

$$\left\{ \begin{array}{l} \Delta = K \left(\sum_{i=1}^K k_i^2 \sum_{i=1}^K k_i^4 - \left(\sum_{i=1}^K k_i^3 \right)^2 \right) + 2 \sum_{i=1}^K k_i \sum_{i=1}^K k_i^2 \sum_{i=1}^K k_i^3 - \left(\sum_{i=1}^K k_i^2 \right)^2 - \left(\sum_{i=1}^K k_i \right)^2 \sum_{i=1}^K k_i^4, \\ \Delta_0 = \sum_{i=1}^K k_i^2 \left(d_0 \sum_{i=1}^K k_i^4 + d_1 \sum_{i=1}^K k_i^3 - \sum_{i=1}^K k_i^2 d_2 \right) - \left(\sum_{i=1}^K k_i^3 \right)^2 d_0 + \sum_{i=1}^K k_i \left(d_2 \sum_{i=1}^K k_i^3 - \sum_{i=1}^K k_i^4 d_1 \right), \\ \Delta_1 = k \left(d_1 \sum_{i=1}^K k_i^4 - d_2 \sum_{i=1}^K k_i^3 \right) + \sum_{i=1}^K k_i \left(d_2 \sum_{i=1}^K k_i^2 - \sum_{i=1}^K k_i^4 d_0 \right) + \sum_{i=1}^K k_i^2 \left(d_0 \sum_{i=1}^K k_i^3 - \sum_{i=1}^K k_i^2 d_1 \right), \\ \Delta_2 = k \left(d_2 \sum_{i=1}^K k_i^2 - d_1 \sum_{i=1}^K k_i^3 \right) + \sum_{i=1}^K k_i \left(d_1 \sum_{i=1}^K k_i^2 + \sum_{i=1}^K k_i^3 d_0 - \sum_{i=1}^K k_i d_2 \right) - \left(\sum_{i=1}^K k_i^2 \right)^2 d_0, \end{array} \right. \quad (9)$$

а

$$\left\{ \begin{array}{l} d_0 = \sum_{i=1}^K \ln z_k, \\ d_1 = \sum_{i=1}^K k_i \ln z_k, \\ d_2 = \sum_{i=1}^K k_i^2 \ln z_k. \end{array} \right. \quad (10)$$

В результаті маємо оцінки

$$\hat{\sigma} = \sqrt{\frac{1}{2B}}, \quad \hat{\mu} = \frac{C}{2B} \quad (11)$$

Висновок

Експерименти на модельних та реальних даних показали, що запропонований метод дозволяє апроксимувати ЕКГ-сигнал з похибкою 1% навіть при високому рівні адитивного шуму, який в експериментах досягав 50%. Така точність визначення невідомих параметрів ЕКГ є прийнятною для правильної інтерпретації реальних сигналів при ЕКГ-діагностиці.

Список використаних джерел

1. Файнзильберг Л.С. Компьютерная диагностика по фазовому портрету электрокардиограммы. – Киев: Освіта України, 2013. – 191 с.

УДК 681.3.06

МОДЕЛЮЮЧИЙ АЛГОРИТМ ДЛЯ ОЦІНКИ НАДІЙНОСТІ СКЛАДНИХ БАГАТОФУНКЦІОНАЛЬНИХ ТЕХНІЧНИХ СИСТЕМ

Мельник А.М.¹⁾, Проць С.Я.²⁾

Тернопільський національний економічний університет

¹⁾ к.т.н., доцент; ²⁾ магістрант

I. Постановка проблеми

Подальше вдосконалення показників надійності сучасних складних систем продовжує залишатися актуальним завданням. Існуючі методи та алгоритми не задовольняють вимогам швидкодії і точності оптимізації надійності на стадії проектування для технічних систем середньої та великої розмірності; більшість розроблених в останні роки методів і алгоритмів є евристичними (або метаевристичними) і не завжди гарантують знаходження точного оптимального рішення [5].

Таким чином, для оцінки надійності складних багатофункціональних систем рекомендується застосовувати імовірнісне моделювання.

