



В. С. Яковина, М. М. Сенів, В. В. Литвин, І. І. Симець

Національний університет "Львівська політехніка", м. Львів, Україна

ПРОГРАМНИЙ МОДУЛЬ РОЗВ'ЯЗУВАННЯ СИСТЕМ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ КОЛМОГОВОРА-ЧЕПМЕНА ДЛЯ АВТОМАТИЗАЦІЇ НАДІЙНІСНОГО ПРОЕКТУВАННЯ

Надзвичайно важливою компонентою якості складних технічних систем є їхня надійність, тобто властивість системи виконувати задані функції, зберігаючи в часі значення експлуатаційних показників у заданих межах, що відповідають умовам використання та заданим режимам технічного обслуговування, збереження і транспортування. Сучасний стан розвитку методів аналізу надійності технічних систем характеризується поєднанням аналітичних методів дослідження надійності з обчислювальними можливостями сучасних комп'ютерних засобів. Тому актуальною проблемою є автоматизація моделювання складних технічних систем яка, за умови відповідного рівня формалізації моделей, уможливує їх побудову та проведення аналізу надійності з використанням сучасних комп'ютерних засобів. В роботі описано програмний модуль автоматизованого розв'язання систем диференціальних рівнянь Колмогорова-Чепмена, який є складовою програмного комплексу автоматизації надійнісного проектування складних технічних систем. Розроблений модуль дає змогу розв'язувати системи диференціальних рівнянь Колмогорова-Чепмена без залучення спеціалізованих програмних продуктів (Matlab, Mathcad) для аналізу структурних схем надійності та автоматизованого визначення надійнісних показників складних технічних систем. Представлений модуль, за рахунок інтеграції в програмний комплекс автоматизації надійнісного проектування, дає змогу швидше опрацювати вхідні дані великих об'ємів та візуалізувати отримані результати обчислень.

Ключові слова: програмне забезпечення; надійність; структурна схема надійності (ССН); граф станів-переходів.

Вступ. Задача оцінки показників надійності сучасних складних технічних систем, особливо систем відповідального призначення, набуває дедалі більшої актуальності, оскільки їх відмови є недопустимі через можливі людські жертви чи величезні матеріальні втрати. Тому питанням забезпечення та оцінки показників надійності приділяють увагу вже на початкових етапах проектування таких систем, де формується їх архітектура, тобто структура та алгоритм функціонування. Технічні системи можуть бути невідновлюваними та відновлюваними, тривалого і короткого часу роботи, резервованими і нерезервованими. Сучасні технічні системи характеризуються великою кількістю елементів (модулів) і, як наслідок, – великими розмірностями простору станів, які можуть сягати десятків-сотень тисяч станів, що практично унеможливує їх аналіз без використання обчислювальних потужностей комп'ютерних засобів. Використання структурно-автоматних моделей (Modarres, Kaminskiy & Krivtsov, 1999; Bennetts, 1982; Polovko & Gurov, 2006) дає змогу автоматизувати побудову графа станів і переходів, але вимагає значних затрат на опис самої моделі.

Сучасний стан розвитку методів аналізу надійності складних технічних систем характеризується поєднанням аналітичних методів дослідження надійності з обчислювальними можливостями сучасних комп'ютерних засобів. Серед аналітичних методів найбільшого поширення набули методи, які базуються на теорії марковських процесів, та, водночас, використовують моделі надійності у вигляді системи диференціальних рівнянь Колмогорова-Чепмена, що дають змогу визначити часові залежності ймовірностей перебування системи у можливих станах (працездатності, відновлення, відмови тощо). Ці методи можна успішно застосувати для надійнісного аналізу технічних систем різного типу: невідновлюваних і відновлюваних, нерезервованих і резервованих з різними видами резервування, з різними пріоритетами ремонту тощо (Bennetts, 1982; Polovko & Gurov, 2006; Volochiy, 2004; Catelani, Ciani & Venzi, 2014). Але використання цих методів для реальних складних систем пов'язане з розв'язанням задач великої розмірності та складними розрахунками, які унеможливають виконання цих робіт вручну.

З огляду на це актуальною проблемою є автоматиза-

Інформація про авторів:

Яковина Віталій Степанович, д-р техн. наук, професор, завідувач кафедри програмного забезпечення.

Email: vitaliy.s.yakovyna@lpnu.ua; <https://orcid.org/0000-0003-0133-8591>

Сенів Максим Михайлович, канд. техн. наук, доцент, кафедра програмного забезпечення. Email: max1sudden@gmail.com

Литвин Василь Володимирович, д-р техн. наук, професор, завідувач кафедри інформаційних систем та мереж.

Email: vasy17.lytvyn@gmail.com; <https://orcid.org/0000-0002-9676-0180>

Симець Іван Ігорович, аспірант, кафедра програмного забезпечення. Email: cemivan1995@gmail.com

Цитування за ДСТУ: Яковина В. С., Сенів М. М., Литвин В. В., Симець І. І. Програмний модуль розв'язування систем диференціальних рівнянь Колмогорова-Чепмена для автоматизації надійнісного проектування. Науковий вісник НЛТУ України. 2019, т. 29, № 5. С. 141–146.

