УДК 004.94

ТЕХНОЛОГІЯ СТВОРЕННЯ ПРОГРАМНОГО КОМПЛЕКСУ ДЛЯ МОДЕЛЮВАННЯ ФІЗІОЛОГІЧНИХ СИСТЕМ

П.М. Ліссов

Інститут програмних систем НАН України,
03182, проспект Академіка Глушкова, 40.
Тел.: (8044)526 5169, pavel@lissov.kiev.ua

Розглянуті проблеми моделювання організму людини. Наводиться опис підходів до створення комплексів моделювання. Запропоновано оригінальний підхід до моделювання взаємодії функціональних систем організму на основі технології COM з забезпеченням гнучкого під'єднання моделей. Наведено опис створеного за даною технологією комплексу та результатів імітаційного моделювання.

Problems of human body modeling are given in the article. Approaches to the creation of complexes for modeling are described. The original approach to modeling of interconnected functional systems of human, based on COM-technology, is offered. It allows flexible connection of models to the complex. The complex, created following given technology, is described and the results of modeling are discussed.

Вступ

Для поглибленого розуміння основних кількісних закономірностей фізіології людини вже більше 50 років створюються математичні моделі. Для імітаційних досліджень фізіології на сучасному рівні необхідно розробляти комплексні моделі, які описують взаємозв'язок різних систем. Ці моделі мають відобразити закономірності функціонування на субклітинному, клітинному, популяційному та системному рівні організації [1, 2]. Таким чином виникає задача створення спеціалізованих програмних засобів, які забезпечують весь цикл проведення комп'ютерних імітаційних досліджень, шляхом компоновки моделей, налаштування їх та аналізу результатів імітаційних експериментів.

Створити цілісну модель організму людини неможливо. Необхідно розробити моделі окремих функціональних систем та алгоритми їх взаємодії. Існує багато підходів до створення таких моделей і поєднання їх у модель цілого організму. Їх можна розділити на дві групи – жорсткі і гнучкі.

У межах жорсткого підходу список можливих ситуацій наперед заданий. Створюється єдина програма, у якій на рівні коду назавжди прописуються моделі та взаємозв'язки між ними. Програма може вільно оперувати групами даних. Прикладами жорсткого підходу є [3 – 5]. Жорсткий підхід дозволяє порівняно швидко створити модель з допомогою сучасних мов програмування із забезпеченням необхідних засобів налаштування та візуалізації без необхідності створення загальних механізмів та технологій. До недоліків жорсткого підходу належить те, що зв'язки між моделями у такому випадку прописані усередині програми і доступні для користувача у найкращому випадку для перегляду, при чому як правило у вигляді документації. Модифікація таких зв'язків користувачем взагалі не передбачена, або реалізована для набору заздалегідь визначених розробником варіантів.

Ще одним суттєвим недоліком «жорсткого» підходу є те, що моделі та весь організм можуть створювати лише розробники комплексу. Це призводить до суттєвих обмежень у моделюванні. Один колектив розробників фізично може створити обмежену за складністю систему моделей. При цьому доводиться реалізовувати моделі, створені і адекватно налаштовані іншими, заново, лише за опублікованими математичними формулами, які часто містять неточності та помилки [1]. Саме це, напевно, є причиною того, що всі існуючі на сьогодні «жорсткі» моделі описують тільки обмежену частину організму людини або лише на поверхневому рівні.

Гнучкий підхід передбачає створення і стандартів опису моделей, і програм для оперування з моделями. Завдяки такому підходу різні колективи розробників можуть створювати свої моделі, проводити експерименти за якими можуть всі користувачі програми. Прикладами таких комплексів моделювання є комплекси JSim [6], BioUML [7] та Simulink [8]. Такі комплекси дають змогу дослідникам у сфері фізіології сконцентруватися на створенні та описі моделі не витрачаючи час на створення інструментів для моделювання. Але гнучкий підхід також має недоліки. Як правило, мова для опису моделей оптимізована для конкретних задач і при виході за межі таких задач розробнику доводиться використовувати досить складні прийоми для отримання бажаного результату (як правило за допомогою використання вбудованих скриптових мов програмування), а опис моделі в такому разі виходить заплутаний і складний для розуміння. Можливості представлення результатів у таких комплексах сильно обмежені. Хоча такі системи інколи мають великий набір інструментів візуалізації результатів, але неможливо передбачити всі вимоги, які можуть виникнути, і розробнику доводиться обмежуватись стандартними засобами. Ще одним значним недоліком є відсутність інструментарію для налаштування моделі. Єдиною можливістю для налаштування є встановлення значень параметрів моделі.

