

**Юрович Іван Васильович**

Аспірант кафедри системного програмування і спеціалізованих комп'ютерних систем,

<https://orcid.org/0009-0005-9575-065X>

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»,  
Київ

**Зайцев Володимир Григорович**

Доктор технічних наук, професор, професор кафедри системного програмування і спеціалізованих комп'ютерних систем, <https://orcid.org/0009-0009-9917-5364>

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»,  
Київ

## МЕТОД ВИЗНАЧЕННЯ ЧАСОВИХ ХАРАКТЕРИСТИК СИСТЕМ РЕАЛЬНОГО ЧАСУ

**Анотація.** Системи реального часу широко розповсюджені і активно впроваджуються у передових розробках багатьох галузей, від медицини до космічної промисловості. Дослідження і моделювання систем реального часу є важливою задачею на етапі проєктування системи, адже воно дає змогу визначити, чи відповідає система, що моделюється, заданим часовим характеристикам, і відповідно визначити спроможність системи задовольнити часові вимоги. Адже успішність роботи систем реального часу залежить не лише від її логічної коректності, але і від часу, за який система генерує результат. Враховуючи, що є різні типи задач, наприклад синхронні і асинхронні, паралельні і послідовні, визначення їх часових характеристик на стадії проєктування є доволі складною проблемою. Також слід враховувати послідовність задач і час їх появи, адже є випадки, коли одна задача має залежність на результат виконання іншої задачі, відповідно час появи нової задачі в черзі планувальника залежить від часу завершення попередньої задачі. У попередніх роботах були проведені дослідження методів оцінки часових характеристик задач в системах реального часу шляхом аналізу даних, отриманих моделюванням розподілу процесорного часу між задачами згідно обраних алгоритмів планувальника з використанням моделі сіток Петрі в однопроцесорних та багатопроцесорних системах. Методи гарантували отримання часових характеристик задач при обранні конкретного типу процесора і планувальника, що є необхідним для початку технічного проєктування системи реального часу. Проте методи не враховували динамічну природу появи задач, що є невід'ємною складовою новітніх систем реального часу. У роботі запропоновано метод для визначення часових характеристик систем реального часу на етапі проєктування системи. Запропонований метод допомагає визначити час появи задачі, враховуючи різні типи задач та їх залежності. Визначені часові характеристики можуть бути використані у подальшому для моделювання роботи системи та визначення оптимального алгоритму планувальника задач.

**Ключові слова:** система реального часу (СРЧ); модель; часові характеристики; планувальник задач; сітка Петрі; адаптивний круїз-контроль

### Постановка проблеми

Системи реального часу набувають все більш вагомого значення у сучасному світі, враховуючи швидкі темпи технологічного розвитку та зростання вимог до швидкості і надійності опрацювання інформації. Зростання обсягу інформації, яка генерується та обробляється різноманітними системами ставить виклик забезпеченню швидкості та точності опрацювання цих даних.

Сучасні системи реального часу можуть мати десятки функцій, які працюють за допомогою

електронних датчиків, їх контролерів та технічних засобів, що керують відповідними елементами системи. Деякі системи мають вимогу невідкладної реакції на зовнішні події, що дає змогу системі зберегти стабільність у критичній ситуації.

Дослідження та розробки у сфері систем реального часу відіграють критичну роль у забезпеченні ефективності, безпеки та надійності різноманітних систем і застосувань у сучасному технологічному світі. Вони допомагають вирішувати виклики, пов'язані з обробкою даних у реальному часі, і сприяють подальшому розвитку інноваційних технологій.

Успішність роботи системи реального часу залежить не тільки від її логічної правильності, але й від швидкості видачі результату. Під час проектування такої системи важливо визначити часові параметри, які використовуються для моделювання її роботи. Для цього необхідно знати такі часові характеристики задач систем реального часу, як час появи задачі, час її виконання і крайній термін завершення виконання задачі. Задачі можуть бути послідовними та паралельними одна до одної. Також задачі можуть з'являтися по ходу виконання програми, адже поява деяких задач залежить від результату виконання попередніх задач, що ускладнює процес моделювання і верифікування СРЧ відповідності заданим часовим вимогам. Ці виклики спонукають до створення методу визначення часових характеристик СРЧ, що враховуватиме динамічну появу задач під час роботи системи і може надалі бути використаний при моделюванні роботи системи, відповідно до обраного алгоритму планувальника задач.