Citation APA: Yakovyna, V. S., Seniv, M. M., Lytvyn, V. V., & Symets, I. I. (2019). Software Module of Solving Systems of Chapman-Kolmogorov Differential Equations for Automation of Reliable Design. *Scientific Bulletin of UNFU*, 29(5), 141–146.

<https://doi.org/10.15421/40290528>

ція моделювання складних технічних систем яка, за умови відповідного рівня формалізації моделей, робить можливою їх побудову та проведення надійнісного аналізу з використанням можливостей сучасних обчислювальних систем. Для вирішення описаної вище проблеми доцільно використовувати спеціалізовані алгоритмічні та програмні засоби, які значно скорочують трудо- та часомісткий процеси оцінювання надійності технічних систем. Складовою такого програмного комплексу є програмний модуль розв'язування систем диференціальних рівнянь Колмогорова-Чепмена, що описаний в цій роботі.

Викладення основного матеріалу. Існують такі способи опису функціонування технічної системи в сенсі її надійності (Bennetts, 1982): структурна схема; функції алгебри логіки; граф станів-переходів; диференціальні та алгебраїчні рівняння.

Структурна схема. Структуру досліджуваного об'єкта представляють у вигляді сукупності однофункціональних вузлів (модулів) та необхідних взаємозв'язків між ними. Алгоритм функціонування відображає характер і послідовність взаємодії між вузлами (модулями). Опис способу функціонування технічної системи з погляду надійності починають із побудови її структурної схеми надійності. *Структурна схема надійності* – це графічне зображення логічної схеми системи у підсистемах і (або) елементах. Це дає змогу визначити шляхи сприятливого стану системи за тим, як пов'язані між собою блоки (підсистеми – елементи) (Modarres, Kaminskiy & Krivtsov, 1999). При графічному поданні структурної схеми надійності кожен елемент (модуль) системи зображають у вигляді геометричної фігури (найчастіше прямокутника) і з'єднують елементи між собою лініями таким способом, щоб структурна схема відображала умову працездатності системи (Software, 2019).

На рис. 1 показано приклади структурних схем надійності деяких нескладних систем.

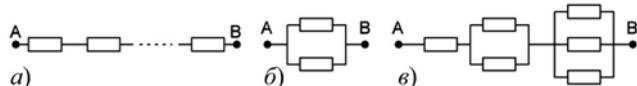


Рис. 1. Структурні схеми надійності простих технічних систем

Із наведених структурних схем наочно видно умови працездатності: наприклад, система (а) працездатна, якщо працездатні усі її елементи; система (б) працездатна, якщо працездатний будь-який її елемент; система (в) працездатна, якщо працездатний перший елемент і будь-який із двох у другій групі і будь-який із трьох у третій групі елементів (Bennetts, 1982; Polovko & Gurov, 2006).

Функції алгебри логіки. Найзручнішим у практичному використанні є спосіб одночасного визначення умов працездатності системи під час створення графічного зображення її структурної схеми надійності. Умови працездатності системи можна описувати за допомогою функцій алгебри логіки під час створення графічного зображення її структурної схеми надійності. Суть цього способу полягає у такому. Під час аналізу топології схеми спочатку виявляються сегменти схеми, які є послідовним з'єднанням елементів, а відтак розглядається спосіб з'єднання виявлених сегментів між собою. При послідовному з'єднанні елементи об'єднуються логічною операцією AND, а при паралельному з'єднанні

сегменти об'єднуються логічною операцією OR (Polovko & Gurov, 2006).

Матриця станів системи. У матриці станів міститься інформація про функціонування системи в сенсі її надійності. Кожен рядок матриці є вектором, компонентами якого слугують ознаки про те, в якому стані перебуває кожен елемент, коли сама система знаходиться в стані і. Елемент може перебувати не тільки у двох станах: працює чи відновлюється, елемент системи може перебувати в стані простою, який зумовлений різними причинами. Для подальшої роботи системи потрібне знання того, чи викликаний зазначений простій елемента чергою на відновлення чи відмовою інших елементів, або елемент є резервним і знаходиться в неавантаженому стані і т. ін. (Polovko & Gurov, 2006).

Граф станів-переходів. Через відмови і відновлення система в дискретні моменти часу переходить з одного стану в інший. У процесі тривалої експлуатації вона може побувати в кожному з можливих станів неодноразово. Тоді зручно процес її функціонування описувати графом, вузлам якого приписуються стани системи, а гілкам – можливі переходи зі стану в стан. Якщо в графі є n -вузлів, то серед них буде k -вузлів, які відповідають несправним станам, і $(n-k)$ – справним. Множина станів і переходів системи є вхідними даними для подальшого дослідження надійності технічної системи та складання диференціальних рівнянь Колмогорова-Чепмена (Volochiy, 2004; Catelani, Ciani & Venzi, 2014).