Гнучкі комплекси, вищенаведені, не надають можливості оперувати структурами змінних та масивами. Цей недолік проявився при моделюванні серцево-судинної системи з великою кількістю компартментів, такої як модель з 156 компартментами [3].

Мета даної роботи – створення спеціалізованого комплексу для моделювання взаємодіючих фізіологічних систем людини (СКМ).

Для досягнення даної мети необхідно розв'язати такі завдання:

- розробити формальні вимоги для опису моделей;
- розробити програму для поєднання моделей у системи;
- забезпечити засоби для аналізу результатів моделювання.

Вимоги до СПК

Комплекс має дозволяти створювати модель віртуального організму як поєднання окремих модулів. Кожен з таких модулів має містити в певній мірі формалізований опис моделі. При цьому необхідно забезпечити можливість вільного використання всіх можливостей ЕОМ для опису моделі та для створення засобів її налаштування та візуалізації результатів.

Всі модулі мають працювати під керуванням певної програми для моделювання, яка забезпечить доступ до функцій настройки моделей та до засобів відображення результатів.

Необхідно створити ряд формальних вимог до моделей, які забезпечать роботу модулів під керуванням згаданої програми та забезпечать можливість доступу до вбудованих засобів налаштування і представлення результатів. При цьому такі формальні вимоги мають представляти максимально широкі можливості для використання нестандартних, не передбачених заздалегідь засобів. Також бажано забезпечити можливість створення модулів на різних сучасних мовах програмування з використанням максимально можливої кількості сучасних технологій.

Необхідно також забезпечити можливість розвитку та доробки існуючої програми керуванням та моделей із забезпеченням сумісності з попередніми версіями.

Створення моделей на основі технології COM

У подальшому під терміном *модель* будемо розуміти математичну модель окремої функціональної системи. Набір таких моделей, який описує роботу взаємопов'язаних функціональних систем, будемо називати *віртуальним організмом*.

Для забезпечення можливості поєднання моделей обрана гнучка схема організації комплексу. Процесом обчислень управляє керуюча програма, до якої моделі приєднуються у вигляді додаткових модулів (plug-in). Моделі можуть бути створені окремо і приєднуватись до віртуального організму користувачем під час роботи СПК.

Модуль – це певна програмна одиниця, яка містить у собі опис моделі певного об'єкту (фізіологічної системи, органу, механізму, зовнішнього середовища і т.д.) а також довільні програмні компоненти для налаштування моделі, відображення результатів тощо.

Найкращим, на нашу думку, є виконання модулів у вигляді окремих бібліотек, які реалізовують певний програмний інтерфейс і поєднуються за допомогою технології COM (Component Object Model) [9].

Взаємодія COM-об'єктів заснована на виклику методів об'єктів. Тому для забезпечення формальних вимог до модулю необхідно визначити інтерфейс з описом потрібних методів і додати до COM-об'єктів підтримку даного інтерфейсу. Тоді управляюча програма й інші модулі зможуть працювати з такими модулями через даний інтерфейс. При цьому залишається підтримка старих інтерфейсів і вже створені модулі продовжать працювати коректно у рамках старої функціональності.

COM-об'єкт може підтримувати необмежену кількість інтерфейсів, тому можна розробити один модуль який, підтримуючи декілька інтерфейсів, зможе бути використаний для вирішення різних задач, а при необхідності забезпечити додаткову функціональність достатньо лише створити необхідний інтерфейс без необхідності принципових змін, переробки і переналаштування моделей.

Важливою перевагою COM-об'єктів є те, що вони можуть бути створені на різних мовах програмування. Більшість сучасних мов програмування підтримують технологію створення COM-об'єктів. Це дає розробнику можливість обрати саме ті засоби для створення модулю, які йому найбільш зручні, а також дозволяє використати існуючі розробки. Крім того, технологія COM підтримується стандартними сервісами Windows, а це забезпечує можливість запуску СПК на довільному комп'ютері з Windows без необхідності встановлення додаткових компонентів.

