

УДК 681.3.06.14

Л.С. Файнзильберг

Инструментальная система для экспериментальной оценки эффективности алгоритмов предварительной обработки сигналов сложной формы

Рассмотрены особенности решения задачи фильтрации сложных сигналов, искаженных помехами различного типа. Предложена архитектура инструментальной системы, ориентированная на экспериментальную проверку эффективности алгоритмов предварительной обработки искаженных сигналов. Приведен пример практического применения инструментальной системы.

Features of solving the problems of filtration of the complex-form signals deformed by the various-type disturbances are considered. The architecture of the tool is suggested directed to the experimental check of the efficiency of the algorithms of preliminary processing of deformed signals. An example of the practical application of the tool is given.

Розглянуто особливості розв'язання задачі фільтрації складних сигналів, спотворених завадами різного типу. Запропоновано архітектуру інструментальної системи, орієнтованої на експериментальну перевірку ефективності алгоритмів попередньої обробки спотворених сигналів. Наведено приклад практичного застосування інструментальної системи.

Введение. Диагностика состояний технических и биологических объектов часто сводится к анализу и интерпретации сигналов сложной формы, порождаемых объектом в процессе функционирования [1]. Состояние объекта оценивают по амплитудно-временным параметрам, характеризующим форму специфических фрагментов сигнала.

Классическими примерами таких сигналов являются электрокардиограммы (ЭКГ), магнитокардиограммы, реограммы и другие биологические сигналы, локальные фрагменты которых несут информацию о состоянии сердечно-сосудистой системы живого организма [2, 3]. В металлургии, минералогии, петрографии и химии широко используют так называемый метод термического анализа [4], в соответствии с которым физико-химические свойства многокомпонентного материала определяют по форме характерных фрагментов сигнала, несущего информацию об изменении температуры образца в процессе его кристаллизации. Список подобных примеров можно продолжить.

Особенность перечисленных сигналов состоит в том, что диагностические признаки локализованы на относительно небольших фрагментах области определения сигнала. Поэтому

классические методы неэффективны для анализа и интерпретации таких сигналов. Например, коэффициенты функционального ряда, построенного на всей области определения, почти нечувствительны к вариациям амплитудно-временных параметров, характеризующих форму информативных фрагментов сигнала [5]. В то же время именно такие параметры несут диагностическую информацию о текущем состоянии объекта.

Компьютерная обработка сигналов сложной формы часто требует разработки нетривиальных алгоритмов, которые не всегда удается получить формальными методами. Поэтому конструктор прикладной информационной технологии (ИТ) должен на модельных и реальных данных проводить многочисленные эксперименты по оценке эффективности того или иного алгоритма прежде чем включить его в состав системы [6].

В реальных условиях типична ситуация, когда объект порождает сигнал на низких уровнях амплитуды. Для регистрации таких сигналов используются высокочувствительные датчики, которые, помимо полезного сигнала, фиксируют электрические и магнитные сигналы от посторонних источников. Поэтому первый этап обработки сигнала направлен на восста-

новление формы полезного сигнала по искаженной реализации.

Цель статьи – предложить с позиций объектно-ориентированного подхода концепцию построения инструментальной системы, ориентированной на экспериментальное исследование компьютерных алгоритмов предварительной обработки сигналов с локально сосредоточенными признаками.

Особенности фильтрации сигналов сложной формы

Предварительную обработку сигналов чаще всего связывают с выделением полезного сигнала $y_0(t)$ по реализации $y(t) = y_0(t) + h(t)$, искаженной аддитивным возмущением $h(t)$.

Согласно общей теории обнаружения сигналов [7] оптимальным фильтром для обнаружения сигнала известной формы на фоне белого гауссовского шума является параллельный набор согласованных фильтров. Например, при противофазной передаче двух сигналов $y_{01}(t)$ и $y_{02}(t) = -y_{01}(t)$ достаточно иметь только один согласованный фильтр.

Если шумы гауссовы, но не «белые», оптимальный фильтр становится значительно более сложным и для точного определения его структуры требуется решение соответствующего интегрального уравнения [7]. В случае бесконечного времени наблюдения оптимальный приемник представляет собой каскадное соединение предварительно «отбеливающего» и согласованного фильтров, причем частотная передаточная функция «отбеливающего» фильтра инверсна функции спектральной плотности шумов, а согласованный фильтр должен быть предварительно настроен на принимаемый полезный сигнал [8]. В тех же случаях, когда шум не является гауссовым, ситуация еще более усложняется: оптимальный (по критерию максимума правдоподобия) фильтр становится нелинейным и его структуру сложно реализовать [9].

Таким образом, даже если форма полезного сигнала $y_0(t)$ заранее известна, построение оптимального фильтра для выделения $y_0(t)$ на

фоне аддитивной помехи $h(t)$ представляет собой далеко не простую задачу, которая имеет формальное решение лишь при определенных предпосылках о модели $h(t)$.

