table_nfilter
= { 3, local_function_table_ };

void mlxNfilter(int nlhs, mxArray * plhs[], int nrhs, mxArray * prhs[]) {
    mxArray * mprhs[1];
    int i;
    if (nlhs > 0) {
        mlfError(
            mxCreateString(
                "Run-time Error: File: nfilter Line: 1 Column: "
                "1 The function \"nfilter\" was called with mor"
                "e than the declared number of outputs (0)."),
            NULL);
    }
    if (nrhs > 1) {
        mlfError(
            mxCreateString(
                "Run-time Error: File: nfilter Line: 1 Column:"
                " 1 The function \"nfilter\" was called with m"
                "ore than the declared number of inputs (1)."),
            NULL);
    }
    for (i = 0; i < 1 && i < nrhs; ++i) {
        mprhs[i] = prhs[i];
    }
    for (; i < 1; ++i) {
        mprhs[i] = NULL;
    }
    mlfEnterNewContext(0, 1, mprhs[0]);
    Mnfilter(mprhs[0]);
    mlfRestorePreviousContext(0, 1, mprhs[0]);
}

```

```

static void mlxNfilter_update_signal(int nlhs,
                                     mxArray * plhs[],
                                     int nrhs,
                                     mxArray * prhs[]) {

    mxArray * mprhs[4];
    mxArray * mplhs[2];
    int i;
    if (nlhs > 2) {
        mlfError(
            mxCreateString(
                "Run-time Error: File: nfilter/update_signal Line: 291 Co"
                "lumn: 1 The function \"nfilter/update_signal\" was calle"
                "d with more than the declared number of outputs (2)."),
            NULL);
    }
    if (nrhs > 4) {
        mlfError(
            mxCreateString(
                "Run-time Error: File: nfilter/update_signal Line: 291 C"
                "olumn: 1 The function \"nfilter/update_signal\" was cal"
                "led with more than the declared number of inputs (4)."),
            NULL);
    }
    for (i = 0; i < 2; ++i) {
        mplhs[i] = NULL;
    }
    for (i = 0; i < 4 && i < nrhs; ++i) {
        mprhs[i] = prhs[i];
    }
    for (; i < 4; ++i) {
        mprhs[i] = NULL;
    }
    mlfEnterNewContext(0, 4, mprhs[0], mprhs[1], mprhs[2], mprhs[3]);
    mplhs[0]
        = Mnfilter_update_signal(
            &mplhs[1], nlhs, mprhs[0], mprhs[1], mprhs[2], mprhs[3]);
    mlfRestorePreviousContext(0, 4, mprhs[0], mprhs[1], mprhs[2], mprhs[3]);
    plhs[0] = mplhs[0];
    for (i = 1; i < 2 && i < nrhs; ++i) {
        plhs[i] = mplhs[i];
    }
    for (; i < 2; ++i) {
        mxDestroyArray(mplhs[i]);
    }
}

static void mlfNfilter_update_gui(mxArray * y,
                                  mxArray * noise,
                                  mxArray * degree,
                                  mxArray * frame) {
    mlfEnterNewContext(0, 4, y, noise, degree, frame);
    Mnfilter_update_gui(y, noise, degree, frame);
    mlfRestorePreviousContext(0, 4, y, noise, degree, frame);
}

static void mlxNfilter_update_gui(int nlhs,
                                   mxArray * plhs[],
                                   int nrhs,
                                   mxArray * prhs[]) {

    mxArray * mprhs[4];
    int i;
    if (nlhs > 0) {
        mlfError(

```

```

        mxCreateString(
            "Run-time Error: File: nfilter/update_gui Line: 297 Col"
            "umn: 1 The function \"nfilter/update_gui\" was called "
            "with more than the declared number of outputs (0)."),
        NULL);
    }
    if (nrhs > 4) {
        mlfError(
            mxCreateString(
                "Run-time Error: File: nfilter/update_gui Line: 297 Co"
                "lumn: 1 The function \"nfilter/update_gui\" was calle"
                "d with more than the declared number of inputs (4)."),
            NULL);
    }
    for (i = 0; i < 4 && i < nrhs; ++i) {
        mprhs[i] = prhs[i];
    }
    for (; i < 4; ++i) {
        mprhs[i] = NULL;
    }
    mlfEnterNewContext(0, 4, mprhs[0], mprhs[1], mprhs[2], mprhs[3]);
    Mfilter_update_gui(mprhs[0], mprhs[1], mprhs[2], mprhs[3]);
    mlfRestorePreviousContext(0, 4, mprhs[0], mprhs[1], mprhs[2], mprhs[3]);
}