II. Мета роботи

Метою наукового дослідження є підвищення якості проектування структурно – складних технічних систем за рахунок розробки і застосування на практиці методів, алгоритмів та методик вирішення завдань оптимізації проектної надійності.

При побудові моделі (розробка моделюючого алгоритму) складний стохастичний процес розглядається як послідовність скінченного числа взаємозалежних елементарних стохастичних актів.

III. Розробка моделюючого алгоритму

Моделюючий алгоритм представляє собою систему правил, що визначають послідовність операцій (арифметичних, логічних і деяких допоміжних), які необхідно виконати над вихідними даними, щоб одержати чисельні значення величин, обраних як шукані характеристики й показників надійності [7].

Складні технічні системи, які можна досліджувати за допомогою моделюючих алгоритмів, повинні задовольняти деяким вимогам, основними з яких є наступні: можливість представлення структури системи мовою графів (надійнісно-функціональних схем (НФС)), кожний компонент системи має тільки два можливих стани: працездатний або непрацездатний, однозначне визначення в будь-який момент часу стану системи через стани її компонентів [6].

Перелік цих обмежень, що накладають на клас розглянутих систем, істотно відображається на принципах моделювання процесів їхнього функціонування. Такі системи представляють собою багатокомпонентні й багатофункціональні системи (n компонентів і m функцій). Дослідження їх надійності проводять роздільно по кожній з функцій, які реалізуються системою (надійність по функціях).

Розв'язування задачі пов'язане з одержанням на комп'ютері безлічі реалізацій процесу виконання кожної з функцій системи. При цьому кожна реалізація $z_i(t)$, $i = 1, 2, \dots$ має два можливих рівні: коли в момент t дана функція виконується системою і другий якщо функція не виконується.

В основі побудови необхідних реалізацій (роздільно по функціях) лежить моделювання процесів функціонування (тобто послідовного чергування працездатних і непрацездатних станів) всіх n компонентів системи. Це пов'язане з тим, що за допомогою комп'ютера відтворюються, по суті, відрізки часу безвідмовної роботи й відновлення кожного компонента системи (а не окремих функцій), і потім на цій основі формуються реалізації безпосередньо процесів, що цікавлять нас.

Процес функціонування системи можна розглядати як послідовну зміну її станів. Інтервали часу, які, підсумовуючись, утворюють в послідовність моментів зміни стану i -го компонента, представляють собою по черзі реалізації випадкових величин (відповідно час безвідмовної роботи й час відновлення). Розподіли цих величин задаються у вхідних даних для розв'язання задачі.

Дослідження процесу функціонування системи за допомогою комп'ютера проводиться послідовно від моменту $t_a = 0$ до $t_0 = T_{\text{мод}}$, причому таке дослідження може бути організоване по різному.

Моделювання на комп'ютері процесів функціонування різних систем пов'язане з виробленням великої кількості випадкових чисел із заданими законами розподілу. Для цієї мети використається звичайно один з наступних способів [3]:

- генерування випадкових чисел спеціальною електронною приставкою до машини – датчиком випадкових чисел;
- одержання випадкових чисел у машині відповідно до заданої програми формування.

Однак, часто застосовується програмний спосіб формування випадкових послідовностей, який базується на використанні деякого рекурентного співвідношення. При цьому як основний механізм генерування випадкових величин використовує послідовність рівномірно розподілених в інтервалі $(0,1)$ випадкових чисел, які піддаються подальшим перетворенням для одержання заданих законів розподілу.

Розглянемо деякі методи формування випадкових величин з різними законами розподілу. i – випадкове число з рівномірним законом розподілу в інтервалі $(0,1)$. Тоді для одержання i із сукупності випадкових чисел, що мають задану функцію розподілу $F(x)$, необхідно вирішити i наступне інтегральне рівняння:

$$\int_{-\infty}^{\infty} f(x) dx = \xi_i \quad (1)$$

i при якому функція розподілу $F(x)$ дорівнює ξ_i . Для деяких законів розподілу рівняння (1) вдається розв'язати безпосередньо. У більшості ж практично важливих випадків рівняння 1) точно не вирішується. Тоді використовують наближені способи перетворення випадкових чисел.