Если функция $y_0(t)$ неизвестна, но может быть описана некоторой параметрической моделью $\mathcal{F}_0(t, \theta)$, задачу фильтрации часто сводят к оценке вектора θ параметров по наблюдаемой реализации. Для дискретного сигнала $y[k], k = 1, 2, 3, \dots$ получение такой оценки возможно на базе рекурсивных процедур, основанных на теории оптимальной фильтрации Калмана [10]. При определенных предпосылках о сигнале и помехе такие процедуры оптимальны в смысле минимума среднеквадратичного отклонения $\mathcal{F}_0(t)$ от $y_0(t)$.

Однако такие процедуры вовсе не гарантируют сохранения формы полезного сигнала на локальных фрагментах области определения и поэтому непригодны для обработки сигналов с локально сосредоточенными признаками. Очевидно, что в основу построения эффективных вычислительных процедур фильтрации сигналов с локально сосредоточенными признаками, должен быть положен иной критерий оптимальности, нежели среднеквадратическое отклонение сигнала на всей области определения. Например, минимум среднеквадратического отклонения сигнала на информативных фрагментах или, еще лучше, минимум отклонения значений диагностических признаков, сосредоточенных на таких фрагментах.

Поскольку не существует универсальной теории, позволяющей формализовать синтез оптимальных фильтров, основанных на таких критериях, не остается ничего иного, как ввести некоторые предположения о модели помехи и методом проб и ошибок искать приемлемые процедуры, обеспечивающие максимально возможное подавление аддитивной помехи при минимальных искажениях значений локально сосредоточенных диагностических признаков.

Естественно, что подтверждением эффективности (прагматической «оптимальности») таких процедур будут уже не формальные при-

емы, а результаты численного моделирования, что вполне допустимо при построении прикладных ИТ.

Характерные типы внешних возмущений

При регистрации сигналов с локально сосредоточенными признаками могут наблюдаться различные типы внешних возмущений. Анализ практических задач позволяет выделить наиболее характерные типы помех, с которыми чаще всего приходится сталкиваться конструктору прикладных ИТ (рис. 1).



Рис. 1. Характерные типы помех

По соотношению ширины спектров помехи и полезного сигнала различают узко- и широкополосные частотные помехи. Естественно, что одна и та же помеха по отношению к одному сигналу может быть узкополосной, а по отношению к другому — широкополосной.

Особый случай составляют так называемые сосредоточенные частотные помехи, которые могут наблюдаться в очень узкой полосе частот, например, сетевые помехи на частоте 50 Гц и кратных частотах. Сложность подавления таких помех на основе узкополосных заградительных (режекторных) фильтров существенно возрастает, если гармоническая помеха сосредоточена в диапазоне «полезных» частот.

К нестационарной помехе можно отнести низкочастотный дрейф (смещение) базового уровня сигнала, например, смещение изолинии ЭКГ.

При решении прикладных задач случайные помехи часто рассматривают как дискретно заданный белый шум — последовательность независимых одинаково распределенных величин. Обычно предполагается, что математическое ожидание такой помехи равно нулю, а дисперсия ограничена некоторой величиной.

Импульсной называется помеха малой длительности, которая в общем случае представ-

ляет собой одиночные выбросы, случайно распределенные по времени и амплитуде. К подобному типу помех можно отнести и резкие изменения базового уровня сигнала («скачки»), происходящие в случайные моменты времени.

Можно предположить, что перечисленные типы помех наблюдаются одновременно и независимо искажают полезный сигнал с локально сосредоточенными признаками. Поэтому будем рассматривать модель помехи как сумму частных моделей

$$h(t) = h_1(t) + h_2(t) + \dots + h_F(t) \quad (1)$$

и строить процедуру обработки сигнала в виде последовательности F фильтров, каждый из которых ориентирован на подавление помех определенного типа (рис. 2).

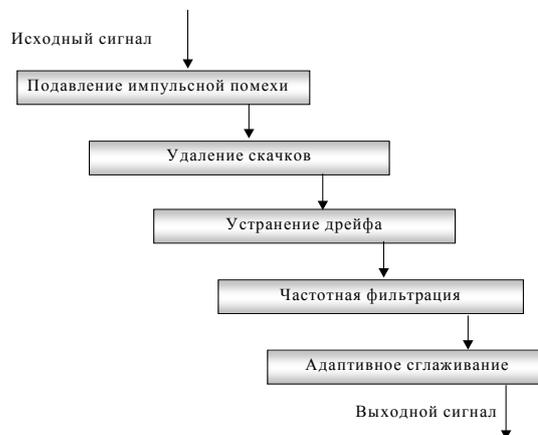


Рис. 2. Каскадная фильтрация сигнала

Архитектура инструментальной системы

Основная идея, которую предлагается положить в основу построения инструментальной системы, состоит в том, что с ее помощью конструктор прикладной ИТ должен иметь возможность

- *проводить* необходимые эксперименты для выбора и оптимальной настройки вычислительных процедур, реализующих отдельные стадии обработки сигнала;
- *реализовать* технологическую цепочку алгоритмов обработки, используя готовые вычислительные компоненты самой инструментальной среды;
- *расширять* при необходимости состав компонентов инструментальной системы (рис. 3).

Заметим, что открытость инструментальной системы и удобство ее развития является существенным требованием, поскольку в соответствии с расширенным толкованием известной теоремы Курта Геделя [11] невозможно в рамках компьютерных наук построить дедуктивную систему доказательства полноты базового набора компонент.