Диференціальні і алгебраїчні рівняння. Надійнісну поведінку системи в часі можна інтерпретувати як випадковий процес із скінченною множиною значень і неперервною зміною аргументу (часу), тобто як дискретно-неперервний стохастичний процес. Часові залежності безумовних ймовірностей значень випадкового процесу та умовних ймовірностей переходів описуються системою диференціальних рівнянь Колмогорова-Чепмена такого виду:

$$\frac{dP_i(t)}{dt} = -\sum_{j=1}^N \lambda_{ij}(t)P_i(t) + \sum_{j=1}^N \lambda_{ji}(t)P_j(t); i, j = \overline{1, N}, \quad (1)$$

де: $P_i(t)$, $P_j(t)$ – ймовірності значень випадкового процесу (ймовірності перебування системи) у момент часу t відповідно у станах x_i та x_j ; $\lambda_{ij}(t)$ – інтенсивність переходу із стану x_i у стан x_j ; $\lambda_{ji}(t)$ – інтенсивність переходу із стану x_j у стан x_i ; N – кількість можливих станів системи.

Розв'язок системи таких рівнянь дає змогу визначити такі показники надійності: ймовірність безвідмовної роботи, функцію та коефіцієнт готовності, середня тривалість безвідмовної роботи, середнє напрацювання на відмову, середня тривалість відновлення тощо (Volochiy, 2004; Catelani, Ciani & Venzi, 2014; Du, Guo & Beeram, 2008; Gorbatyy, 2014).

Аналіз сучасного стану ПЗ для надійнісного проектування. Відомі програмні продукти для надійнісного проектування, такі як RAM Commander (ALD, Israel), ReliaSoft Synthesis Master Suite (ReliaSoft, USA), Reliability Workbench (Isograph, UK), FavoWeb та інші, дають змогу автоматизувати процес розрахунку показників надійності, проте вимагають попередньої "ручної" побудови графа станів та переходів (Bennetts, 1982; Polovko & Gurov, 2006; Volochiy, 2004).

RAM Commander (ціна \$1,810.00/one-time/user) (Commander, 2019) – комплексний пакет програм для вдосконалення і дослідження надійності електромеханічних,

механічних та інших систем, створений провідними фахівцями з надійності. RAM Commander охоплює весь спектр завдань, пов'язаних з надійністю, починаючи від створення й імпортування переліку матеріалів і закінчуючи складанням звітів щодо надійності, готовності, ремонтпридатності (рис. 2).

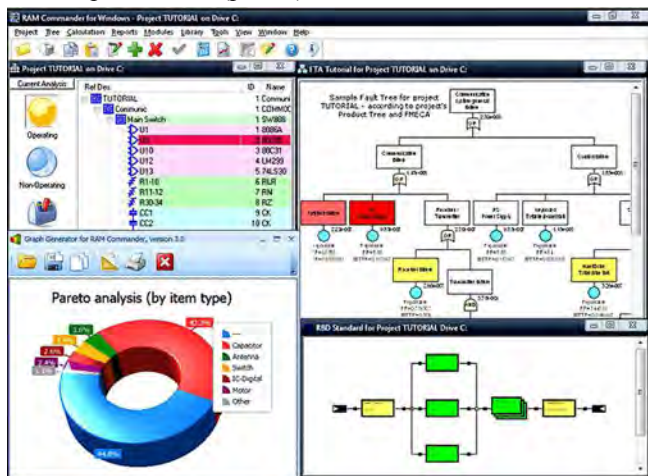


Рис. 2. Робоче вікно RAM Commander

RAM Commander дає змогу проводити розрахунок функціональної надійності, аналіз видів і наслідків потенційних відмов по методу FMEA / FMESA, будувати і аналізувати дерева відмов (FTA – fault tree analysis), створювати температурні криві, оптимізувати ремонт і кількість запасних частин, вибирати і робити коригувальні дії, проводити аналіз чутливості і влаштовувати експертні оцінки.

Однією з найважливіших переваг RAM Commander є відповідність проведених у ньому розрахунків основним міжнародним стандартам: MILHDBK-217 F Notice, Бельоре Issue, HRD4, HRD5 British Telecom.

FavoWeb – це веб-орієнтована FRACAS-система (англ. *Failure Reporting Analysis and Corrective Action System* – Система оповіщення про відмови, аналіз і коригувальні дії).

Система дає змогу створювати бази даних, переводячи різноманітні дані в структурованій інформації про надійність. Має потужний механізм коригувальних дій: підтримка роботи груп аналізу відмов / дефектів / матеріалів, аналіз глибинних причин відмов, усунення проблем. Містить модуль Workflow для автоматизації сповіщення про відмови і серійні номери.

Reliability Workbench (\$1,499.00/one-time/user) (Workbench, 2019) – це система, яка складається з набору продуктів для виконання ряду стандартних методів аналізу надійності. Система автоматично формує розрахункові аналітичні моделі надійності і безпеки систем та обчислює ймовірність безвідмовної роботи, середнє напруження на відмову, коефіцієнт готовності, середня тривалість відновлення, ймовірність відмови відновленої системи.

Описані вище програмні комплекси за всіх їхніх переваг мають низку недоліків, серед яких основними є висока вартість ліцензії, складність навчання й обслуговування програмного засобу (для роботи з ними потрібно мати певну кваліфікацію в області теорії надійності), потребу певних "ручних" розрахунків вхідних даних (умова працездатності, множина станів тощо). Тому виникає потреба у створенні простого та багатфункціонального програмного комплексу для надійнісного

проектуювання, який також мав би високу швидкодію. Складовою такого комплексу є програмний модуль розв'язування систем диференціальних рівнянь Колмогорова-Чепмена.