Кожна модель представляється як «чорний ящик», який має визначений набір входів, виходів та параметрів. Під входами ми розуміємо змінні, значення яких треба передати моделі перед початком кожного циклу обчислень. Параметри – значення, які не змінюються протягом одного експерименту. Для параметра як правило вимагається наявність значення, водночас як для входу є можливість не встановити значення (треба бути дуже обережним із створенням входів на які обов'язково повинні бути задані значення так як вони обмежують можливості індивідуального використання моделі).

У загальному випадку розрахунок експерименту поділяється на послідовність кроків. На кожному кроці модель отримує значення вхідних змінних та розраховує значення виходів. СПК не накладає такого обмеження

на моделі. Однак модель не обов'язково повинна проводити циклічні обчислення. Кожна модель запускається в окремому потоці та може використовувати будь-який алгоритм обчислень. Наприклад, при створенні моделі навантаження, яке задається до початку експерименту, циклічні обчислення не потрібні. Значення виходу відоме для довільного моменту часу і тому модуль ігнорує значення кроку обчислень і одразу готова видавати значення виходів.

Обмеження, які накладаються на модель, такі:

- кожен модуль має методи для встановлення значень параметрів та настройки входів. Під настройкою входів ми розуміємо можливість пов'язати із входом моделі константу або вихід іншої моделі (в такому випадку на вхід встановлюється інтерфейс моделі і ідентифікатор виходу);
- всі параметри, входи та виходи моделі пронумеровані і можуть (необов'язково) мати строкові ідентифікатори, які спрощують визначення їх номерів;
- всі параметри, входи та виходи мають числовий тип (double). Таке обмеження пов'язане з тим, що необхідно визначити єдиний тип даних для передачі даних між моделями. При цьому інтерпретацію таких чисел визначає розробник модуля – це можуть бути як власне дробові числові дані, так і цілі числа або дані інших типів, таких як логічного типу, представлені у вигляді певних числових констант. У таких випадках важливо розуміти, що комплекс не надає засобів для контролю за коректністю вводу таких даних. У випадку використання таких параметрів розробнику модуля варто розробити інтерфейсну частину, яка дозволить задати значення таких параметрів з урахуванням їх логічного значення;
- кожна модель має метод, що повертає розрахований час та значення заданого виходу у заданий момент часу (менший, ніж розрахований час). При цьому бажано, щоб модель повертала значення змінної для будь-якого моменту часу до вказаного розрахованого моменту;
- у разі, коли на вхід моделі поставлено вихід іншої моделі, дана модель сама відповідає за отримання їх значень від іншої моделі через переданий їй інтерфейс;
- кожна модель має методи керування, які включають запуск обчислень, паузу та зупинку експерименту.

Такі обмеження реалізуються за допомогою визначення інтерфейсу IModel. Кожен модуль має містити клас, який реалізовує цей інтерфейс. Організація обчислень усередині модуля, а також наявність довільних додаткових засобів визначається розробником модуля.

У найпростішому випадку архітектура комплексу показана на Рис. 1.

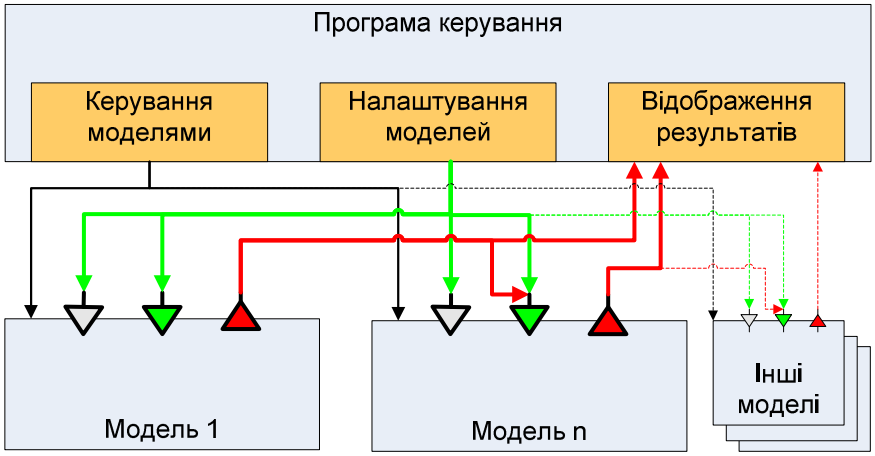

Рис. 1. Архітектура базової частини комплексу

До програми керування можна приєднати довільну кількість модулів. Чорні стрілки показують що команди ідуть від програми керування до всіх модулів. Зелені стрілки демонструють що при налаштуванні модулів програма керування встановлює значення параметрів та входів до початку експерименту. Червоними стрілками показано як результати обчислень під час та після обчислень від модуля може отримувати як програма керування так і інші моделі. При цьому ще раз зазначимо що моделі та керуюча програма мають самі забирати потрібні результати у інших модулів виконуючи запити до цих модулів.