Рис. 3. Концептуальная идея инструментальной системы

Будем рассматривать структуру ИТ обработки сигнала как технологическую цепочку конечного числа N_Q вычислительных процедур (элементарных обработок), посредством которых вектор $Y = (y[1], \dots, y[K])$ дискретных значений наблюдаемого сигнала последовательно преобразуется в выходной продукт – оценку \mathcal{E} текущего состояние объекта:

$$D_R^{(1)} = A_1(\alpha_1, Y) \rightarrow \dots \rightarrow D_R^{(i)} = A_i(\alpha_i, D_S^{(i)}) \rightarrow \dots \rightarrow D_R^{(N_Q)} = A_{N_Q}(\alpha_{N_Q}, D_S^{(N_Q)}) = \mathcal{E} \quad (2)$$

где $A_i(\cdot)$ – алгоритм преобразования исходных данных $D_S^{(i)}$ в результат $D_R^{(i)}$ обработки на i -й стадии, α_i – параметры настройки алгоритма, причем в общем случае под исходными данными i -й стадии понимается не только резуль-

тат $D_R^{(i-1)}$, но и любые предшествующие ему результаты.

Если структура (2) определена и конструктор ИТ имеет полную информацию о каждой i -й стадии обработки $D_R^{(i)} = A_i(D_S^{(i)})$, то синтез ИТ сводится к традиционным этапам жизненного цикла разработки программного обеспечения. Однако опыт показывает, что при синтезе новых ИТ основное время затрачивается не столько на программную реализацию вычислительных алгоритмов, сколько на экспериментальные исследования, направленные на поиск подходящих алгоритмов и их оптимальную настройку.

Определим с позиций объектно-ориентированного подхода (ООП) главные программные классы, на основе которых может быть реализовано ядро инструментальной системы, удовлетворяющей сформулированным принципам.

В соответствии с методологией ООП данные и методы их обработки следует инкапсулировать в единый объект. Следуя выражению (1), для подавления внешних искажений требуется реализовать цепочку элементарных стадий обработки (фильтров), направленных на подавление аддитивных помех определенного типа. Поэтому представляется естественным проводить декомпозицию системы с точки зрения этих стадий, представив подсистему предварительной обработки в виде последовательности взаимодействующих объектов элементарной обработки (ЭО).

В данном случае предполагается, что каждый объект ЭО

- реализует отдельный алгоритм фильтрации;
- при необходимости отображает результат на экране монитора;
- выполняет роль приемника и/или источника информации при реализации каскадного алгоритма фильтрации.

Для реализации такого подхода в рамках ООП необходимо разработать абстрактный класс, реализующий в самых общих чертах модель ЭО. В этом классе входные и выход-

ные данные образуют единый блок данных, а методы обработки и визуализации – единый блок методов. Далее следует построить иерархию наследников класса, в которых последовательно конкретизируются представления о данных, методах их обработки и визуализации. В результате на базе производных классов могут быть разработаны конкретные экземпляры объектов ЭО, реализующие отдельные стадии фильтрации и визуализацию результатов.

Отсюда следует, что абстрактный класс ЭО должен содержать как минимум два поля (свойства) – «Исходные данные (ИД)» и «Результат обработки (РО)», а также виртуальные методы работы с этими полями – «Обработать ИД», «Отобразить РО» и «Передать РО».

Понятно, что для организации взаимодействия различных объектов ЭО необходим внешний координатор, который также желательно реализовать в виде отдельного объекта – менеджера системы (МС). Такой менеджер должен обеспечить

- порождение требуемого экземпляра ЭО и регистрацию его в своем поле;
- передачу ЭО исходных данных;
- вызов метода ЭО «Выполнить обработку»;
- передачу ЭО команды на отображение результата (при необходимости).

На рис. 4 показан фрагмент укрупненной схемы взаимодействия (диаграммы последовательности в терминах *UML*) объекта МС с двумя объектами ЭО, один из которых служит источником информации, а другой – приемником.

Рассмотренная модель построения инструментальной системы представляется достаточно простой и вполне логичной с точки зрения объектно-ориентированного подхода. Однако при более глубоком анализе оказывается, что эта модель не обладает требуемой гибкостью. Дело в том, что в общем случае для выполнения i -й стадии обработки может потребоваться не только результат $(i-1)$ -й стадии, но и какой-либо другой из предшествующих стадий. В таких ситуациях приходится оставлять

активным объект ЭО только с целью сохранения результата обработки. Понятно, что такая организация системы не может быть признана удовлетворительной.

Кроме того, разнообразие алгоритмов, которые могут потребоваться для реализации отдельных стадий ИТ, существенно больше, чем разнообразие методов визуализации результатов обработки.