Розроблення та інтеграція програмного модуля автоматизованого розв'язання систем диференціальних рівнянь Колмогорова-Чепмена. Розроблений модуль є функціональною частиною програмного комплексу для аналізу надійнісних характеристик технічних систем на основі структурних схем надійності. Для повноцінної роботи модуль потребує вхідної інформації про стани і всі зв'язки між ними для досліджуваної системи та інформації про компоненти системи (інтенсивності відмов і відновлення модулів).

Тому для зручної взаємодії та обміну інформацією між компонентами програмного комплексу розроблено гнучку модульну архітектуру із використанням модуля контролю (рис. 3). Цей комплекс складається із підсистем, які відповідають за візуалізацію і обчислення певних характеристик, а також модуля контролю, який забезпечує швидкий і зручний зв'язок між функціональними модулями.

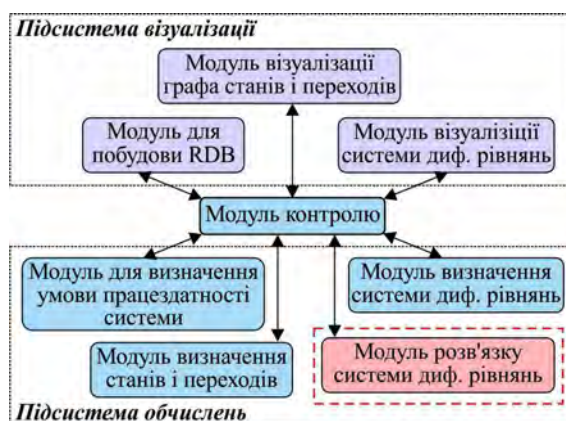


Рис. 3. Місце модуля у програмному комплексі

Розроблений програмний модуль розв'язує системи диференціальних рівнянь методом Рунге-Кутта. Унаслідок роботи створеного модуля отримуємо ймовірності перебування досліджуваної системи у всіх можливих станах на заданому проміжку часу, і на основі цих результатів можна визначити інтеграл (2), який є характеристикою надійності системи – середня тривалість безвідмовної роботи (середня тривалість напрацювання до відмови)

$$t_{cep} = \int_0^{\infty} p(t) dt . \quad (2)$$

Для програмної реалізації цього модуля розроблено гнучку ієрархію класів, яка дає змогу легко модифікувати цей модуль (рис. 4), а саме швидко і зручно додавати функції для розв'язку системи диференціальних рівнянь іншими відомими методами (Ейлера, Мілна тощо) або із використанням стороннього API (Matlab, Matcad).

Нижче наведено перелік основних класів модуля розв'язування систем диференціальних рівнянь Колмогорова-Чепмена:

- RungeKuttaMethod – клас призначений для розв'язку системи диференціальних рівнянь методом Рунге-Кутта.
- ControlModule – клас призначений для координування роботи системи, відповідає за обмін даними і зв'язок між програмними модулями системи.

- SystemState, State – класи для представлення матриці станів і переходів між ними.
- SystemConfiguration – клас для різноманітних програмних конфігурацій.

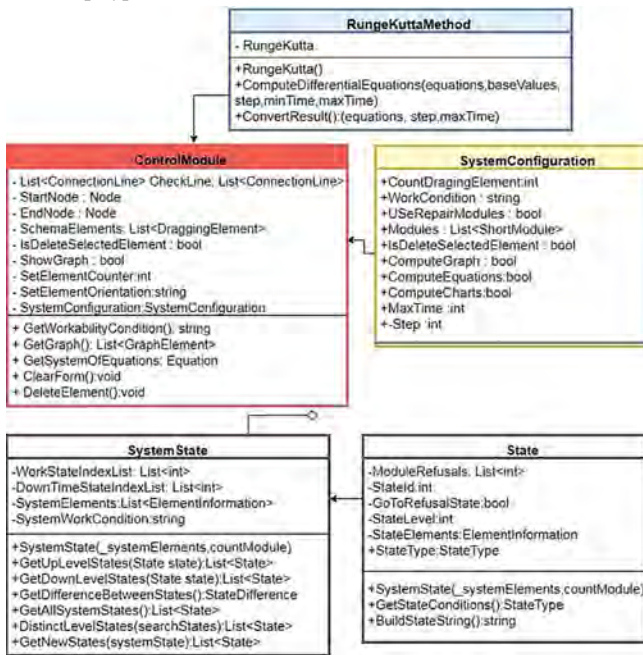


Рис. 4. Діаграма класів модуля розв'язування систем диференціальних рівнянь Колмогорова-Чепмена