Для забезпечення роботи без взаємоблокувань необхідно ввести обмеження: кожна модель має право вимагати дані від інших моделей лише для моменту часу, який вона вже розрахувала (і, відповідно, може віддати іншим моделям). Відповідність модулів даному обмеженню не перевіряється комплексом, і лежить «на совісті» розробників моделі. Порушення даного обмеження допустиме лише в особливих випадках і буде суттєвим недоліком модуля, так як використання такого модуля при створенні віртуального організму може призвести до взаємоблокування. А враховуючи існуючі підходи до опису та моделювання фізичних та фізіологічних (природних) процесів, використання для розрахунку поточного моменту даних «з майбутнього» не потрібне.

Для забезпечення зручного налаштування моделей модуль може мати спеціальну екранну форму. У другій версії інтерфейсу IModel передбачено метод, при виклику якого відкривається така екранна форма. Потім з неї можна отримати доступ до інших, закладених розробником форм для налаштування моделей. Передбачено

окремий метод, через який керуюча програма передає модулю весь набір модулів, підключених на даний момент. Таким чином стає можливим створення взаємопов'язаних моделей, які зможуть більш тісно взаємодіяти між собою таким чином, як це передбачено їх розробником.

У моїй реалізації модуль не викликає методи керуючої програми, а методи інших модулів викликає лише під час обчислень для отримання значень входів. Отже модуль працює «пасивно», виконуючи певні дії лише у відповідь на запити управляючої програми, виконані у вигляді виклику методів даного модулю. Втім це не є обмеженням комплексу і розробник може визначити додатковий інтерфейс і змінити схему взаємодії модулів і комплексу, що дозволить йому зробити модулі «активними».

Завдяки такому підходу процес створення віртуального організму можна розділити на такі етапи:

- кваліфіковані програмісти за наданими їм описами математичних моделей створюють окремі програмні модулі;
- програмісти разом з експертами у предметній області налаштовують моделі;
- експерт-фізіолог з окремих модулів збирає віртуальний організм та проводить його додаткове налаштування під свої потреби;
- користувач (дослідник, лікар, викладач, студент) використовує віртуальний організм для вирішення своїх задач – навчальних, дослідницьких або прикладних.

Відповідно до поставлених задач та інших особливостей використання комплексу, окремі етапи можуть проводитись одними і тими ж людьми та взагалі об'єднуватись.

Запропонований підхід дозволить на першому та частково другому етапах використовувати всі можливості сучасних мов програмування, вже на другому етапі мати зручні інструменти для налаштування моделі, а на третьому і четвертому етапах використати всі переваги можливості гнучкого поєднання моделей, в тому числі використати моделі, розроблені іншими колективами.

Приклад застосування

Першим віртуальним організмом, для створення якого був використаний даний комплекс, став організм, який складається з серцево-судинної системи, системи барорецепторної регуляції та нирок. Взаємозв'язок моделей системи показано на Рис. 2. Опис математичних моделей комплексу подано у [3, 4].

