УДК 615.8-7:004.94

Ю.В. Миргород

Національний технічний університет «Харківський політехнічний інститут», Харків

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПІДСИСТЕМИ ІДЕНТИФІКАЦІЇ БІОСИГНАЛІВ НА ОСНОВІ ПЕРЕТВОРЕННЯ ХОКА

У статті розглянута побудова програмної моделі підсистеми ідентифікації біосигналів. Наведено вибір інструментальних засобів і методології побудови програмного забезпечення.

Ключові слова: *фізіологічні квазіперіодичні сигнали (ФКС), перетворення Хока, прості ознак, діаграма класів, структура програмного забезпечення.*

Вступ

Постановка проблеми. Сучасні системи медичної діагностики використовують комп'ютерні комплекси, що містять у собі програмну реалізацію різних методів [1 – 6]. Авторами був запропонований метод обробки біосигналів на основі перетворення Хока [7 – 9], однак залишається відкритим питання побудови програмної моделі підсистеми ідентифікації біосигналів.

Аналіз літератури. Аналіз літератури дозволив вибрати засоби для створення програмного забезпечення для реалізації методу обробки ФКС на основі перетворення Хока.

Була проаналізована методологія створення програмного забезпечення[10].

Були проаналізовані існуючі бази даних[11] та принципи побудови баз даних.

У [12] розглянуті складові програмного забезпечення.

У [12] розглянуті існуючі засоби створення програмного забезпечення.

У [13] розглянуті принципи побудови UML-діаграм.

Мета статті – створення програмної моделі підсистеми обробки сигналів на основі перетворення Хока.

Основний матеріал

Розробка структури програмної моделі підсистеми обробки сигналів на основі перетворення Хока. Для створення підсистеми структурної ідентифікації біомедичних сигналів необхідно до запропонованого методу реалізувати програмне забезпечення (ПЗ), що задовольняє наступним вимогам:

–зручність уведення необхідної інформації;

–наочність результатів структурної ідентифікації;

–можливість накопичення досвіду при виділенні структурних елементів біомедичних сигналів одного типу;

–сумісність з іншими підсистемами з метою подальшої обробки отриманої інформації.

Як було зазначено у вимогах до програмного забезпечення, підсистема має працювати у режимі навчання і розпізнавання. Під навчанням будемо розуміти налаштування роботи підсистеми, а під режимом розпізнавання - власне процес структурної ідентифікації заданого біомедичного сигналу (рис. 1). Підсистема матиме 3 основні шари. Перший шар - це здобуття вихідного сингалу і перетворення його у спеціальний формат для подальшої роботи. Другий шар відповідатиме за навчання системи шляхом завдання еталону лікарем-оператором, завдання кроку квантування, вибір відведень сигналу тощо. Функцією третього шару є розпізнавання сигналу, тобто класифікацію структурних елементів на задані класи.

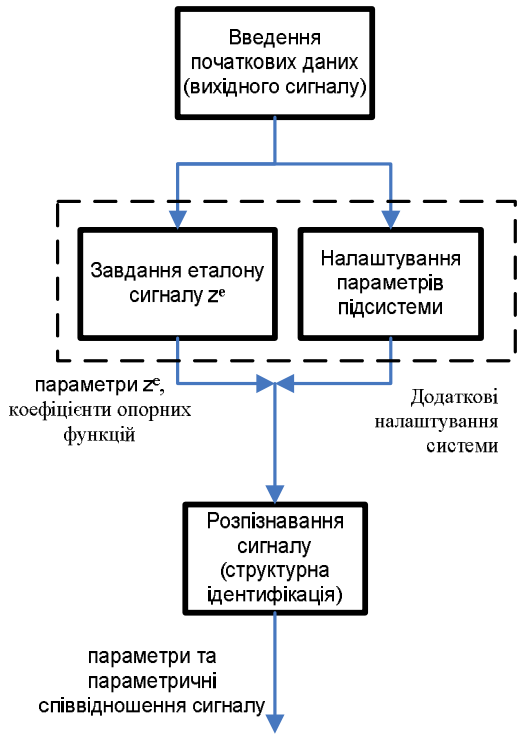

Рис. 1. Структурна схема підсистеми структурної ідентифікації біомедичних сигналів

Розробимо UML-діаграму головних класів підсистеми обробки біосигналів (рис. 2).