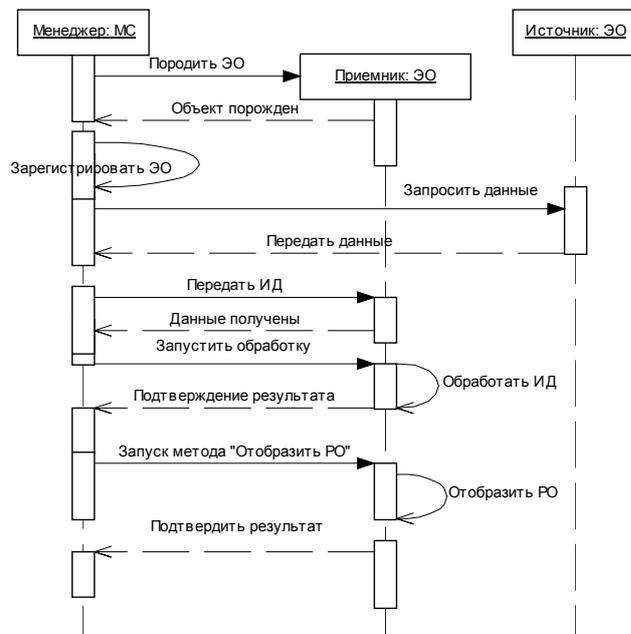


Рис. 4. Диаграмма взаимодействия менеджера системы с объектами ЭО

Принимая во внимание эти факты, предлагается несколько иначе взглянуть на декомпозицию системы: вычленив данные и методы их визуализации из класса ЭО и рассматривать их как независимый класс – модель носителя данных (МНД), но оставить за ЭО возможность ссылаться на нужный экземпляр объекта МНД (рис. 5).

При объединении в одном объекте данных и методов их визуализации открываются возможности:

- получать доступ к данным через экранные графические образы, отображающие эти данные;
- обеспечивать необходимую привязку графического образа, визуализирующего результат обработки, к графическому образу исходных данных.

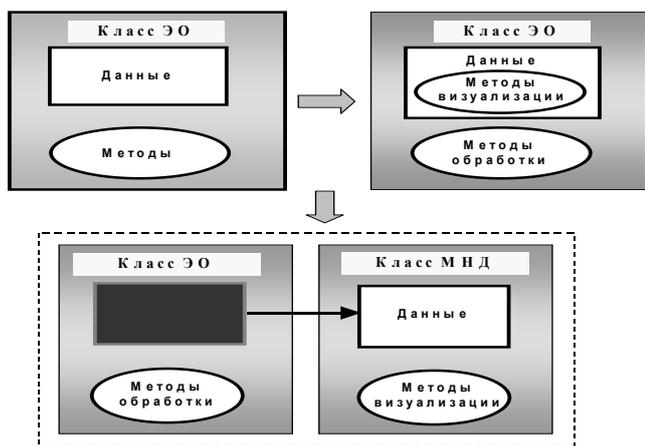


Рис. 5. Декомпозиция класса элементарной обработки данных (ЭО)

Для решения этой задачи в рамках ООП предлагается разработать абстрактный класс **МНД**, реализующий обобщенную модель носителя данных, а затем построить иерархию наследников этого класса, и в них последовательно конкретизировать представления о данных и методах их визуализации.

При таком подходе каждый отдельный экземпляр наследника **МНД** может выполнять роль либо источника, либо приемника данных, а совокупность порожденных объектов **МНД** образует виртуальную базу данных, управление которой следует поручить **МС**.

Поскольку экземпляры объектов **МНД** должны взаимодействовать с объектами **ЭО**, представляется целесообразным метод порождения объектов **МНД** «делегировать» от менеджера непосредственно к объектам **ЭО**, сохранив за менеджером лишь функции регистрации. Такое распределение координационных функций может быть реализовано передачей соответствующих ссылок (указателей) на нужные типы объектов.

Предлагается также сосредоточить процедуры, реализующие вычислительные алгоритмы обработки данных на разных стадиях, в библиотеке вычислительных процедур (**БВП**), которая может разрабатываться независимо от структуры проектируемой системы, оставив в объекте **ЭО** лишь возможность ссылаться на конкретную вычислительную процедуру.

В результате приходим к трехуровневой схеме организации управления объектами системы (рис. 6). Менеджер системы (**МС**) порождает и взаимодействует с объектами **ЭО**, которые, в свою очередь, порождают объекты **МНД**. При этом объекты **ЭО** вызывают требуемую вычислительную процедуру из библиотеки **БВП**, передают результат ее выполнения в объекты **МНД**, а последние обеспечивают хранение и, при необходимости, отображение результата обработки на экране монитора.

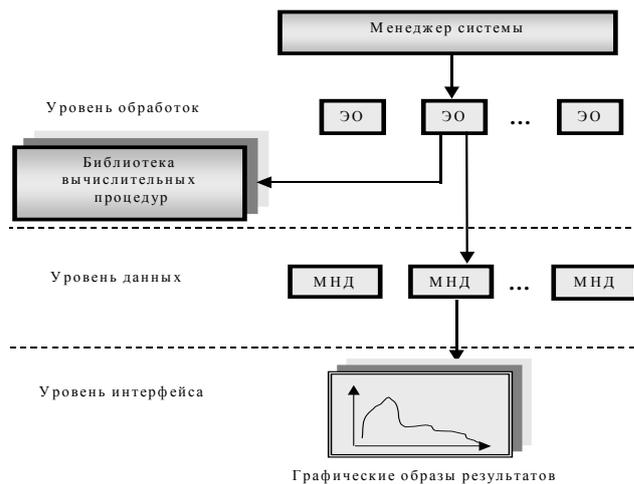


Рис. 6. Трехуровневая схема построения инструментальной системы

Укрупненная модель взаимодействия **МС** с объектами элементарной обработки (**ЭО**) и носителя данных (**МНД**) показана на рис. 7. В отличие от ранее рассмотренной схемы (рис. 4), теперь уже линия жизни объекта **ЭО** всегда завершается после выполнения очередной стадии обработки. Активным остается объект **МНД**-приемник и, возможно, объект **МНД**-источник, если содержащиеся в нем данные потребуются на последующих стадиях обработки.