Вирішено використовувати програмну реалізацію методу Рунге-Кутта, тому що цей метод має високу точність і, незважаючи на свою трудомісткість, широко використовується при чисельному рішенні систем диференціальних рівнянь. Метод Рунге-Кутта стійкий до помилок у початкових умовах і до спотворень у правій частині рівняння, оскільки кінцеві зміни цих величин призводять до кінцевих змін у результатах обчислень, а не накопичуються при просуванні від кроку до кроку. На рис. 5 зображено реалізацію методу Рунге-Кутта мовою програмування C#.

```

RungeKuttaMethod.cs -> X GraphWindow.xaml.cs GraphWindow.xaml MatLabEquation.cs
VIPER
VIPER.System_of_differential_equations.RungeKuttaMethod
19 for (int k = 0; k < maxTime; k+=step)
20 {
21     var resultStep1 = new List<double>();
22     for (int i = 0; i < equationCount; i++)
23     {
24         k0[i] = k * equationCount = Func(initialValues, i, equationCount, coefficients) * step;
25     }
26     for (int i = 0; i < equationCount; i++)
27     {
28         k1_help[i] = initialValues[i] + k0[i] * k * equationCount / 2;
29     }
30     for (int i = 0; i < equationCount; i++)
31     {
32         k2[i] = k * equationCount = Func(k1_help, i, equationCount, coefficients) * step;
33     }
34     for (int i = 0; i < equationCount; i++)
35     {
36         k2_help[i] = initialValues[i] + k2[i] * k * equationCount / 2;
37     }
38     for (int i = 0; i < equationCount; i++)
39     {
40         k3[i] = k * equationCount = Func(k2_help, i, equationCount, coefficients) * step;
41     }
42     for (int i = 0; i < equationCount; i++)
43     {
44         k3_help[i] = initialValues[i] + k3[i] * k * equationCount;
45     }
46     for (int i = 0; i < equationCount; i++)
47     {
48         k4[i] = k * equationCount = Func(k3_help, i, equationCount, coefficients) * step;
49     }
50     for (int i = 0; i < equationCount; i++)
51     {
52         resultStep1[i] = initialValues[i] + i.0 / 4.0 * (k0[i] + k * equationCount) + 2 * k1[i] + k * equationCount;
53     }
54     for (int i = 0; i < equationCount; i++)
55     {
56         initialValues[i] = resultStep1[i] + k * equationCount;
57         resultStep1.Add(resultStep1[i] + k * equationCount);
58     }
59 }
  
```

Рис. 5. Реалізація методу мовою програмування C#

Практичне застосування програмного модуля автоматизованого розв'язання систем диференціальних рівнянь Колмогорова-Чепмена та верифікація отриманих результатів. Для демонстрації роботи цього модуля

потрібно спочатку побудувати структурну схему надійності (із відповідними налаштуванням параметрів модулів) і визначити умову працездатності системи (рис. 6) (Seniv, Yakovyna & Symets, 2018).

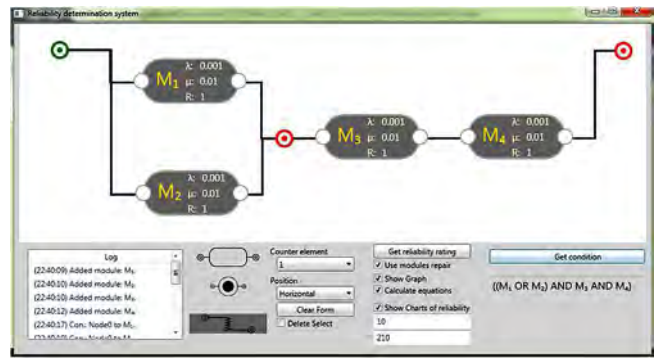


Рис. 6. Вікно побудови структурної схеми надійності

Далі на основі умови працездатності визначається граф станів і переходів (Bobalo, Yakovyna, Seniv & Symets, 2018) для системи і вже на основі матриці станів за допомогою методу, описаного в цій роботі, визначається і розв'язується системи диференціальних рівнянь Колмогорова-Чепмена (рис. 8).

Так виглядає вікно відображення системи диференціальних рівнянь Колмогорова-Чепмена, де можна побачити таблицю з відображенням системи і відповідно типу стану, який рівняння описує (рис. 7).

Equations	StateType
1 $dp_1(t) / dt = - (\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4) * P_1(t)$	WorkingState
2 $dp_2(t) / dt = - (\mu_1 + \lambda_2 + \lambda_3 + \lambda_4) * P_2(t) + (\lambda_1) * P_1(t)$	WorkingState
3 $dp_3(t) / dt = - (\lambda_1 + \mu_2 + \lambda_3 + \lambda_4) * P_3(t) + (\lambda_2) * P_1(t)$	WorkingState
4 $dp_4(t) / dt = - (\mu_3) * P_4(t) + (\lambda_3) * P_1(t)$	DowntimeState
5 $dp_5(t) / dt = - (\mu_4) * P_5(t) + (\lambda_4) * P_1(t)$	DowntimeState
6 $dp_6(t) / dt = - (\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4) * P_6(t) + (\mu_1) * P_2(t)$	WorkingState
7 $dp_7(t) / dt = - (\mu_2 + \mu_2) * P_7(t) + (\lambda_2) * P_2(t) + (\lambda_1) * P_3(t)$	DowntimeState
8 $dp_8(t) / dt = - (\mu_2 + \mu_2) * P_8(t) + (\lambda_3) * P_2(t)$	DowntimeState
9 $dp_9(t) / dt = - (\mu_2 + \mu_4) * P_9(t) + (\lambda_4) * P_2(t)$	DowntimeState
10 $dp_{10}(t) / dt = - (\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4) * P_{10}(t) + (\mu_1) * P_3(t)$	WorkingState
11 $dp_{11}(t) / dt = - (\mu_2 + \mu_3) * P_{11}(t) + (\lambda_3) * P_3(t)$	DowntimeState

Рис. 7. Система диференціальних рівнянь Колмогорова-Чепмена

Після розв'язку модуль дає можливість перегляду результатів у відповідному табличному представленні на проміжку заданого часу (рис. 9).