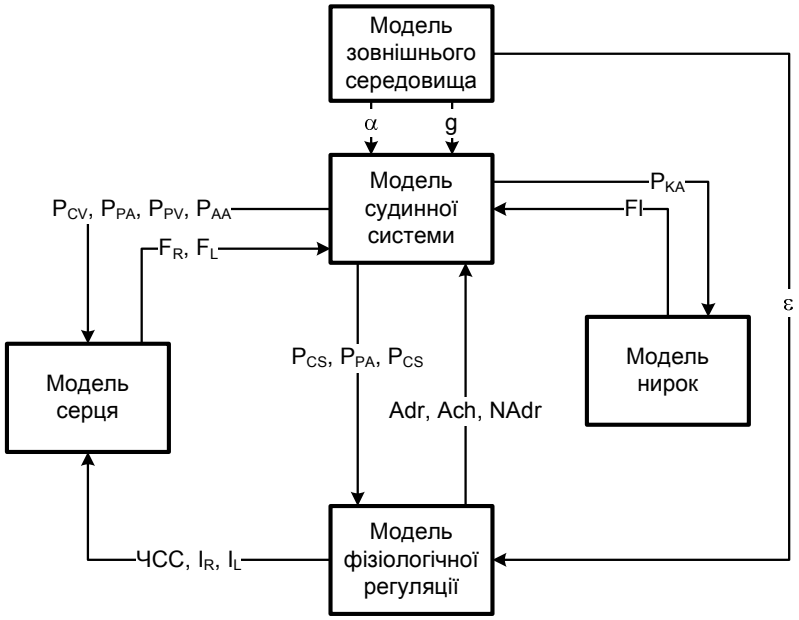

Рис. 2. Схема взаємозв'язку моделей організму

Модель зовнішнього середовища представлена декількома окремими моделями навантажень – для куту тіла, гравітаційного перевантаження та атмосферного тиску.

Гнучка організація віртуального організму дозволила нам описати два варіанти моделей серця – для детальних та тривалих експериментів. Перша описує роботу серця на малих інтервалах [3] і дозволяє імітувати гемодинаміку в межах серцевого циклу, але при сучасному рівні техніки дозволяє розрахувати експеримент тривалістю лише у декілька хвилин. Друга модель описує діяльність серця за залежністю середнього потоку через серце від середніх значень тиску, отриманій за допомогою першої моделі, і таким чином дозволяє імітувати лише середні значення але на тривалих проміжках, таких як декілька годин або навіть днів. При цьому вибір тієї чи іншої моделі серця можна зробити безпосередньо перед початком експерименту лише вказавши залежність входів та виходів, без необхідності жодних змін у налаштуваннях моделей.

Оформлення модулів у вигляді програмних бібліотек дає можливість використовувати максимально зручні засоби для представлення математичних моделей та організації налаштування. Так, для опису судинної системи

використовуються масиви, що дозволяє представити систему як набір однакових об'єктів, які відрізняються лише значеннями параметрів і описуються за допомогою одного набору рівнянь. Навантаження на систему (модель зовнішнього середовища) користувач задає за допомогою графічного «малювання» динаміки навантаження (Рис. 3), що часто є зручнішим, ніж зазначення формул.

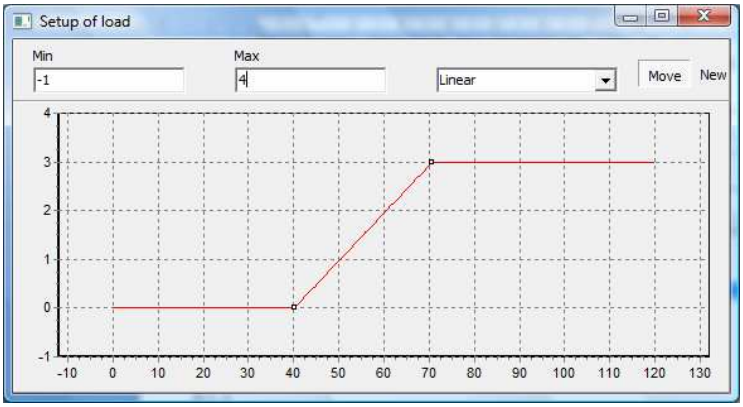

Рис. 3. Графічне представлення навантаження

Для початкового приблизного налаштування розроблені спеціальні підпрограми, які дозволяють розрахувати «ідеальні» значення параметрів для досягнення бажаних значень виходів. У нашому випадку для всієї серцево-судинної системи практично неможливо виміряти всі параметри, такі як опори ділянок та значення ненапруженого об'єму. За основу для налаштування ми взяли нормальні значення для тиску, об'єму і потоку в певних контрольних точках [10]. Враховуючи відомі фізіологічні характеристики та логічні залежності ці значення були виставлені для інших ділянок. Нам були відомі не параметри моделі, а «ідеальні» значення виходів. Далі за допомогою спеціально створеної підпрограми був проведений розрахунок параметрів – гідродинамічного опору, ненапруженого об'єму, констант апроксимації залежності тиску від об'єму. Це дозволило досить швидко налаштувати модель на отримання адекватних результатів.