Объектно-ориентированные модели ядра инструментальной системы

В основу построения обобщенной **МНД** положена идея разработки абстрактного класса, на базе которого создаются объекты инструментальной системы предварительной обработки сигналов. Эти объекты должны обеспечивать:

- хранение содержательных данных, необходимых для организации элементарных вычислительных процедур на различных стадиях предварительной обработки сигнала;
- отображение результатов обработки отдельных стадий в виде экранных графических образов и текстовой информации;
- доступ к данным через их графические образы;
- модификацию графических образов (перемещение, масштабирование, изменение типа и др.) (рис. 8).

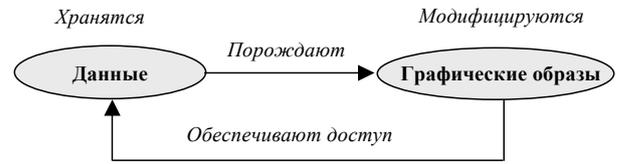


Рис. 8. Концептуальная идея построения МНД

Идея построения обобщенной модели обработки данных (**МОД**) состоит в разработке абстрактного класса с универсальными полями, свойствами и методами, инвариантными относительно конкретной процедуры обработки. При этом **МОД** содержит виртуальный метод «пустой» обработки, который доопределяется в наследниках этого класса (рис. 9).

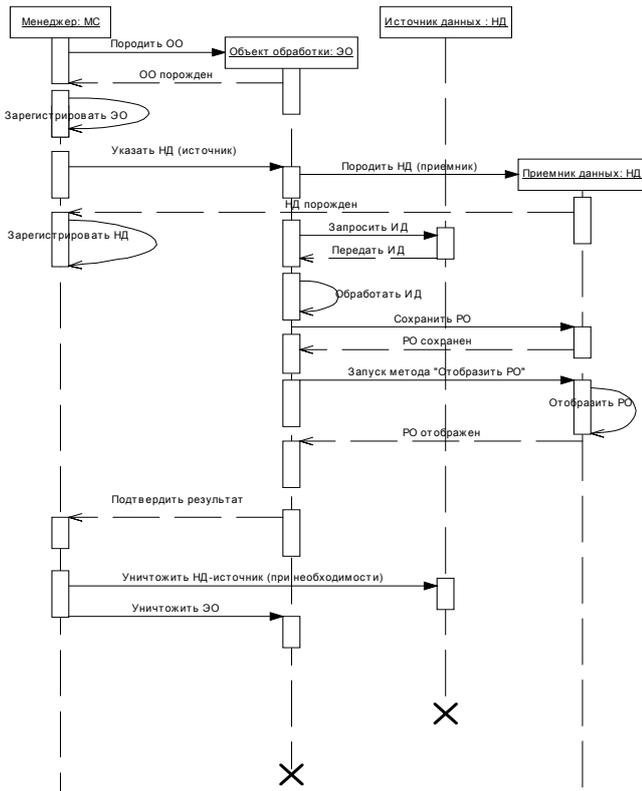


Рис. 7. Диаграмма взаимодействия менеджера системы (МС) с объектами элементарной обработки (ЭО) и носителя данных (НД)

С этой целью предлагается рассматривать МНД в виде тройки

$$\text{МНД} = \langle D_C, D_G, G_M \rangle, \quad (3)$$

где D_C – поля содержательных данных, D_G – поля графических данных, G_M – методы отображения и модификации графических образов. Упрощенная структура абстрактного класса, реализующего эту модель, рассмотрена в работе [12].

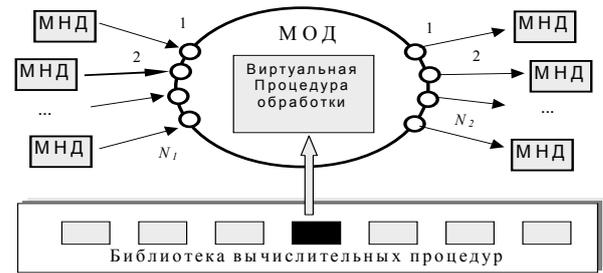


Рис. 9. Упрощенная схема компоновки элементарной стадии обработки: МОД – модель обработки данных, МНД – модель носителя данных

Абстрактный класс **МОД** позволяет унифицировать понятие элементарной обработки данных, упорядочить процесс создания программных компонент, реализующих отдельные стадии обработки, и тем самым организовать удобный интерфейс взаимодействия между активными экземплярами (объектами) класса **МНД** и конкретной процедурой обработки.