static void Mfilter(mxArray * action) {
    mexLocalFunctionTable save_local_function_table_
        = mclSetCurrentLocalFunctionTable(&_local_function_table_nfilter);
    int nargin_ = mclNargin(1, action, NULL);
    mxArray * hlpStr1 = NULL;
    mxArray * ttlStr = NULL;
    mxArray * frm = NULL;
    mxArray * frmindx = NULL;
    mxArray * deg = NULL;
    mxArray * degindx = NULL;
    mxArray * hf = NULL;
    mxArray * hd = NULL;
    mxArray * hn = NULL;
    mxArray * ud = NULL;
    mxArray * ha2 = NULL;
    mxArray * ha1 = NULL;
    mxArray * mcwHndl = NULL;
    mxArray * mcwPos = NULL;
    mxArray * labelPos = NULL;
    mxArray * frmPos = NULL;
    mxArray * frmBorder = NULL;
    mxArray * promptStr = NULL;
    mxArray * spacing = NULL;
    mxArray * labelHt = NULL;
    mxArray * bottom = NULL;
    mxArray * right = NULL;
    mxArray * left = NULL;
    mxArray * top = NULL;
    mxArray * frame = NULL;
    mxArray * degree = NULL;
    mxArray * noise = NULL;
    mxArray * tposition8 = NULL;
    mxArray * tposition7 = NULL;
    mxArray * tposition6 = NULL;
    mxArray * slimax = NULL;
    mxArray * tposition5 = NULL;
    if (nargin_ == 0) {
        /*
         * action = 'initialize';

```

```

        */
        mlfAssign(&action, _mxarray0_);
/*
* end
*/
}
/*
*
* x1 = ecg(500).';
*/
mlfAssign(&x1, mlfTranspose(mlfNfilter_ecg(_mxarray2_)));
/*
* x = [x1;x1;x1;x1;x1;x1];
*/
mlfAssign(
    &x,
    mlfVertcat(
        mclVv(x1, "x1"),
        mclVv(x1, "x1"),
        mclVv(x1, "x1"),
        mclVv(x1, "x1"),
        mclVv(x1, "x1"),
        mclVv(x1, "x1"),
        NULL));
/*
* y = sgolayfilt(x,0,15); %Бесшумный сигнал
*/
mlfAssign(
    &y, mlfSgolayfilt(mclVv(x, "x"), _mxarray3_, _mxarray4_, NULL, NULL));
/*
*
* switch action,
*/
{
    mxArray * v_ = mclInitialize(mclVa(action, "action"));
    if (mclSwitchCompare(v_, _mxarray0_)) {
        /*
        * case 'initialize',
        *
        * % Initialize Graphics
        * hfig = figure('NumberTitle','off',...
        */
        mlfAssign(
            &hfig,
            mlfNFigure(
                1, _mxarray5_, _mxarray7_, _mxarray9_, _mxarray11_, NULL));
        /*
        * 'Name','Neuro-Filter EKG - Furkat Rahmatov');
        * bgc = get(0,'DefaultUIControlBackgroundColor');
        */
        mlfAssign(&bgc, mlfNGet(1, _mxarray3_, _mxarray13_, NULL));
        /*
        *
        * % Структура(рамка) для всего средства управления
        * framepos = [.73 .02 .26 .90];
        */
        mlfAssign(&framepos, _mxarray15_);
        /*
        * uicontrol( ...
        */
        mclAssignAns(
            &ans,
            mlfNUicontrol(
                0,

```

```

        _mxarray17_,
        _mxarray19_,
        _mxarray21_,
        _mxarray23_,
        _mxarray25_,
        mclVv(bgc, "bgc"),
        _mxarray27_,
        mclVv(framepos, "framepos"),
        NULL));
/*
 * 'Style','frame',...
 * 'Units','normalized',...
 * 'BackgroundColor',bgc,...
 * 'Position',framepos);
 *
 * % Структура(рамка) для секции параметров
 * framepos2 = [.74 .36 .24 .23];
 */
mlfAssign(&framepos2, _mxarray29_);
/*
 * uicontrol( ...
 */
mclAssignAns(
    &ans,
    mlfNUicontrol(
        0,
        _mxarray17_,
        _mxarray19_,
        _mxarray21_,
        _mxarray23_,
        _mxarray31_,
        _mxarray33_,
        _mxarray27_,
        mclVv(framepos2, "framepos2"),
        NULL));
/*
 * 'Style','frame',...
 * 'Units','normalized',...
 * 'ForegroundColor','black',...
 * 'Position',framepos2);
 *
 * % Параметры Нейро Фильтра
 * titlepos = [.76 .57 .19 .03];
 */
mlfAssign(&titlepos, _mxarray35_);
/*
 * titlelabel = 'Параметры фильтра';
 */
mlfAssign(&titlelabel, _mxarray37_);
/*
 * uicontrol( ...
 */
mclAssignAns(
    &ans,
    mlfNUicontrol(
        0,
        _mxarray39_,
        mclVv(titlelabel, "titlelabel"),
        _mxarray41_,
        _mxarray43_,
        _mxarray17_,
        _mxarray45_,
        _mxarray21_,
        _mxarray23_,

```