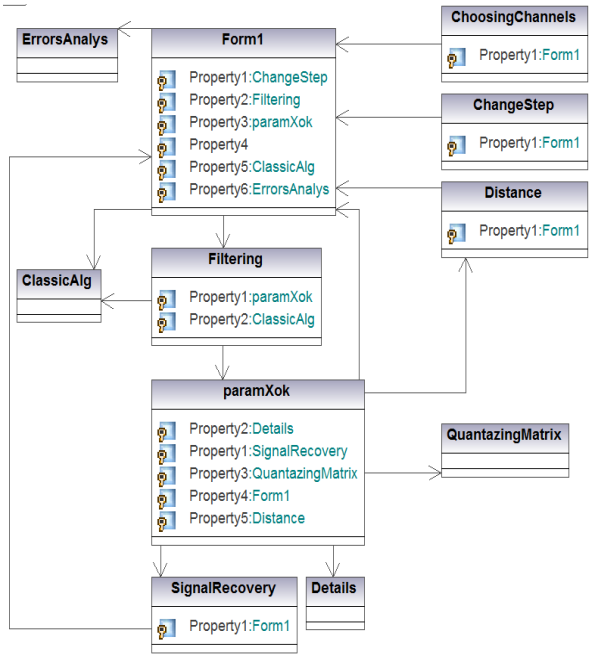

Рис. 2. Діаграма класів підсистеми структурної ідентифікації

Таким чином, сформулюємо вимоги до класів підсистеми структурної ідентифікації біомедичних сигналів:

–підсистема повинна працювати у режимі навчання. Для цього буде використовуватися клас *Form1, ChoosingChannels, ChangeStep*;

–підсистема повинна працювати у режимі розпізнавання. підсистема повинна виконувати класифікацію структурних елементів на задані класи незалежно від типу біомедичного сигналу, що досліджується. За це відповідатимуть класи *paramXok, Distance, Form1*;

–підсистема повинна мати простий і зручний інтерфейс, тому що може бути використана фахівцями в різних областях, що могли б самостійно за мінімальний час освоїти роботу з підсистемою. Для цього буде використано клас *Form1*;

–при роботі в діалоговому режимі необхідно забезпечити максимальна зручність уведення необхідної інформації і збереження отриманих результатів до наступного сеансу роботи. Для цього буде використано клас *Form1*;

–підсистема повинна мати можливість візуального представлення результатів структурної ідентифікації біомедичних сигналів. За це відповідатиме клас *Form1, Details, QuantazingMatrix*;

–підсистема має можливість побудови відновленого сигналу методом, описаним у розділах 2 і 3. Для цього буде створено клас *SignalRecovery*;

–для використання накопиченого досвіду при структурній ідентифікації біомедичних сигналів для виділення інформативних фрагментів різного типу сигналів необхідно формувати базу даних підсистеми. Для цього буде використано клас *Form1*;

–результати роботи підсистеми повинні бути доступні іншим підсистемам біомеханічної системи (див. рис. 1) для їхньої подальшої обробки. Для цього буде використано клас *Form1*;

–у цілому структура підсистеми повинна відповідати схемі, що представлена на рис. 1.

Організація вхідних і вихідних даних. Підсистема обробки та структурної ідентифікації біосигналів має на вході решітчасту функцію, що представляє собою відліки сигналів. Процедура обробки може виконуватися одразу після реєстрації сигналу чи незалежно від часу прийому сигналу. Таким чином, необхідно передбачити зберігання вихідної інформації.

Для цього було запропоновано створити базу даних, що містить у собі 2 таблиці:

1. Перша таблиця матиме інформацію про пацієнта.
2. Друга зберігатиме дані про сигнали, що асоційовані з пацієнтом.

Опишемо детальніше поля таблиць (табл. 1).