Класс **МОД** поддерживает унифицированный интерфейс вызова любой вычислительной процедуры:

$$\langle \text{Имя} \rangle (I_1, \dots, I_{N_1}; L_1^{(I)}, \dots, L_{N_1}^{(I)}; O_1, \dots, O_{N_1}; L_1^{(O)}, \dots, L_{N_2}^{(O)}; \alpha_1, \dots, \alpha_{N_{1\alpha}}; \text{Mess}), \quad (4)$$

где

$I_1, \dots, I_{N_1}; O_1, \dots, O_{N_1}$ – указатели на массивы входных и выходных данных;

$L_1^{(I)}, \dots, L_{N_1}^{(I)}; L_1^{(O)}, \dots, L_{N_2}^{(O)}$ – число элементов в соответствующих массивах;

$\alpha_1, \dots, \alpha_{N_{1\alpha}}$ – настроечные параметры процедуры обработки;

Mess – строка сообщения об исключительной (аварийной) ситуации.

Такой интерфейс позволяет распараллелить работу программистов, участвующих в создании инструментальной системы. Прикладные программисты могут сосредоточиться на своей главной задаче – реализации эффективных вычислительных процедур обработки данных, не заботясь об организации диалога с пользователем и методах графического представления результатов, а системные программисты – рассматривать процедуру обработки как некий «черный ящик».

На базе производных классов **МОД** реализовать широкий набор вычислительных компонент, обеспечивающих предварительную обработку сигнала с локально сосредоточенными признаками. Оценка эффективности этих компонент проводилась по схеме (рис. 10). Для этого в системе обеспечена возможность «наложения» гармонических, случайных и импульсных помех на реальные и тестовые сигналы, а также имитации различных других искажений, с которыми приходится иметь дело при решении практических задач обработки сигналов.

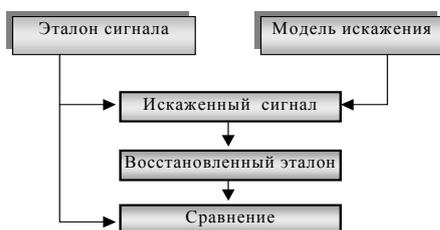


Рис. 10. Общая схема организации экспериментов

Доступ к данным осуществляется по принципу «*Point and Select*»: для доступа к данным достаточно указать мышью на соответствующий график в рабочем окне, а для вызова требуемой вычислительной процедуры – выбрать соответствующий пункт меню. Тем самым конструктору ИТ предоставляется возможность реализации цепочки вычислительных процедур из готовых компонент самой инструментальной системы (рис. 11).

Удобной функцией системы является режим отображения, при котором пользователь имеет возможность в одном окне перемещать и масштабировать один график на фоне остальных.

Эта функция позволяет не только численно, но и визуально оценивать эффективность процедур фильтрации. Полезной находкой системы является также подход, обеспечивающий интерактивную настройку параметров вычислительных процедур с использованием идей когнитивной компьютерной графики.

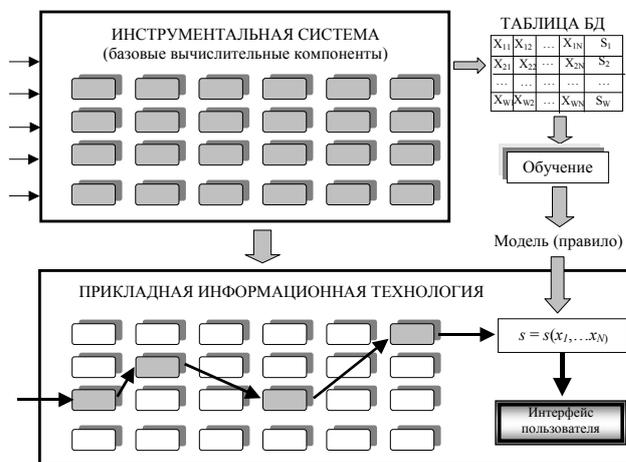


Рис. 11. Организация цепочки вычислительных процедур

На рис. 12 показано рабочее окно системы при настройке процедуры сглаживания зашумленного сигнала (верхний график). Благодаря «виртуальной» связи органа управления и графического образа, отображающего результат обработки (нижний график), пользователь может визуально определить «оптимальное» значение параметра настройки вычислительной процедуры. При нажатии клавиши ОК найденное таким образом значение параметра настройки сохраняется в поле соответствующего объекта носителя данных. Аналогично осуществляется

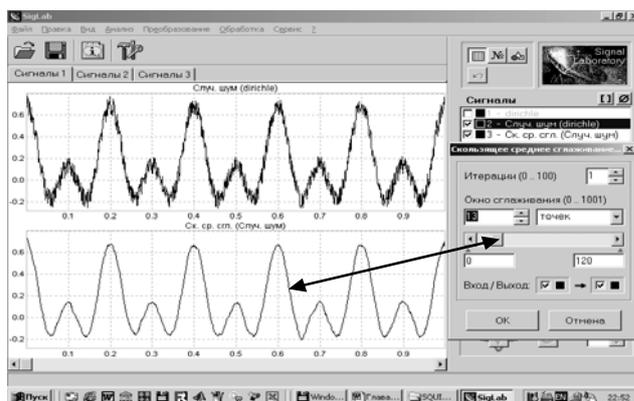


Рис. 12. Интерактивная настройка параметров вычислительного компонента

настройка и в других случаях, когда отсутствуют формальные методы определения оптимальных значений настроечных параметров.