```

        _mxarray25_,
        mclVv(bgc, "bgc"),
        _mxarray31_,
        _mxarray33_,
        _mxarray27_,
        mclVv(titlepos, "titlepos"),
        NULL));
/*
 * 'String',titlelabel,...
 * 'HorizontalAlignment','left',...
 * 'Style','text',...
 * 'Units','normalized',...
 * 'BackgroundColor',bgc,...
 * 'ForegroundColor','black',...
 * 'Position',titlepos);
 * % SG меню градуса (степени)Число слоев
 * %купхад полином
 * menunamel = 'Степень полинома';
 */
mlfAssign(&menunamel, _mxarray47_);
/*
 * tposition1 = [.75 .5 .2 .05];
 */
mlfAssign(&tposition1, _mxarray49_);
/*
 * uicontrol( ...
 */
mclAssignAns(
    &ans,
    mlfNUicontrol(
        0,
        _mxarray39_,
        mclVv(menunamel, "menunamel"),
        _mxarray41_,
        _mxarray43_,
        _mxarray17_,
        _mxarray45_,
        _mxarray21_,
        _mxarray23_,
        _mxarray31_,
        _mxarray33_,
        _mxarray27_,
        mclVv(tposition1, "tposition1"),
        NULL));
/*
 * 'String',menunamel,...
 * 'HorizontalAlignment','left',...
 * 'Style','text',...
 * 'Units','normalized',...
 * 'ForegroundColor','black',...
 * 'Position',tposition1);
 * mposition1 = [.77 .46 .1 .05];
 */
mlfAssign(&mposition1, _mxarray51_);
/*
 * popstr1 = {'1','2','3','4'};
 */
mlfAssign(&popstr1, _mxarray53_);
/*
 * degreehdl = uicontrol('String',popstr1,...
 */
mlfAssign(
    &degreehdl,
    mlfNUicontrol(

```

```

1,
_mxarray39_,
mclVv(popstr1, "popstr1"),
_mxarray17_,
_mxarray63_,
_mxarray65_,
_mxarray67_,
_mxarray21_,
_mxarray23_,
_mxarray69_,
_mxarray71_,
_mxarray27_,
mclVv(mposition1, "mposition1"),
_mxarray73_,
mlfStr2double(mclVv(popstr1, "popstr1")),
_mxarray75_,
_mxarray77_,
NULL));
/*
 * 'Style','Popup',...
 * 'Tag','degreepopup',...
 * 'Units','normalized',...
 * 'BackgroundColor','white',...
 * 'Position',mposition1,...
 * 'UserData',str2double(popstr1),...
 * 'CallBack','sgolaydemo(''recal'')');
 * % SG создают меню размера Число нейроны
 * menuname2 = 'Размер структуры';
 */
mlfAssign(&menuname2, _mxarray79_);
/*
 * tposition2 = [.75 .4 .2 .05];
 */
mlfAssign(&tposition2, _mxarray81_);
/*
 * uicontrol( ...
 */
mclAssignAns(
    &ans,
    mlfNUicontrol(
        0,
        _mxarray39_,
        mclVv(menuname2, "menuname2"),
        _mxarray17_,
        _mxarray45_,
        _mxarray41_,
        _mxarray43_,
        _mxarray21_,
        _mxarray23_,
        _mxarray31_,
        _mxarray33_,
        _mxarray27_,
        mclVv(tposition2, "tposition2"),
        NULL));
/*
 * 'String',menuname2,...
 * 'Style','text',...
 * 'HorizontalAlignment','left',...
 * 'Units','normalized',...
 * 'ForegroundColor','black',...
 * 'Position',tposition2);
 * mposition2 = [.77 .37 .1 .05];
 */
mlfAssign(&mposition2, _mxarray83_);

```