### **Аналіз основних досліджень і публікацій**

У пропонованій роботі розглянуто алгоритм роботи адаптивного круїз-контролю як приклад системи реального часу. Ця технологія широко розповсюджена у сучасних автомобілях і є частиною удосконаленої системи допомоги водію (ADAS). Покращення систем реального часу в автомобільній сфері, зокрема дослідження, що базуються на CAN шині, активно ведуться протягом останніх десятиліть та впроваджуються у нові покоління автомобілів.

Tindell та інші [1, с. 149] показали, як дослідження планування на випередження з фіксованим пріоритетом для однопроцесорних систем можуть бути адаптовані та застосовані до планування повідомлень на CAN. Цей аналіз забезпечив метод розрахунку найгіршого часу відповіді для всіх повідомлень CAN. Використовуючи цей аналіз, стало можливим розробити системи на основі CAN з правильною синхронізацією, забезпечуючи гарантії того, що всі повідомлення і сигнали, які вони несуть, будуть відповідати встановленим термінам.

Дослідження Tindell та інших мали суттєвий вплив на розробку периферійних пристроїв CAN на чіпі, таких як, наприклад, Motorola msCAN [2, с. 17], і стали основою для великої кількості робіт з теорії планування та роботи під час виникнення помилок для CAN (Punnekkat та ін. [3, с. 258]; Nolte та ін. [4, с. 197; 5, с. 200]; Broster та ін. [6, с. 269; 7, с. 55]; Hansson та ін. [8, с. 1240]; Broster та Burns [9, с. 410]).

У 1995 р. дослідження Tindell та інших [1, с. 149] було використано компанією Volvo Car Corporation і успішно застосовано в конфігурації та аналізі шин

CAN для майбутнього Volvo S80 (Casparsson та ін. [10, с. 1]). Після успіху цього проекту компанія Volcano Communications Technologies AB використала аналіз Tindell та ін. як основу комерційного інструменту для аналізу планування CAN під назвою Volcano Network Architect.

З 1998 р. інструменти та концепція Volcano (Casparsson та ін. [10, с. 1]) використовувалися при проектуванні та розробці мереж CAN та електронних систем для багатьох нових моделей автомобілів компанії Volvo, а також багатьох інших моделей автомобілів від різних виробників.

До робіт Tindell та ін. використання CAN шини на рівні до 30–40% було типовим для автомобільних програм, а також потрібно було проводити ретельне тестування системи, щоб мати впевненість у тому, що повідомлення відповідатимуть встановленим часовим проміжкам. З появою систематичного підходу, заснованого на аналізі планування, використання шини CAN збільшилося майже до 80% [11, с. 243], гарантуючи при цьому дотримання часових обмежень.

Однак у подальших дослідженнях Davis та ін. [11, с. 243] звернули увагу, що аналіз Tindell та ін. [1] має суттєвий недолік, адже він може потенційно призвести до обчислення оптимістичного найгіршого часу відповіді для CAN повідомлень та порушень часових обмежень. Davis та ін. [11, с. 248] запропонували новий підхід, який усуває оптимізм в аналізі, використовуючи наявні результати щодо випереджаючого планування із фіксованими пріоритетами та довільними термінами, а також щодо планування без випередження з фіксованим пріоритетом. У вищезгаданій роботі було використано планування без випередження, адже його можна розглядати як окремий випадок планування з випередженням: пріоритет завдання піднімається до найвищого перед виконанням, щоб запобігти випередженню. Девіс та ін. також показали, що оптимальне призначення пріоритету Audsley [12, с. 9] може бути застосоване до повідомлень CAN.

Останнім часом було проведено низку досліджень щодо використання сіток Петрі у контексті систем реального часу. Зокрема, сітки Петрі можуть бути використані для визначення відповідності системи заданим часовим вимогам [13, с. 49] [14, с. 79] [15, с. 45] на етапі проектування системи. Недавні дослідження доводять [13, с. 60], що таким чином можна визначити терміни виконання задач та вибрати оптимальний метод реалізації програмного і апаратного забезпечення на початкових етапах розроблення системи. У вищезгаданій роботі було запропоновано спосіб оцінки часових характеристик задач систем реального часу за допомогою аналізу даних, які були

отримані при моделюванні. Під час моделювання відбувався розподіл процесорного часу між задачами відповідно до вибраних алгоритмів планування задач та з використанням програмної моделі системи, що являє собою сітку Петрі.