Для верифікації результатів роботи цього модуля виконано порівняння розв'язку систем диференціальних рівнянь для структурних схем різної конфігурації модулів та їх характеристик. Для порівняння вибрано такі відомі програмні комплекси для розв'язку систем диференціальних рівнянь:

- Matlab – пакет прикладних програм для числового аналізу, а також мова програмування, що використовується в цьому пакеті. Система створена компанією The MathWorks і є зручним засобом для роботи з математичними матрицями, малювання функцій, роботи з алгоритмами, створення робочих оболонок (*User Interfaces*) з програмами в інших мовах програмування;
- Mathcad – система комп'ютерної алгебри з класу систем автоматизованого проектування, орієнтована на підготовку інтерактивних документів з обчисленнями і візуальним супроводженням, відрізняється легкістю використання і застосування для колективної роботи;
- Scilab Online – он-лайн сервіс наукових програм для чисельних обчислень, що надає потужне відкрите середовище для інженерних і наукових розрахунків.

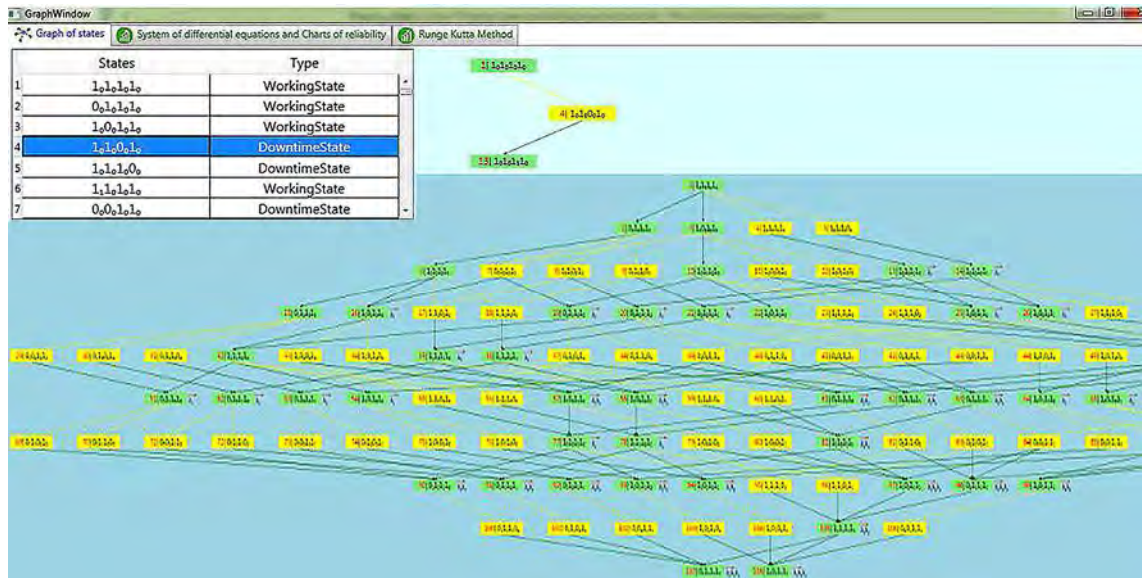


Рис. 8. Вкладка відображення графа станів і переходів

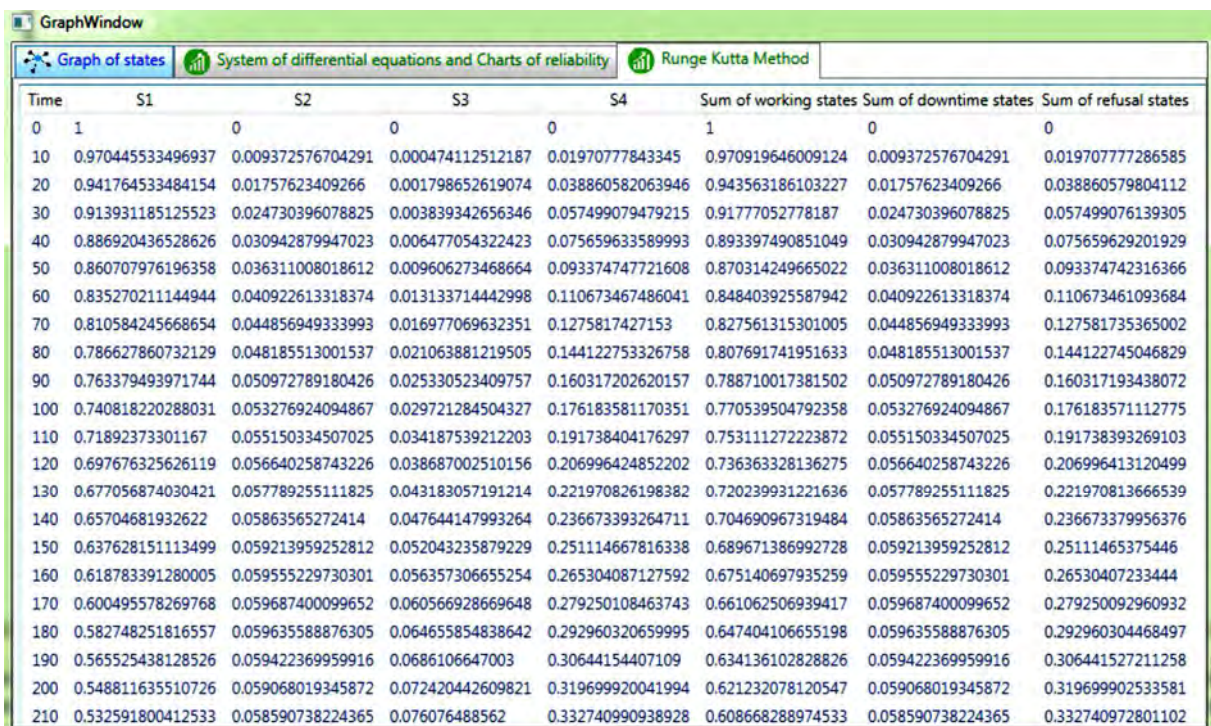


Рис. 9. Результат роботи модуля

Таблиця. Тривалість розв'язання систем диференціальних рівнянь різними програмними засобами

№	Кількість рівнянь	Тривалість роботи					
		модуля, с	Matlab, с	Mathcad, с	Scilab, с	Matlab API, с	Scilab API, с
1	3	0,0827	0,0691	0,0701	0,0801	1,123	1,902
2	15	0,142	0,135	0,129	0,139	1,271	2,235
3	63	0,196	0,184	0,178	0,194	1,491	2,498
4	255	0,221	0,190	0,195	0,230	1,702	2,635