Для забезпечення адекватної поведінки системи у динаміці також створені спеціальні підпрограми, такі як підпрограма корекції залежності тиску від об'єму. Однак в даному випадку автоматично ми можемо отримати лише приблизні значення параметру, для забезпечення адекватних результатів необхідне ручне доналаштування. Для ручного налаштування створено інструменти для внесення абсолютних і відносних змін у обрані набори параметрів. При цьому створена система наборів параметрів, згрупованих за різними критеріями, оперуючи якими можна досить швидко адекватно змінити поведінку системи; користувач також може сам створити необхідну групу за допомогою гнучких фільтрів. Таким чином користувач може окремо змінювати параметри, наприклад, для всіх артеріальних ділянок, або для всіх ділянок легень, так як показано на Рис. 4.

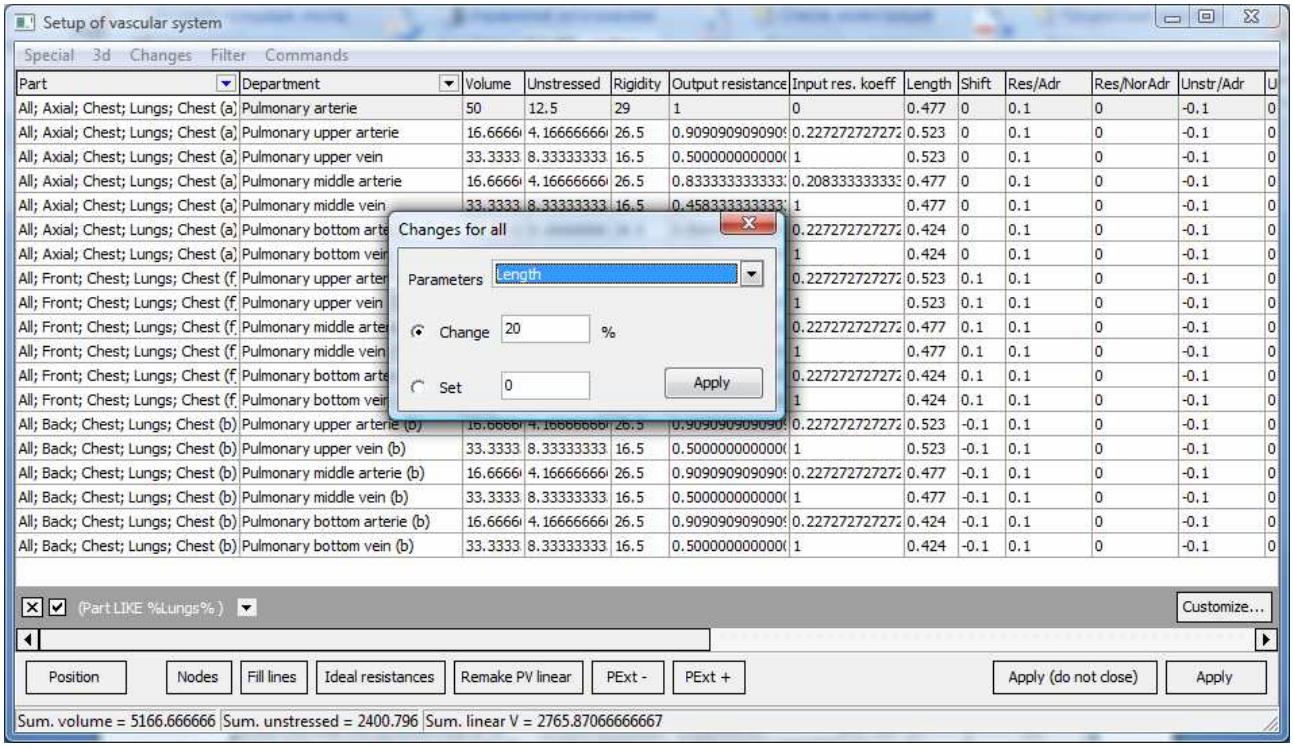

Рис. 4. Вікно налаштування моделі судинної системи

Результати моделювання

Для демонстрації можливостей СПК та адекватності моделей проводиться експеримент, у якому на віртуальний організм діє ортостатичне навантаження. Використовується модуль судинної системи з 155 компартментів, модуль насосної діяльності пульсуючого серця, модуль барорефлекторної регуляції серцево-судинної системи та модуль зовнішнього навантаження. Модулі поєднані у відповідності до Рис. 2.