Практические результаты

Инструментальная система позволила провести необходимые исследования при конструировании целого ряда прикладных систем обработки сигналов сложной формы. В частности, с использованием этой системы создана компьютерная система ФАЗАГРАФ [1], предназначенная для оперативной оценки функционального состояния сердечно-сосудистой системы человека.

Система ФАЗАГРАФ представляет собой портативный микропроцессорный сенсор, обеспечивающий регистрацию электрокардосигнала с пальцев рук испытуемого, и компьютерную программу, реализующую оригинальный метод обработки сигнала в фазовом пространстве координат (рис. 13).



Рис. 13. Компьютерная система ФАЗАГРАФ (сенсор и диск с программным обеспечением)

Поскольку система предназначена для регистрации и обработки физиологического сигнала низкого уровня, снимаемого с пальцев пользователя в любых условиях, в том числе бытовых, исходный сигнал был существенно искажен помехами различного типа. В то же время построение фазового портрета ЭКГ, необходимого для интерпретации сигнала в соответствии с методом, предложенным в [13], требует использования процедуры численного дифференцирования, которая может быть применена только к сигналу, «очищенному» от помех. Поэтому одна из первоочередных задач состояла в выборе и настройке процедур каскадной фильтрации, обеспечивающих восстановление полезного сигнала по искаженной реализации.

Для подавления импульсной помехи применен известный прием ранговой фильтрации. Фильтрованное значение сигнала $\mathfrak{F}[k]$ в k -й момент времени определяется как центральное значение (медиана) упорядоченного массива $y[k-m], \dots, y[k], \dots, y[k+m]$ в скользящем окне с нечетным числом точек. Для устранения одиночных импульсных помех достаточно положить $m = 1$. Численные эксперименты показали, что при $m > 5$ применение процедуры становится неэффективным из-за искажений фронтов полезного сигнала.

Для устранения искажений, вызванных резкими скачками базового уровня, строится профиль изменения среднего значения сигнала, вычисленный «левосторонним» и «правосторонним» окнами

$$\chi[k] = \frac{1}{W_0} \left(\sum_{j=1}^{W_0} y[k-j] - \sum_{j=1}^{W_0} y[k+j] \right), \quad (5)$$

где $W_0 = \text{const}$ – ширина окна. Затем определяется подмножество точек $\Omega_\chi = \{k \in [1, K] : \chi[k] > \chi_0\}$, где $\chi_0 = \text{const}$ – заданный порог. Если $\Omega_\chi \neq \emptyset$, то определяется значение точки $k^* = \arg \max_{k \in \Omega_\chi} \chi[k]$, в которой наблюдается резкий перепад сигнала, и вводится поправка

$$\tilde{y}[k] = \begin{cases} y[k], & \text{при } k < k^*, \\ y[k] + \chi[k^*], & \text{при } k \geq k^*. \end{cases} \quad (6)$$

Для устранения низкочастотного дрейфа используется уже другой вычислительный компонент. Дрейф предварительно оценивается по формуле

$$y_B[k] = \frac{1}{2W_B + 1} \sum_{j=-W_B}^{W_B} y[k-j], \quad k \in [W_B, K - W_B],$$

где $W_B = \text{const}$ – настраиваемый параметр, а затем массив следующим образом корректируется:

$$\tilde{y}[k] = y[k] - y_B[k] + \Delta_y, \quad (7)$$

где

$$\Delta_y = \frac{\max_{W_B \leq k \leq K - W_B} y[k] - \min_{W_B \leq k \leq K - W_B} y[k]}{2}. \quad (8)$$

Для подавления сосредоточенных частотных помех используется узкополосный режекторный фильтр, основанный на процедурах прямого и обратного дискретных преобразований Фурье (ДПФ) [14]. Отличительной особенностью фильтра является включение в него дополнительной поисковой процедуры, обеспечивающей уменьшение известного эффекта растекания спектра помехи.

Для реализации этой процедуры последовательно уменьшается число K_j обрабатываемых точек входного массива данных от начального значения K до $K - \delta_K$. Далее для каждого $K_j \in [K - \delta_K, K]$, вычисляются спектральные компоненты $C_n(K_j)$, $n = -K_j/2, \dots, 0, \dots, K_j/2$ (значения модулей ДПФ).

Затем определяется оптимальная длина массива K_{opt} по условию

$$K_{\text{opt}} = \arg \max_{K_j \in [K - \delta_K, K]} \left\{ \frac{\max_{n \in \Omega_h} C_n(k)}{\sum_{n \in \Omega_h} C_n(k)} \right\},$$

где Ω_h – множество номеров спектральных компонент с частотами, принадлежащими интервалу $[f_{\min}, f_{\max}]$, где предположительно сосредоточена гармоническая помеха и в качестве оценки частоты f_g помехи принимаем значение максимальной спектральной компоненты на интервале $[f_{\min}, f_{\max}]$, вычисленную при K_{opt} .