5	1023	0,299	0,250	0,274	0,302	1,995	2,891
6	4095	0,402	0,320	0,307	0,406	2,002	3,000
7	7789	0,559	0,500	0,530	0,551	2,254	3,152
8	12789	0,692	0,586	0,575	0,680	2,392	3,302
9	20548	0,721	0,642	0,652	0,706	2,525	3,502
10	29632	0,803	0,713	0,703	0,789	2,878	3,789
11	51789	1,127	0,981	0,932	1,103	3,541	4,256
12	121481	2,154	1,571	1,612	1,901	5,032	6,120

Виконано тестування швидкодії роботи розробленого модуля і перелічених вище програмних засобів. Також виконані тестові запуски безпосередньо в програм-

них засобах і в розробленому модулі із використанням API функцій Matlab і Scilab Online. Результати наведено в таблиці та на рис. 10.

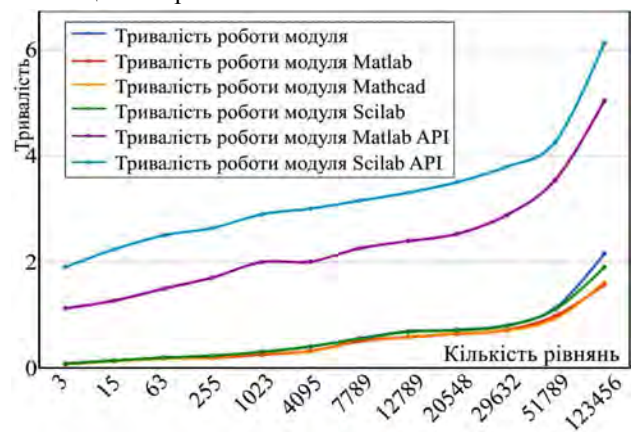


Рис. 10. Залежність тривалості роботи компонент від кількості рівнянь для різних систем

Дані із таблиці та графіку свідчать про те, що розроблений модуль для розв'язку системи диференціальних рівнянь дещо поступається у швидкодії програмним засобам Matlab і Mathcad, і майже на рівні працює з он-лайн сервісом Scilab. Також, на основі досліджень, впливає, що розроблений модуль працює значно швидше, ніж ПЗ із використанням API функцій. Це відбувається тому, що більшість часу витрачається на відправлення, опрацювання і отримання даних у виконавчі середовища, аніж розв'язок рівнянь у цих середовищах.

Враховуючи отримані результати, можна зробити висновок, що найдоцільніше використовувати власний модуль для розв'язку, хоча він і повільніший, ніж відомі програмні засоби, але використання цих засобів за допомогою API функцій є недоцільним, оскільки багато часу витрачається на зв'язок з середовищем.