```

/*
 * popstr2 = {'5','15','25','55'};
 */
mlfAssign(&popstr2, _mxarray85_);
/*
 * framehdl = uicontrol('String',popstr2,...
 */
mlfAssign(
    &framehdl,
    mlfNUicontrol(
        1,
        _mxarray39_,
        mclVv(popstr2, "popstr2"),
        _mxarray17_,
        _mxarray63_,
        _mxarray65_,
        _mxarray95_,
        _mxarray69_,
        _mxarray71_,
        _mxarray21_,
        _mxarray23_,
        _mxarray27_,
        mclVv(mposition2, "mposition2"),
        _mxarray73_,
        mlfStr2double(mclVv(popstr2, "popstr2")),
        _mxarray75_,
        _mxarray77_,
        NULL));
/*
 * 'Style','Popup',...
 * 'Tag','framepopup',...
 * 'BackgroundColor','white',...
 * 'Units','normalized',...
 * 'Position',mposition2,...
 * 'UserData',str2double(popstr2),...
 * 'Callback','sgolaydemo(''recal'')');
 * % Шумовая разница slider
 * sliname1 = 'Шумовой уровень';
 */
mlfAssign(&sliname1, _mxarray97_);
/*
 * tposition4 = [.76 .8 .21 .05];
 */
mlfAssign(&tposition4, _mxarray99_);
 * noise = 0.7;
 */
mlfAssign(&noise, _mxarray109_);
/*
 * degree = 0;
 */
mlfAssign(&degree, _mxarray3_);
/*
 * frame = 5;
 */
mlfAssign(&frame, _mxarray145_);
/*
 * %subplot('Position',[.07 .05 .65 .23]), plot(y)
 * %set(gca,'XTick',[])
 * %axis([0 3000 -.6 .8])
 * % Set up the MiniCommand Window
 * top=0.30;
 */
mlfAssign(&top, _mxarray146_);
/*

```

```

    * left=0.09;
    */
mlfAssign(&left, _mxarray147_);
/*
    * right=0.70;
    */
mlfAssign(&right, _mxarray109_);
/*
    * bottom=0.05;
    */
mlfAssign(&bottom, _mxarray148_);
/*
    * labelHt=0.05;
    */
mlfAssign(&labelHt, _mxarray148_);
/*
    * spacing=0.005;
    */
mlfAssign(&spacing, _mxarray149_);
/*
    * promptStr=str2mat(' ', ' % Олинган натижалар хакида', ...
    */
mlfAssign(
    &promptStr,
    mlfStr2mat(
        _mxarray150_, _mxarray152_, _mxarray152_, _mxarray152_, NULL));
mlfAssign(&frmBorder, _mxarray154_);
/*
    * frmPos=[left-frmBorder bottom-frmBorder ...
    */
mlfAssign(
    &frmPos,
    mlfHorzcat(
        mclMinus(mclVv(left, "left"), mclVv(frmBorder, "frmBorder")),
        mclMinus(
            mclVv(bottom, "bottom"), mclVv(frmBorder, "frmBorder")),
        mclPlus(
            mclMinus(mclVv(right, "right"), mclVv(left, "left")),
            mclMtimes(_mxarray155_, mclVv(frmBorder, "frmBorder"))),
        mclPlus(
            mclMinus(mclVv(top, "top"), mclVv(bottom, "bottom")),
            mclMtimes(_mxarray155_, mclVv(frmBorder, "frmBorder"))),
        NULL));
/*
    * (right-left)+2*frmBorder (top-bottom)+2*frmBorder];
    * uicontrol( ...
    */
mclAssignAns(
    &ans,
    mlfNUicontrol(
        0,
        _mxarray17_,
        _mxarray19_,
        _mxarray21_,
        _mxarray23_,
        _mxarray27_,
        mclVv(frmPos, "frmPos"),
        _mxarray25_,
        _mxarray156_,
        NULL));
/*
    * 'Style','frame', ...
    * 'Units','normalized', ...
    * 'Position',frmPos, ...

```

```

* 'BackgroundColor',[0.50 0.50 0.50]);
* % Then the text label
* labelPos=[left top-labelHt (right-left) labelHt];
*/
mlfAssign(
    &labelPos,
    mlfHorzcat(
        mclVv(left, "left"),
        mclMinus(mclVv(top, "top"), mclVv(labelHt, "labelHt")),
        mclMinus(mclVv(right, "right"), mclVv(left, "left")),
        mclVv(labelHt, "labelHt"),
        NULL));