Таблиця 1

Опис атрибутів таблиць бази даних

Назва таблиці	Назва атрибуту	Тип	Участь у ключах	Опис
patient	id	int	PK	Номер пацієнта
	first_name	text		Фамілія
	middle_name	text		По батькові
	last_name	text		Ім’я
	age	int		Вік
	diagnosis	text		Діагноз
	notes	text		Додаткова інформація
Signals	id	uniqueidentifier	PK	Номер запису
	date	datetime		Дата запису
	value	xml		Значення сигналу та допоміжних даних (таких як функція диференціації відстані, розпізнані комплекси тощо)
	patient_id	int	FK	Номер пацієнта, з яким асоційовано сигнал
	sr_kompl	xml		Показники середнього комплексу(ів) сигналу

Вибір засобів для створення програмного забезпечення підсистеми структурної ідентифікації біомедичних сигналів. У даний час для створення програмного продукту існує ряд мов програмування високого рівня і програмних середовищ під різні операційні системи, такі як Mac OS, Unix, Windows і так далі. З огляду на сучасні вимоги до інтерфейсу програмного продукту, що розробляється, а також, приймаючи до уваги той факт, що найбільше поширення одержали операційні системи Windows, програмне забезпечення підсистеми доцільно створювати для використання в цих операційних системах.

Найбільш розповсюдженими на даний момент мовами програмування є C++/C#, Java, що є основою різних середовищ програмування. На базі мов C++/C# під Windows існують наступні середовища програмування:

–Microsoft Visual Studio;

–Inprise (Borland) C++ Builder.

На базі мови Java можна виділити наступні інтерактивні середовища програмування:

–Idea

–Eclipse

–NetBeans.

Виходячи з того, що найбільш поширеною мовою програмування на сьогодні є C#, було вирішено використати її.

На даний момент найбільше поширення одержало інтерактивне середовище програмування Microsoft Visual Studio.

Середовище Microsoft Visual Studio - це комбінація декількох найважливіших технологій:

–високопродуктивний компілятор у проміжний код і компілятор Just-in-time, що забезпечує високу продуктивність при оптимізації додатків для цільової платформи.

–об’єктно-орієнтована модель компонентів, основний упор якої робиться на максимальне повторне використання коду, що дозволяє розроблювачам будувати додатка досить швидко з заздалегідь підготовлених об'єктів, а також дає їм можливість створювати свої власні об'єкти для середовища Microsoft Visual Studio. Завдяки проміжному коду, що створює компілятор, додатки є мово-незалежними на платформі .NET Framework;

–візуальне (а, отже, і швидкісне) побудова додатків з програмних прототипів. Середовище Microsoft Visual Studio містить у собі повний набір візуальних інструментів для швидкісної розробки додатків (RAD - rapid application development), що підтримує розробку користувальницького інтерфейсу і підключення до корпоративних баз даних. GDI+ - бібліотека для для графічної роботи нового покоління; об'єкти керування даними, графічні об'єкти, об'єкти мультімедіа, діалоги й об'єкти керування файлами;

–засобу для побудови баз даних, що маштабуються, що дозволяють використовувати той самий додаток як для локального, так і для клієнт-серверного варіантів [11].

Таким чином, з огляду на переваги Microsoft Visual Studio, дана середовище програмування щонайкраще підходить для створення програмного забезпечення підсистеми структурою ідентифікації біомедичних сигналів, що задовольняє запропонованим вище вимогам.

Рівень представлення. Для зручності роботи з розглянутим біомедичним сигналом доцільно представити його у виді графіка. Виходячи з того, що одна реалізація сигналу може бути досить довгою (кілька десятків тисяч відліків функції), то при виводі таких сигналів стандартними засобами Microsoft Visual Studio в процесі роботи з ним можуть виникати значні затримки. Це обумовлено тим, що сигнал цілком міститься у відповідний графічний компонент, і при будь-якому зрушенні відбувається досить довге перемальовування даного компонента на екрані. Тому виникла необхідність у створенні власного компонента для виводу на екран сигналів будь-якої довжини.

При розробці компонента швидкого виведення графіки на дисплей монітора був врахований той факт, що засобами операційної системи Windows швидше за все виводиться бітова графіка. Виходячи з вище сказаного, виводиться не весь сигнал, а лише його ділянка. При цьому з цієї ділянки сигналу формується бітове зображення, що назвемо поточним кадром.

Завдання опорних точок виконується за допомогою маніпулятора «миша» та графічних можливостей системи (рис. 3).