Висновки. У роботі представлено програмний модуль розв'язування систем диференціальних рівнянь Колмогорова-Чепмена для автоматизації надійнісного проектування. Розроблений модуль дає змогу розв'язувати системи диференціальних рівнянь Колмогорова-Чепмена методом Рунге-Кутта без залучення спеціалізованих програмних продуктів (Matlab, Mathcad) для аналізу структурних схем надійності та автоматизованого визначення надійнісних показників складних технічних систем. Представлений модуль, за рахунок інтеграції в програмний комплекс автоматизації надійнісного проектування, дає змогу в 2-3 рази швидше, ніж програмні додатки з використанням API-функцій спеціалізованих програмних продуктів (Matlab, Mathcad), опрацьовувати вхідні дані великих об'ємів та візуалізувати отримані результати обчислень. Використання програмного модуля розв'язування систем диференціальних рівнянь Колмогорова-Чепмена для надійнісного проектування також знімає проблему купівлі ліцензій спеціалізованих програмних продуктів (Matlab, Mathcad).

Перелік використаних джерел

- Bennetts, R. G. (1982). Analysis of Reliability Block Diagrams by Boolean Techniques. *IEEE transactions on reliability*, R-31(2), 159–166.
- Bobalo, Yu., Yakovyna, V., Seniv, M., & Symets, I. (2018). Technique of automated construction of states and transitions graph for the analysis of technical systems reliability. *Proceedings of the 13th International scientific and technical conference CSIT-2018*, (pp. 314–317). September 11–14. Lviv.
- Catelani, M., Ciani, L., & Venzi, M. (2014). Improved RBD analysis for reliability assessment in industrial application. *IEEE International Instrumentation and Measurement Technology Conference Proceedings*, (pp. 670–674). IEEE, Montevideo, Uruguay. <https://doi.org/10.1109/I2MTC.2014.6860827>
- Commander. (2019). RAM Commander. Retrieved from: <http://www.alsoftware.com>, last accessed 2018/12/20.
- Du, X., Guo, J., & Beeram, H. (2008). Sequential optimization and reliability assessment for multidisciplinary systems design. *Structural and Multidisciplinary Optimization*, 35(2), 117–130. <https://doi.org/10.1007/s00158-007-0121-7>
- Gorbatyy, I. V. (2014). Investigation of the technical efficiency of state-of-the-art telecommunication systems and networks with limited bandwidth and signal power. *Automatic Control and Computer Sciences*, 48(1), 47–55. <https://doi.org/10.3103/S0146411614010039>
- Modarres, M., Kaminskiy, M., & Krivtsov, V. (1999). *Reliability Engineering and Risk Analysis*. Marcel Decker, NYC, NY USA.
- Polovko, A. M., & Gurov, S. V. (2006). *The basics of reliability theory*. St. Petersburg: BHV-Peterburg, 704 p.
- Seniv, M., Yakovyna, V., & Symets, I. (2018). Software for visualization of reliability block diagram and automated formulation of operability conditions of technical systems. *Perspective technologies and methods in MEMS design (MEMSTECH2018): Proceedings of the 14th International Conference*, (pp. 191–195). April 18–22. Lviv – Polyana.
- Software. (2019). Relx Software. Retrieved from: <http://www.relexsoftware.it>, last accessed 2019/05/17.
- Volochiy, B. Ya. (2004). *Technology of modeling algorithms for behavior of information systems*. Lviv: Lviv Polytechnic Publishing House, 220 p.
- Workbench. (2019). Reliability Workbench. Retrieved from: <http://www.armsreliabilitysoftware.com/software-solutions>, last accessed 2018/06/15.

V. S. Yakovyna, M. M. Seniv, V. V. Lytvyn, I. I. Symets
Lviv Polytechnic National University, Lviv, Ukraine

SOFTWARE MODULE OF SOLVING SYSTEMS OF CHAPMAN-KOLMOGOV DIFFERENTIAL EQUATIONS FOR AUTOMATION OF RELIABLE DESIGN

The extremely important component of the quality of complex technical systems is their reliability, that is, the specific feature of the system to perform specified functions, while maintaining the value of operational characteristics within the specified limits, conforming to the conditions of use and the specified modes of maintenance, storage and transportation. The current state of development of the methods for analysing the reliability of technical systems is characterized by a combination of analytical research methods of reliability with the computing capabilities of modern computer facilities. Technical systems are characterized by a large number of elements (modules) and, consequently, large dimensions of space of the states that can reach tens and hundreds of thousands of states, which make it virtually impossible to analyze them without the use of computation power of computer facilities. Therefore, the relevant problem is the automation of complex technical systems modeling which, when the appropriate level of model formalization is provided, enables model development and reliability analysis using modern computers. This publication describes the software module for the automated solution of the systems of Chapman–Kolmogorov differential equations, which are an integral part of the software complex for automation of reliable design of complex technical systems. The developed module allows solving systems of Chapman–Kolmogorov differential equations without the involvement of specialized software products (Matlab, Mathcad), for the analysis of structural schemes of reliability and automated determination of reliability indexes of complex technical systems. The presented module, by integrating into a software automation complex of reliable design, enables more rapid processing of large volumes of input data and visualizing the obtained results of calculations.

Keywords: software; reliability; reliability block diagram (RBD); states and transitions graph.