mlfAssign(
    &mcwPos,
    mlfHorzcat(
        mclVv(left, "left"),
        mclVv(bottom, "bottom"),
        mclMinus(mclVv(right, "right"), mclVv(left, "left")),
        mclMinus(
            mclMinus(
                mclMinus(mclVv(top, "top"), mclVv(bottom, "bottom")),
                mclVv(labelHt, "labelHt")),
            mclVv(spacing, "spacing")),
        NULL));
/*
* mcwHndl=uicontrol( ...
*/
mlfAssign(
    &mcwHndl,
    mlfNUicontrol(
        1,
        _mxarray17_,
        _mxarray162_,
        _mxarray41_,
        _mxarray43_,
        _mxarray21_,
        _mxarray23_,
        _mxarray112_,
        _mxarray164_,
        _mxarray25_,
        _mxarray158_,
        _mxarray27_,
        mclVv(mcwPos, "mcwPos"),
        _mxarray129_,
        _mxarray165_,
        _mxarray39_,
        mclVv(promptStr, "promptStr"),
        NULL));
/*
* 'Style','edit', ...
* 'HorizontalAlignment','left', ...
* 'Units','normalized', ...
* 'Max',10, ...
* 'BackgroundColor',[1 1 1], ...
* 'Position',mcwPos, ...
* 'Callback','graf2d('eval')', ...
* 'String',promptStr);
* % Save this handle for future use
* set(gcf,'UserData',mcwHndl);
*/
mclAssignAns(
    &ans,
    mlfNSet(

```

```

    0, mlfGcf(), _mxarray73_, mclVv(mcwHndl, "mcwHndl"), NULL));
mclPrintAns(&ans, mlfNTitle(0, _mxarray167_, NULL));
/*
 * ha1 = subplot('Position',[.07 .67 .65 .23]); plot(y)
 */
mlfAssign(
    &ha1, mlfNSubplot(1, _mxarray27_, _mxarray169_, NULL, NULL));
mclPrintAns(&ans, mlfNPlot(0, mclVv(y, "y"), NULL));
/*
 * set(gca,'XTick',[])
 */
mclPrintAns(
    &ans, mlfNSet(0, mlfGca(NULL), _mxarray171_, _mxarray173_, NULL));
/*
 * axis([0 3000 -.6 .8])
 */
mclPrintAns(&ans, mlfNAxis(0, NULL, NULL, _mxarray174_, NULL));
/*
 * title('Зашумлённая ЭКГ - Shovqinli EKG')
 */
mclPrintAns(&ans, mlfNTitle(0, _mxarray176_, NULL));
/*
 * ha2 = subplot('Position',[.07 .36 .65 .23]); plot(y)
 */
mlfAssign(
    &ha2, mlfNSubplot(1, _mxarray27_, _mxarray178_, NULL, NULL));
mclPrintAns(&ans, mlfNPlot(0, mclVv(y, "y"), NULL));
/*
 * set(gca,'XTick',[])
 */
mclPrintAns(
    &ans, mlfNSet(0, mlfGca(NULL), _mxarray171_, _mxarray173_, NULL));
/*
 * axis([0 3000 -.6 .8])
 */
mclPrintAns(&ans, mlfNAxis(0, NULL, NULL, _mxarray174_, NULL));
/*
 * title('Фильтрованным ЭКГ - Filtrlangan EKG')
 */
mclPrintAns(&ans, mlfNTitle(0, _mxarray180_, NULL));
/*
 * ud.handles.ax = [ha1 ha2];
 */
mlfIndexAssign(
    &ud,
    ".handles.ax",
    mlfHorzcat(mclVv(ha1, "ha1"), mclVv(ha2, "ha2"), NULL));
/*
 * ud.handles.uis = [degreehdl framehdl sliderhdl];
 */
mlfIndexAssign(
    &ud,
    ".handles.uis",
    mlfHorzcat(
        mclVv(degreehdl, "degreehdl"),
        mclVv(framehdl, "framehdl"),
        mclVv(sliderhdl, "sliderhdl"),
        NULL));
/*
 * set(hfig,'UserData',ud);
 */
mclAssignAns(
    &ans,
    mlfNSet(

```

```

        0, mclVv(hfig, "hfig"), _marray73_, mclVv(ud, "ud"), NULL));
/*
 * update_gui(y,noise,degree,frame)
 */
mlfNfilter_update_gui(
    mclVv(y, "y"),
    mclVv(noise, "noise"),
    mclVv(degree, "degree"),
    mclVv(frame, "frame"));
/*
 *
 * case 'recal',
 */
} else if (mclSwitchCompare(v_, _marray182_)) {
    /*
    * hfig = gcf;
    */
    mlfAssign(&hfig, mlfGcf());
    /*
    * ud = get(hfig, 'UserData');
    */
    mlfAssign(&ud, mlfNGet(1, mclVv(hfig, "hfig"), _marray73_, NULL));
    /*
    * hn = ud.handles.uis(3);
    */
    mlfAssign(
        &hn,
        mlfIndexRef(mclVv(ud, "ud"), ".handles.uis(?)", _marray184_));
    /*
    * hd = ud.handles.uis(1);
    */
    mlfAssign(
        &hd,
        mlfIndexRef(mclVv(ud, "ud"), ".handles.uis(?)", _marray185_));
    /*
    * hf = ud.handles.uis(2);
    */
    mlfAssign(
        &hf,
        mlfIndexRef(mclVv(ud, "ud"), ".handles.uis(?)", _marray155_));
    /*
    *
    * noise = get(hn, 'Value');
    */
    mlfAssign(&noise, mlfNGet(1, mclVv(hn, "hn"), _marray107_, NULL));
    /*
    * degindx = get(hd, 'Value');
    */
    mlfAssign(
        &degindx, mlfNGet(1, mclVv(hd, "hd"), _marray107_, NULL));
    /*
    * deg = get(hd, 'UserData');
    */
    mlfAssign(&deg, mlfNGet(1, mclVv(hd, "hd"), _marray73_, NULL));
    /*
    * degree = deg(degindx);
    */
    mlfAssign(
        &degree,
        mclArrayRef1(mclVv(deg, "deg"), mclVv(degindx, "degindx")));
    /*
    *
    * frmindx = get(hf, 'Value');
    */

```