### Мета статті

Метою роботи є дослідження методу визначення часових характеристик систем реального часу, які у подальшому можуть бути використані для моделювання роботи системи, відповідно до вибраного алгоритму планувальника задач з метою вибору найпридатнішого алгоритму планувальника задач.

### Виклад основного матеріалу

Система реального часу (СРЧ) це набір програмного й апаратного забезпечення, що обробляє події зовнішнього (по відношенню до системи) середовища, дотримуючись необхідних часових обмежень. Успішність роботи такої системи визначається не тільки її логічною правильністю, а й часом, за який система продукує результат. Важливо зазначити, що система реального часу часто ототожнюється зі швидкодіючою системою, проте це не відповідає дійсності. Для систем реального часу затримки у виконанні практично нівелюються, якщо завдання було виконано у відведений проміжок часу. Відповідно, вимоги щодо СРЧ поділяють на два типи: слабкі (soft) – якщо недотримання часових обмежень порушено, то продуктивність системи знижується, проте система продовжує функціонувати; жорсткі (hard) – недотримання часових характеристик вважається за збій у функціонуванні всієї системи.



Рисунок 1 – Типовий алгоритм системи реального часу

Алгоритм роботи будь-якої системи реального часу може бути спрощений і представлений за допомогою трьох основних дій: збір даних, обробка даних і керування системою, у відповідності до

отриманих результатів (рис. 1). На етапі збирання даних проводиться опитування датчиків системи та збирання необхідної інформації. Протягом етапу обробки даних відбувається аналіз вхідних даних, обчислення похідних значень, специфічних для конкретної системи. Після того як дані зібрані і опрацьовані, відбувається коригування системи, шляхом управління наявними в системі апаратними і програмними засобами.

Задачі систем реального часу, відносно характеру їх появи, поділяються на декілька типів:

– періодичні – задачі, які мають виконуватися через певні регулярні інтервали часу. Тобто, необхідність їх виконання виникає в моменти часу, що повторюються через рівні інтервали;

– спорадичні – задачі, які виконуються нерегулярно або у відповідь на певні події чи умови. Відповідно, необхідність їх виконання виникає у випадкові моменти часу;

– фонові – задачі, які виконуються лише за наявності вільних ресурсів. Вони не мають сурових часових обмежень і можуть бути відкладені або перервані.

Задачі СРЧ характеризуються такими часовими параметрами [13]:

$r$  – (reachability time) – момент появи завдання;  
 $d$  – (absolute deadline) – абсолютний крайній термін виконання, тобто момент часу, до якого задача повинна бути завершена;

$s$  – (start time) – момент часу, коли задача фактично починає виконуватись на процесорі;

$c$  – (completion time) – момент часу, коли задача закінчила роботу;

$D$  – (relative time) – час між появою задачі до її крайнього терміну виконання:  $D = d - r$ ;

$E$  – (execution time) – час виконання задачі:  $e = c - s$ ;

$R$  – (response time) – час відгуку, тобто час від появи задачі до її фактичного завершення виконання:  $R = c - r$ .

На рис. 2 проілюстровані параметри задачі СРЧ.

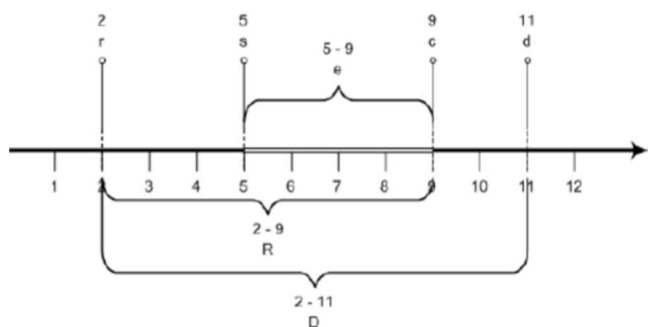


Рисунок 2 – Параметри задачі СРЧ

Для того щоб визначити терміни виконання задач та вибрати оптимальний метод реалізації програмного і апаратного забезпечення до етапу розробки потрібно знати часові характеристики задач системи, зокрема час появи задачі, час виконання задачі та кінцевий термін виконання задачі. Параметри задач системи реального часу можна визначити, якщо розбити всю систему на окремі задачі і детально проаналізувати кожну з них. Для цього необхідно визначити послідовність виконання задач, а також їх взаємозалежності. Такий підхід допомагає чітко зрозуміти, як кожна задача впливає на інші, які ресурси вона потребує, і які часові обмеження мають бути дотримані.