Імітується експеримент, у якому тіло людини спочатку знаходиться у лежачому положенні, після 30-ї секунди за 10 с людина встає (кут тіла до горизонту лінійно зростає до $\pi/2$ і залишається на такому рівні до кінця експерименту. Експозиція 60 с, крок обчислень 0.001 с. На організм діє нормальне гравітаційне навантаження 1g.

Вікно відображення результатів обчислень показано на рис. 5.

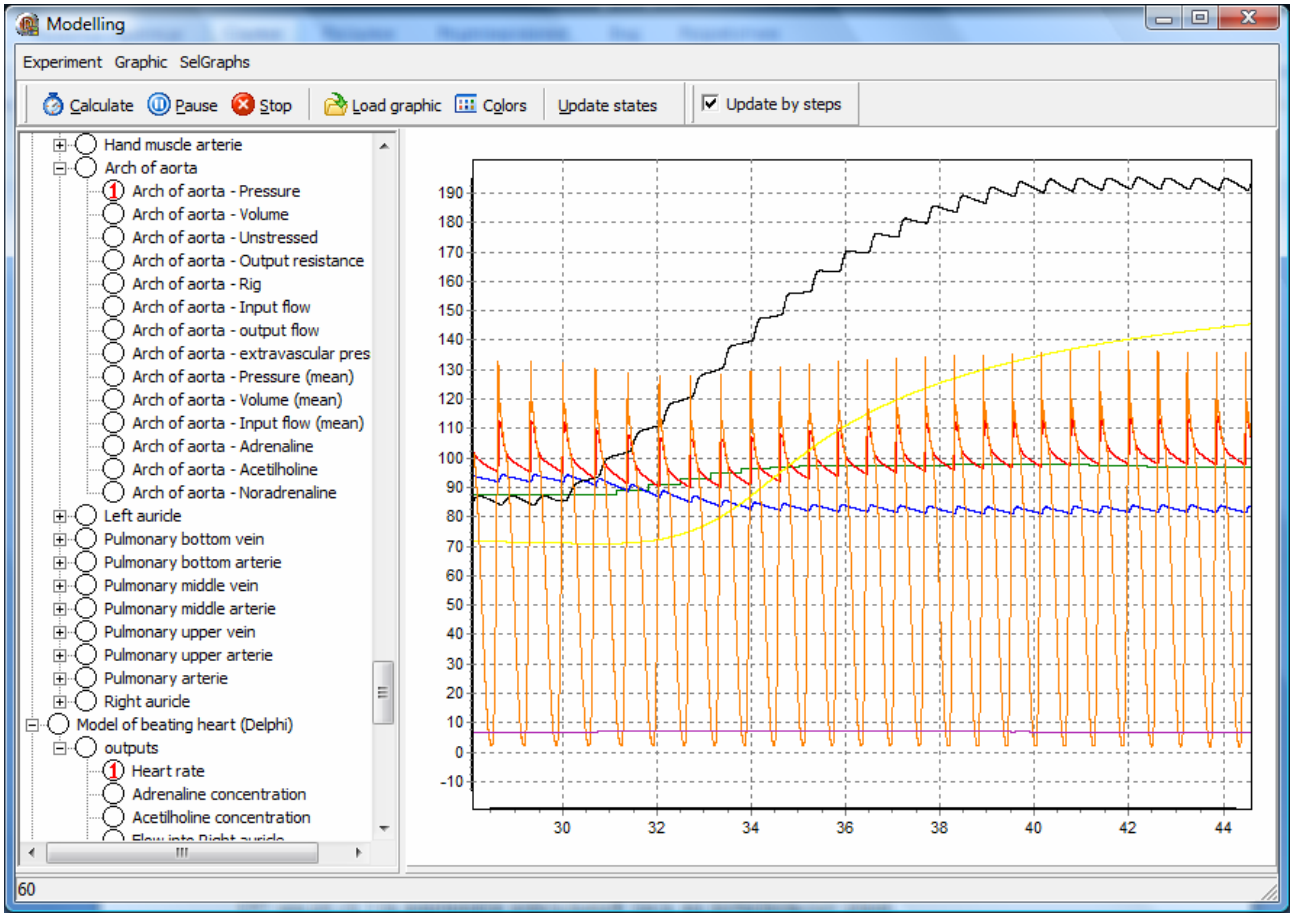

Рис. 5. Результати експерименту

На графіку показано частоту серцевих скорочень (ЧСС) зеленим, тиск у дузі аорти (червоний), тиск у каротидній зоні (синій), тиск у артерії стопи (чорний), жорсткість артерії стопи (жовтий), об'єм артерії стопи та об'єм лівого шлуночку (оранжевий).

Видно, що у відповідь на навантаження знижується тиск у каротидній зоні та аорті (модель судинної системи). Внаслідок цього змінюється концентрація гормонів (модель регуляції), що призводить до росту ЧСС та росту жорсткості судин. Це дозволяє втримати на стабільному рівні об'єм судин нижньої частини тіла (у нашому випадку об'єм артерії стопи повертається до значення, близького до початкового, за 10 секунд, при рості тиску від 85 до 195 мм.рт.ст.) та вирівняти аортальний тиск на рівні, вищому за початковий.

Таким чином, основні показники гемодинаміки, отримані за допомогою комплексу, відповідають нормальним значенням для гемодинаміки людини при заданому навантаженні.

Вікно для відображення результатів відображує виходи моделей у згрупованому вигляді. Групи формуються за модулями, усередині кожного модулю групування задає сам модуль. Це дозволяє організувати виходи в структуру, в якій можна швидко знайти необхідний вихід для відображення. Таке групування є необхідним якщо врахувати що модель судинної системи налічує близько 2200 виходів.

Вікно для відображення результатів може бути доповнене додатковими функціями по обробці результатів. Перевагою запропонованого підходу є те, що такі функції можна додавати без втручання у існуючі модулі. У нашому випадку розроблені засоби збереження на диск результатів експерименту та їх завантаження і відображення. Така можливість реалізована за допомогою окремого модуля, який відповідає наведеним вимогам до модулів комплексу і приєднується до комплексу стандартними засобами. Це дає можливість при вже створенні віртуального організму забезпечити збереження результатів, навіть без взаємодії з керуючою програмою.