```

mlfAssign(
    &frmindx, mlfNGet(1, mclVv(hf, "hf"), _mxarray107_, NULL));
/*
 * frm = get(hf, 'UserData');
 */
mlfAssign(&frm, mlfNGet(1, mclVv(hf, "hf"), _mxarray73_, NULL));
/*
 * frame = frm(frmindx);
 */
mlfAssign(
    &frame,
    mclArrayRef1(mclVv(frm, "frm"), mclVv(frmindx, "frmindx")));
/*
 * update_gui(y, noise, degree, frame)
 */
mlfNfilter_update_gui(
    mclVv(y, "y"),
    mclVv(noise, "noise"),
    mclVv(degree, "degree"),
    mclVv(frame, "frame"));
/*
 *
 * case 'info',
 */
} else if (mclSwitchCompare(v_, _mxarray186_)) {
    mlfAssign(&t1Str, _mxarray188_);
    /*
     *
     * hlpStr1= ...
     */
    mlfAssign(
        &hlpStr1,
        mlfHorzcat(
            _mxarray190_,
            mlfSprintf(NULL, _mxarray192_, NULL),
            _mxarray194_,
            mlfSprintf(NULL, _mxarray192_, NULL),
            _mxarray196_,
            mlfSprintf(NULL, _mxarray192_, NULL),
            _mxarray198_,
            mlfSprintf(NULL, _mxarray192_, NULL),
            _mxarray200_,
            mlfSprintf(NULL, _mxarray192_, NULL),
            _mxarray202_,
            mlfSprintf(NULL, _mxarray192_, NULL),
            _mxarray203_,
            mlfSprintf(NULL, _mxarray192_, NULL),
            _mxarray205_,
            mlfSprintf(NULL, _mxarray192_, NULL),
            _mxarray202_,
            mlfSprintf(NULL, _mxarray192_, NULL),
            _mxarray207_,
            mlfSprintf(NULL, _mxarray192_, NULL),
            _mxarray209_,
            mlfSprintf(NULL, _mxarray192_, NULL),
            _mxarray211_,
            mlfSprintf(NULL, _mxarray192_, NULL),
            _mxarray202_,
            mlfSprintf(NULL, _mxarray192_, NULL),
            _mxarray202_,
            mlfSprintf(NULL, _mxarray192_, NULL),
            _mxarray202_,
            NULL));
    mlfHelpwin(

```

```

        mclVv(hlpStr1, "hlpStr1"), mclVv(ttlStr, "ttlStr"), NULL, NULL);
/*
 *
 * case 'avtor',
 */
} else if (mclSwitchCompare(v_, _mxarray213_)) {
    /*
     * ttlStr = 'Авторы данного работ:';
     */
    mlfAssign(&ttlStr, _mxarray215_);
    /*
     *
     * hlpStr1= ...
     */
    mlfAssign(
        &hlpStr1,
        mlfHorzcat(
            _mxarray217_,
            mlfSprintf(NULL, _mxarray192_, NULL),
            _mxarray219_,
            mlfSprintf(NULL, _mxarray192_, NULL),
            _mxarray221_,
            mlfSprintf(NULL, _mxarray192_, NULL),
            _mxarray202_,
            mlfSprintf(NULL, _mxarray192_, NULL),
            _mxarray202_,
            mlfSprintf(NULL, _mxarray192_, NULL),
            _mxarray202_,
            NULL));
    * '' sprintf('\n')...
    * '' sprintf('\n')...
    * '';
    *
    * helpwin(hlpStr1, ttlStr);
    */
    mlfHelpwin(
        mclVv(hlpStr1, "hlpStr1"), mclVv(ttlStr, "ttlStr"), NULL, NULL);
/*
 *
 * case 'done',
 */
} else if (mclSwitchCompare(v_, _mxarray223_)) {
    /*
     * close(gcf);
     */
    mclAssignAns(&ans, mlfNClose(0, mlfGcf(), NULL));
/*
 * end
 */
}
mxDestroyArray(v_);
}

```