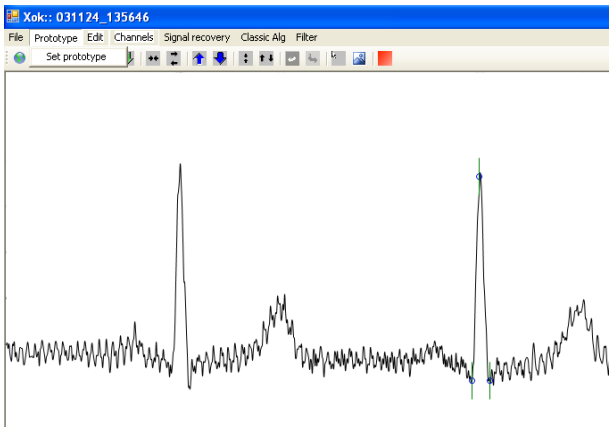

Рис. 3. Завдання опорних точок

Треба відзначити, що позиціонування опорної точки на графіку по вертикалі розраховується автоматично програмою, а по горизонталі – оператором вручну за рахунок установки опорних точок.

Виходячи з того, що розміщення опорних точок еталонного об'єкта біомедичного сигналу, що до-

сліджується, здійснюються вручну, то можливі помилки оператора. Тому в підсистемі передбачена можливість точної корекції місця розташування кожної опорної точки. Для цього передбачені кнопки відображення вісей координат та масштабування на панелі керування, що дозволяє відображати горизонтальну та вертикальну вісі та зміну масштабу відображення сигналу.

Таким чином, у підсистемі максимальна довжина біомедичного сигналу, що досліджується, і кількість розставлених опорних точок для завдання еталонного об'єкта обмежуються лише обсягом вільної динамічної пам'яті.

Програма надає можливість переглядати значення матриці квантування простору ознак. Результат показано на рис. 4.

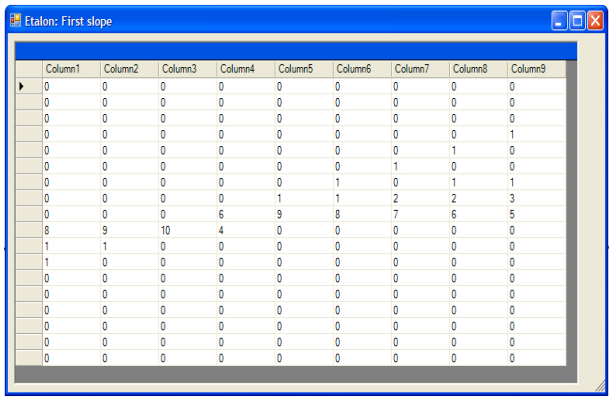

Рис. 4. Матриця квантування простору ознак

На рис. 5 зображено кардіограму та відповідну функцію диференціації відстані.

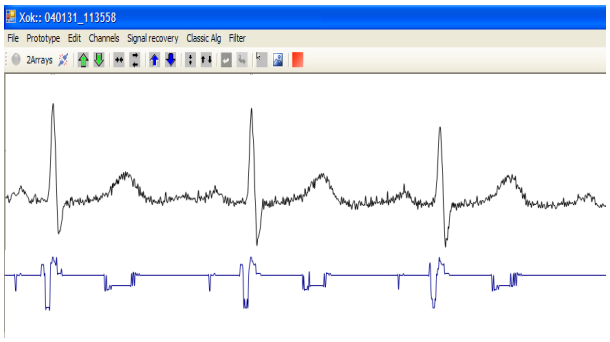

Рис. 5. Кардіограма та функція диференціації відстані

Висновки

Базуючись на роботі, що виконано, можна зробити наступні висновки:

1. Сформульовані вимоги до програмного забезпечення підсистеми ідентифікації біосигналів.

2. Розроблено трирівневу структуру підсистеми: рівень даних, бізнес-логіки та відображення.

3. Проведено аналіз програмних засобів для створення програмного забезпечення.

4. Розроблена програма роботи підсистеми ідентифікації біосигналів за допомогою мови програмування C# у середовищі Microsoft Visual Studio.

5. Розроблено елемент керування для відображення графічної інформації.