Висновки

Запропонований комплекс дозволяє розв'язати завдання, поставлені перед ним. Ми отримали можливість окремо розробити модулі з математичними моделями для опису окремих функціональних систем організму різного рівня, налаштувати їх, та створювати систему модулів вільно вказуючи взаємозв'язки входів та виходів. Таким чином ми отримуємо комплекс, який дозволяє створювати системи з набору моделей, вибираючи таку їх конфігурацію, яка максимально відповідає завданням конкретного експерименту.

Результати модельних експериментів на даному комплексі доводять зручність використання комплексу для налаштування моделей та перегляду результатів.

1. *Hunter P., Robbins P., Noble D.* The IUPS human physiome project // *Eur. J. Physiol.* – №445(1). – 2002. – P.1–9.
2. *Rothe C.F., Gersting J.M.,* Cardiovascular interactions: an interactive tutorial and mathematical model // Am. Phys. Society. – 2002. – Advances of physiological education, № 2 – P. 98–109.
3. *Григорян Р.Д., Лиссов П.Н.* Программный имитатор сердечно-сосудистой системы человека на основе ее математической модели // Проблеми програмування. – 2004. – № 4. – С. 100–111.
4. *Григорян Р.Д., Атоев К.Л., Лиссов П.Н., Томин А.А.* Программно-моделирующий комплекс для теоретических исследований взаимодействия физиологических систем человека // Проблеми програмування. – 2006. – №1. – С. 79–92.
5. *Quantitative* Circulatory Physiology. User's guide. http://www.biosim.com/usersguide/usersguide.html
6. http://nsr.bioeng.washington.edu
7. *Kolpakov F.A.*, BioUML – framework for visual modeling, 2002.
8. *Simulink* Release Notes, http://www.mathworks.com/access/helpdesk/help/pdf_doc/simulink/rn.pdf
9. http://www.microsoft.com/com/default.mspx
10. *Справочник* по космической биологии и медицине / Под ред. А.И. Бурназяна, О.Г. Газенко. – М.: Медицина – 1983. – 352 с.