```

static mxArray * Mnfilter_update_signal(mxArray * * f,
                                       int nargout_,
                                       mxArray * y,
                                       mxArray * noise,
                                       mxArray * degree,
                                       mxArray * frame) {
    mexLocalFunctionTable save_local_function_table_
        = mclSetCurrentLocalFunctionTable(&_local_function_table_nfilter);
    mxArray * s = NULL;
    mxArray * v = NULL;
    mclCopyArray(&y);
}

```

```

mclCopyArray(&noise);
mclCopyArray(&degree);
mclCopyArray(&frame);
/*
 * %UPDATE_SIGNAL Повторно вычисляет шумный ECG и фильтрованные ECG сигналы.
 * v = 0.05*noise*randn(3000,1);    % Шум
 */
mlfAssign(
    &v,
    mclMtimes(
        mclMtimes(_mxarray148_, mclVa(noise, "noise")),
        mlfNRandomn(1, _mxarray225_, _mxarray185_, NULL)));
/*
 * s = y + v;          % Шумный ЭКГ
 */
mlfAssign(&s, mclPlus(mclVa(y, "y"), mclVv(v, "v")));
/*
 * f = sgolayfilt(s,degree,frame);
 */
mlfAssign(
    f,
    mlfSgolayfilt(
        mclVv(s, "s"),
        mclVa(degree, "degree"),
        mclVa(frame, "frame"),
        NULL,
        NULL));
mclValidateOutput(s, 1, nargout_, "s", "nfilter/update_signal");
mclValidateOutput(*f, 2, nargout_, "f", "nfilter/update_signal");
mxDestroyArray(v);
mxDestroyArray(frame);
mxDestroyArray(degree);
mxDestroyArray(noise);
mxDestroyArray(y);
mclSetCurrentLocalFunctionTable(save_local_function_table_);
return s;
/*
 *
 */
}

static void Mnfilter_update_gui(mxArray * y,
                                mxArray * noise,
                                mxArray * degree,
                                mxArray * frame) {
    mexLocalFunctionTable save_local_function_table_
        = mclSetCurrentLocalFunctionTable(&_local_function_table_nfilter);
    mxArray * ans = NULL;
    mxArray * hdf = NULL;
    mxArray * hdn = NULL;
    mxArray * axhdlfp = NULL;
    mxArray * axhdlnp = NULL;
    mxArray * ud = NULL;
    mxArray * f = NULL;
    mxArray * s = NULL;
    mxArray * hfig = NULL;
    mclCopyArray(&y);
    mclCopyArray(&noise);
    mclCopyArray(&degree);
    mclCopyArray(&frame);

    mlfAssign(&hfig, mlfGcf());
    /*
     * [s, f] = update_signal(y,noise,degree,frame);

```

```

    */
    mlfAssign(
        &s,
        mlfNfilter_update_signal(
            &f,
            mclVa(y, "y"),
            mclVa(noise, "noise"),
            mclVa(degree, "degree"),
            mclVa(frame, "frame")));
    /*
    * ud = get(hfig, 'UserData');
    */
    mlfAssign(&ud, mlfNGet(1, mclVv(hfig, "hfig"), _mxarray73_, NULL));
    /*
    * axhdlnp = ud.handles.ax(1);
    */
    mlfAssign(
        &axhdlnp, mlfIndexRef(mclVv(ud, "ud"), ".handles.ax(?)", _mxarray185_));
    /*
    * axhdlfp = ud.handles.ax(2);
    */
    mlfAssign(
        &axhdlfp, mlfIndexRef(mclVv(ud, "ud"), ".handles.ax(?)", _mxarray155_));
    /*
    * hdn = get(axhdlnp, 'Children');
    */
    mlfAssign(&hdn, mlfNGet(1, mclVv(axhdlnp, "axhdlnp"), _mxarray226_, NULL));
    /*
    * hdf = get(axhdlfp, 'Children');
    */
    mlfAssign(&hdf, mlfNGet(1, mclVv(axhdlfp, "axhdlfp"), _mxarray226_, NULL));
    /*
    * set(hdn, 'YData', s);
    */
    mclAssignAns(
        &ans, mlfNSet(0, mclVv(hdn, "hdn"), _mxarray228_, mclVv(s, "s"), NULL));
    /*
    * set(hdf, 'YData', f);
    */
    mclAssignAns(
        &ans, mlfNSet(0, mclVv(hdf, "hdf"), _mxarray228_, mclVv(f, "f"), NULL));
    mxDestroyArray(hfig);
    mxDestroyArray(s);
    mxDestroyArray(f);
    mxDestroyArray(ud);
    mxDestroyArray(axhdlnp);
    mxDestroyArray(axhdlfp);
    mxDestroyArray(hdn);
    mxDestroyArray(hdf);
    mxDestroyArray(ans);
    mxDestroyArray(frame);
    mxDestroyArray(degree);
    mxDestroyArray(noise);
    mxDestroyArray(y);
    mclSetCurrentLocalFunctionTable(save_local_function_table_);
    /*
    *
    */
}