Список літератури

1. Hough P. Method and means for recognizing complex patterns / P. Hough // US Patent N 3069654, Dec. 18, 1962.

2. Duda R. Use of the Hough Transformation to detekt lines and curves in pictures / R. Duda, P. Hart // Communs. ACM. – 1972. – Vol. 15, N 1. – P. 11-15.

3. Вайнштейн Г.Г. Идентификация графических объектов на основе преобразования к пространству параметров / Г.Г. Вайнштейн, Е.А. Москвина, Д.А. Белов // Машинная диагностика и информационный поиск в медицине. – М.: Наука, 1986. – С. 98-108.

4. Поворознюк А.И. Преобразование пространства признаков при идентификации структурных элементов квазипериодических сигналов / А.И. Поворознюк, А.Е. Филатова // Компьютерное моделирование: сб. научн. тр. – Белгород: БелГТАМС, 1998. – С. 33-39.

5. Поворознюк А.И. Оптимизация пространства признаков при структурной идентификации квазипериодических сигналов / А.И. Поворознюк, А.Е. Филатова // Информационные системы: сб. научн. тр. – Х.: НАНУ, ПАНИ, ХВУ, 1998. – Вып. 1 (9). – С. 112-115.

6. Поворознюк А.И. Выбор метрики пространства признаков в задаче структурной идентификации квазипериодических сигналов / А.И. Поворознюк, А.Е. Филатова // Системний аналіз, управління і інформаційні технології: Вісник Харківського державного політехнічного університету. – Х.: ХДПУ, 2000. – Вип. 99. – С. 138-141.

7. Поворознюк А.И. Применение преобразования Хока для структурной идентификации физиологических сигналов / А.И. Поворознюк // Моделювання та інформаційні технології. Збірник наукових праць інституту проблем моделювання в енергетиці. – К.: ІПМЕ, 2003. – Вип. 22. – С. 143-149.

8. Поворознюк А.И. Оптимизация метода идентификации биосигналов на основе преобразования Хока / А.И. Поворознюк, Ю.В. Миргород // Проблемы информатики и моделирования: Вестник национального технического университета «ХПИ». – Х., 2009. – Вып. 13. – С. 128-132.

9. Миргород Ю.В. Исследование параметров преобразования Хока при структурной идентификации квазипериодических биосигналов / Ю.В. Миргород, А.И. Поворознюк // Прикладная радиоэлектроника. – Х., 2007. – Т. 6, № 1. – С. 62-66.

10. Hong Zhu Software Design Methodology //AAAI Press, British Library of Congress Cataloguing in Publication Data, 2005. – 340 p.

11. Мак-Домальд М. Microsoft ASP.NET 3.5 с примерами на C# и Silverlight 2 для профессионалов: пер. с англ. / Метью Мак-Домальд, Марио Шпунта. – 3-е изд. – М.: ООО «И.Д. Вильямс», 2009. – 1408 с.

12. Джеффри Р. CLR via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке C# / Рихтер Джеффри. – Питер, 2007. – 656 с.

13. Weilkiens T. System engineering with SysML/UML: modeling, analysis, design / Tim Weilkiens // The OMG Press. ISBN 978-0-12-374274-2. – 2007. – 307 p.

Надійшла до редколегії 16.12.2010

Рецензент: д-р техн. наук, проф. В.Д. Дмитрієнко, Національний технічний університет «ХПІ», Харків.

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПОДСИСТЕМЫ ИДЕНТИФИКАЦИИ БИОСИГНАЛОВ НА ОСНОВЕ ПРЕОБРАЗОВАНИЯ ХОКА

Ю.В. Мирогород

В статье рассмотрено построение программной модели подсистемы идентификации биосигналов. Приведен выбор инструментальных средств и методологии построения программного обеспечения.

Ключевые слова: *физиологические квазипериодические сигналы (ФКС), преобразование Хока, пространство параметров, диаграмма классов, структура программного обеспечения.*

SOFTWARE DESIGN FOR BIOSIGNALS PROCESSING BLOCK BASED ON HOUGH TRANSFORMATION

Y.V. Myrgorod

The article has observation of program model design for biosignals processing block based on Hough transformation. Software design tools and software design methodology are given.

Keywords: *physiological quasi-periodic signals, Hough transformation, parameter space, class diagram, software structure.*