Моделюючий алгоритм для оцінки надійності складних багатофункціональних технічних систем складається із чотирьох підалгоритмів і логічного оператора, що перевіряє умову закінчення моделювання. Реалізація моделі на комп'ютері представляє собою послідовне теоретичне відтворення процесу, що моделює реальну фізичну систему.

Висновок

Особливістю запропонованого методу є те, що одержувана в результаті моделювання інформація за своєю природою аналогічна тій інформації, яку можна було б отримати в процесі дослідження реальної системи, однак обсяг її значно більший і на її одержання затрачається менше коштів і часу. Тому метод імовірнісного моделювання більш ефективний у порівнянні з дослідженням реальної системи.

Список використаних джерел

1. А.М. Лоу .Имитационное моделирование / А.М. Лоу, В.Д. Кельтон. - 3-е изд. - СПб.: Питер, 2004. - 847 с.
2. Васілевський О.М. Нормування показників надійності технічних засобів: навч. посібник / О.М. Васілевський, В.О. Поджаренко. - Вінниця: ВНТУ, 2010, - 129 с.
3. В.М. Томашевський. Моделювання систем: підручник / М. Томашевський. - К.: ВНУ, 2005. - 352 с.
4. Шеннон Р. Имитационное моделирование систем - искусство и наука / Шеннон Р. - М.: Мир, 1978. - 418 с.
5. Щураков В. В. Надежность программного обеспечения систем обработки данных: Учебник / Щураков В. В. - М.: Финансы и статистика, 2007. - 272 с.
6. <http://forum1.au.nung.if.ua/forum/index.php?showtopic=505>
7. <http://links.dir.com.ua/linkinfo.php?linkID=16947>
8. <http://www.proelectro.info/info.php?i=3337>

УДК 004.424

ІМІТАЦІЙНА МОДЕЛЬ РОБОТИ ОКРЕМИХ КОМПОНЕНТ ЯДРА ОПЕРАЦІЙНОЇ СИСТЕМИ

Голець Є.Я.

*Тернопільський національний економічний університет,
магістрант*

І. Постановка проблеми

Складність алгоритмічних процесів, які реалізовані в ядрі операційної системи (ОС), значно ускладнює їх розуміння та подальше прийняття рішень щодо оптимізації використання комп'ютерних систем. Розробка програмного засобу для наочної імітації роботи окремих компонент ядра ОС є надзвичайно актуальною проблемою.

ІІ. Мета роботи

Одним із важливих компонент сучасної комп'ютерної системи є програмна складова, а саме операційна система [1].

Метою наукової роботи є розробка програмного засобу візуальної імітації, емуляції роботи ядра ОС, що забезпечує оптимальне використання ресурсів комп'ютерних систем.

ІІІ. Метод розподілу пам'яті з використанням дискового простору

Неефективне використання існуючих як програмних, так і апаратних ресурсів пов'язане із масштабним використанням засобів комп'ютерної техніки, що в свою чергу вимагає значних матеріальних затрат. Будь-яку ОС можна розглядати як набір певних складових, кожен з яких відповідає за конкретні функції. Саме архітектура ОС визначається таким набором компонентів і порядком їхньої взаємодії один з одним та із зовнішнім середовищем. Збільшення функціональності прикладного програмного забезпечення при обмежених ресурсах оперативної пам'яті поставило перед розробниками ОС задачу розміщення в пам'яті програм, розмір яких перевищує вільну пам'ять. Одним із шляхів вирішення цієї проблеми був поділ програм на частини — оверлеї (логічно завершені блоки програми). Всі оверлеї зберігалися на диску і переміщалися між пам'яттю та диском засобами операційної системи, однак поділ програми на частини і планування черговості їх завантаження в оперативну пам'ять здійснював програміст. Такий підхід суттєво зменшує надійність виконання програми та операційної системи. Розвиток методів організації обчислювального процесу в цьому напрямку привів до появи методу, відомого як віртуальна пам'ять [3].

Операційна система при використанні віртуальної пам'яті повинна вирішувати наступні задачі:

- розміщення даних на запам'ятовуваних пристроях різного типу, наприклад, частина