static mxArray * Mnfilter_ecg(int nargout_, mxArray * L) {
    mexLocalFunctionTable save_local_function_table_
        = mclSetCurrentLocalFunctionTable(&_local_function_table_nfilter);
    mxArray * x = NULL;

```

```

mxArray * slope = NULL;
mxArray * m = NULL;
mxArray * i = NULL;
mxArray * d = NULL;
mxArray * a = NULL;
mxArray * d0 = NULL;
mxArray * a0 = NULL;
mclCopyArray(&L);
/*
 * % ЭКГ генератор.
 * % x = есг (L)
 * % Производит piecewise линейный ECG сигнал длины L
 * % Должен пост-пригладить это с более гладким N-пунктом:
 * % y = sgolayfilt (x, d, F), обычно с d=0, и N=3,5,9, и т.д.
 * a0 = [0,1,40,1,0,-34,118,-99,0,2,21,2,0,0,0]; % Шаблон
 */
mlfAssign(&a0, _mxarray230_);
/*
 * d0 = [0,27,59,91,131,141,163,185,195,275,307,339,357,390,440];
 */
mlfAssign(&d0, _mxarray232_);
/*
 * a = a0 / max(a0);
 */
mlfAssign(
    &a,
    mclMrdivide(mclVv(a0, "a0"), mlfMax(NULL, mclVv(a0, "a0"), NULL, NULL)));
/*
 * d = round(d0 * L / d0(15)); % Масштаб их, чтобы
соответствовать в длине L
 */
mlfAssign(
    &d,
    mlfRound(
        mclMrdivide(
            mclMtimes(mclVv(d0, "d0"), mclVa(L, "L")),
            mclIntArrayRef1(mclVv(d0, "d0"), 15))));
/*
 * d(15)=L;
 */
mclIntArrayAssign1(&d, mclVa(L, "L"), 15);
/*
 *
 * for i=1:14,
 */
{
    int v_ = mclForIntStart(1);
    int e_ = 14;
    if (v_ > e_) {
        mlfAssign(&i, _mxarray173_);
    } else {
        /*
         * m = d(i) : d(i+1) - 1;
         * slope = (a(i+1) - a(i)) / (d(i+1) - d(i));
         * x(m+1) = a(i) + slope * (m - d(i));
         * end
         */
        for (; ; ) {
            mlfAssign(
                &m,
                mlfColon(
                    mclIntArrayRef1(mclVv(d, "d"), v_),
                    mclMinus(
                        mclIntArrayRef1(mclVv(d, "d"), v_ + 1), _mxarray185_),

```

```

        NULL));
    mlfAssign(
        &slope,
        mclMrdivide(
            mclMinus(
                mclIntArrayRef1(mclVv(a, "a"), v_ + 1),
                mclIntArrayRef1(mclVv(a, "a"), v_)),
            mclMinus(
                mclIntArrayRef1(mclVv(d, "d"), v_ + 1),
                mclIntArrayRef1(mclVv(d, "d"), v_))));
    mclArrayAssign1(
        &x,
        mclPlus(
            mclIntArrayRef1(mclVv(a, "a"), v_),
            mclMtimes(
                mclVv(slope, "slope"),
                mclMinus(
                    mclVv(m, "m"), mclIntArrayRef1(mclVv(d, "d"), v_))),
            mclPlus(mclVv(m, "m"), _mxarray185_));
    if (v_ == e_) {
        break;
    }
    ++v_;
}
    mlfAssign(&i, mlfScalar(v_));
}
}
mclValidateOutput(x, 1, nargout_, "x", "nfilter/ecg");
mxDestroyArray(a0);
mxDestroyArray(d0);
mxDestroyArray(a);
mxDestroyArray(d);
mxDestroyArray(i);
mxDestroyArray(m);
mxDestroyArray(slope);
mxDestroyArray(L);
mclSetCurrentLocalFunctionTable(save_local_function_table_);
return x;
/*
 *
 *
 */
}

```