Последняя стадия каскадной фильтрации – подавление ограниченных по уровню случайных помех, для чего используется алгоритм адаптивного сглаживания [15].

Многочисленные эксперименты, проведенные с помощью разработанной инструментальной системы на модельных и реальных данных, подтвердили эффективность предложенных алгоритмов предварительной обработки сигналов сложной формы.

Пример каскадной фильтрации реальной ЭКГ (рис. 14, а), показывает, что после применения процедур устранения тренда (рис. 14, б), частотной фильтрации (рис. 14, в) и адаптивного сглаживания (рис. 14, г) удастся достаточно эффективно восстановить форму информативных фрагментов полезного сигнала,

тем самым обеспечив его дальнейший анализ и интерпретацию в прикладных ИТ, в частности в компьютерной системе ФАЗАГРАФ.

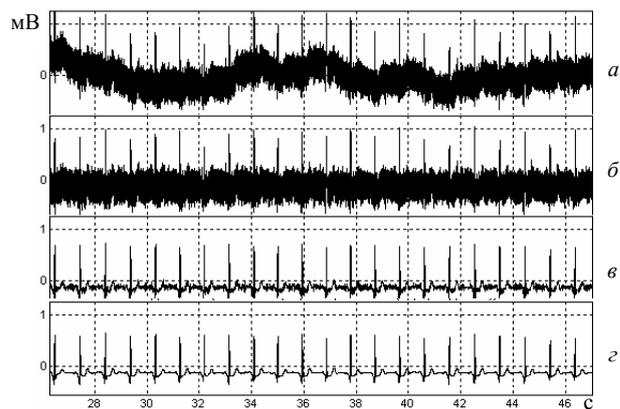


Рис. 14. Пример каскадной фильтрации реальной ЭКГ

Заключение. Разработанная инструментальная система содержит широкий набор вычислительных компонент обработки сигналов сложной формы. Это позволяет существенно ускорить трудоемкий процесс выбора и настройки вычислительных компонент, обеспечивающих восстановление сигналов сложной формы по искаженным реализациям, и повысить потребительские свойства новых ИТ анализа и интерпретации сигналов.

1. Бургер И.А. Техническая диагностика – М: Машиностроение, 1978. – 240 с.
2. Fainzilberg L.S. Nowa metoda interpretacji zapisu EKG w balaniach skriningowych oraz w opiece domowej // «Zdrowie publiczne». – 2005. – 115, – N 4. – P. 458–464.
3. Примин М.А., Недайвода И.В. Алгоритмы отображения и оценивания магнитокардиосигнала по двумерному распределению вектора плотности тока // УСиМ. – 2003. – № 5. – С. 19–25.
4. Берг Л.Г. Введение в теорию термографии. – М.: Наука, 1969. – 380 с.
5. Браверманн Э.М., Мучник И.Б. Структурные методы обработки эмпирических данных. – М.: Наука, 1983. – 492 с.
6. Файнзилберг Л.С. Синтез информационных технологий обработки сигналов в задачах косвенного контроля и диагностики // УСиМ. – 1998. – № 2. – С. 40–47.
7. Ziemer R.E., Peterson R.L. Digital communications and Spread Spectrum. – New York: MacMillan, 1985. – 386 p.

Окончание на стр. 53

8. *Милстайн Л.Б.* Методы подавления помех в системах радиосвязи с широкополосными сигналами // ТИИЭР. – 1988. – Т. 76. – № 6. – С. 19–36.
9. *Ogawa J., Cho S.J., Morinaga N., Namekawa T.* Optimum detection of Mary-RSK signal in the presence of CW interference // Trans. IEESE Japan. –1981. – **Е 64**. – Р. 800–806.
10. *Балакришнан А.* Теория фильтрации Калмана. – М.: Мир, 1988. – 168 с.
11. *Брюхович Е.И.* Плоды эволюции философии науки. Действительная роль экономики и супер-ЭВМ в жизни человечества на современной стадии развития цивилизации // Наука и науковедение. – 2001. – № 4. – С. 53–68.
12. *Файнзильберг Л.С.* Объектно-ориентированная модель данных в информационных технологиях обработки сигналов // УСиМ. – 1998. – № 3. – С. 37–41.
13. *Пат. № 24517* Україна, МКИ А61 В 5/024. Спосіб інтегральної оцінки поточного стану серцево-судинної системи людини / Л.С. Файнзильберг; Заявлено 21.05.97; Опубл. 30.10.98, Бюл. № 5. – 4 с.
14. *Файнзильберг Л.С.* Частотно-избирательная фильтрация в информационных технологиях обработки сигналов // УСиМ. – 2002. – № 2. – С. 54–59.
15. *Файнзильберг Л.С.* Адаптивное сглаживание шумов в информационных технологиях обработки физиологических сигналов // Математичні машини і системи. – 2002. – № 3. – С. 96–104.

Поступила ХХ.03.2007
Тел. для справок: (044) 411-6904 (Киев)
© Л.С. Файнзильберг, 2008