У рамках цього дослідження проаналізуємо алгоритм роботи адаптивного круїз-контролю. Круїз-контроль – це система, що підтримує сталу швидкість автомобіля без участі водія. Наприклад, на спусках система може пригальмовувати, а на підйомах може автоматично додавати обертів двигуна, щоб швидкість авто залишалася сталою. Адаптивний круїз-контроль виконує всі ті ж функції, що і звичайний, але додатково слідкує за відстанню до автомобіля, що рухається попереду.

Система адаптивного круїз контролю може бути представлена за допомогою таких задач:

- T1 – зчитування потрібної швидкості;
- T2 – визначення поточної швидкості;
- T3 – обчислення зміни швидкості, яка потрібна для досягнення заданих параметрів.

Якщо поточне авто їде повільніше заданої швидкості, то обчислюється величина збільшення обертів двигуна, і виконується задача T4. Якщо ж поточне авто їде швидше очікуваного, то обчислюється величина зменшення швидкості. Якщо авто їде занадто швидко, то буде виконана задача T6, а якщо перевищення швидкості незначне, то виконається задача T5;

- T4 – збільшення обертів двигуна;
- T5 – зменшення обертів двигуна для незначного зменшення швидкості;
- T6 – гальмування для швидкого зменшення швидкості;
- T7 – визначення відстані до авто попереду;
- T8 – обчислення прискорення поточного авто;

- T9 – обчислення прискорення авто попереду.

Задачі системи можна класифікувати за типом так:

- введення даних (T1, T2, T7);
- обробка даних (T3, T8, T9);
- керування складовими системи (T4, T5, T6).

Після визначення задач системи, що проектується, доцільно створити блок-схему алгоритму системи. За допомогою блок-схеми в

подальшому можна визначити залежності між задачами, їх послідовність та час появи. Блок-схема алгоритму роботи адаптивного круїз-контролю представлена на рис. 3.

Як показано на рис. 3, система має набір послідовних задач, наприклад T1 і T2, T8 і T9. Ці задачі виконуються послідовно, одна за одною, а отже, додавання їх у чергу задач теж відбувається одночасно.

У системі також наявні задачі, що виконуються паралельно. Наприклад, задачі T1 та T7 не є взаємозалежними, вони виконуються паралельно і незалежно одна від одної. Виклик задач T1 та T7 провокується таймером, тобто ці задачі є періодичними.

Проте є задачі, які виконуються не завжди, а отже, є деяка умова їх виклику та додавання у чергу задач. Тобто, деякі задачі можуть виникати в процесі роботи алгоритму, залежно від поточного стану системи та середовища. Такими задачами є T3, T4, T5, T6, а також T8 і T9. Отже, задачі не завжди перебувають у черзі виконання задач, а можуть додаватися динамічно. Це означає, що деякі задачі мають залежність на результат виконання інших задач, а отже, вони додаються у чергу тільки після виконання попередніх задач. Наприклад, задача T3 потрапить у чергу задач після виконання задач T1 і T2 і за умови, що потрібна зміна швидкості авто.

Як видно з алгоритму, задачі T1, T2 та T7 є періодичними, тобто такими, що викликаються через рівні проміжки часу. Це потрібно для постійного моніторингу дорожньої ситуації. Період задач вважається наперед відомим, і залежить від вимог системи, що проектується. Інші задачі, як-от T3 і T8 є асинхронними, тобто виникають лише за певних умов. Відповідно, під час роботи системи можуть з'являтися нові задачі, які раніше не були в черзі планувальника задач. Тому на кожному такті роботи планувальника задач потрібно перевіряти чергу на появу нових задач і підлаштовувати порядок виконання задач відповідно до оновленої черги завдань.

Проаналізувавши алгоритм роботи системи, можна визначити часові характеристики, залежності між задачами, а також послідовність їх виконання. У подальшому отримані дані можна використати для моделювання роботи системи реального часу, щоб провести оцінювання часових характеристик задач шляхом аналізу даних, отриманих за допомогою моделювання розподілу процесорного часу між задачами, відповідно до обраного алгоритму планувальника задач. Зокрема, можна використати запропонований метод оцінювання часових характеристик за допомогою сіток Петрі [1].

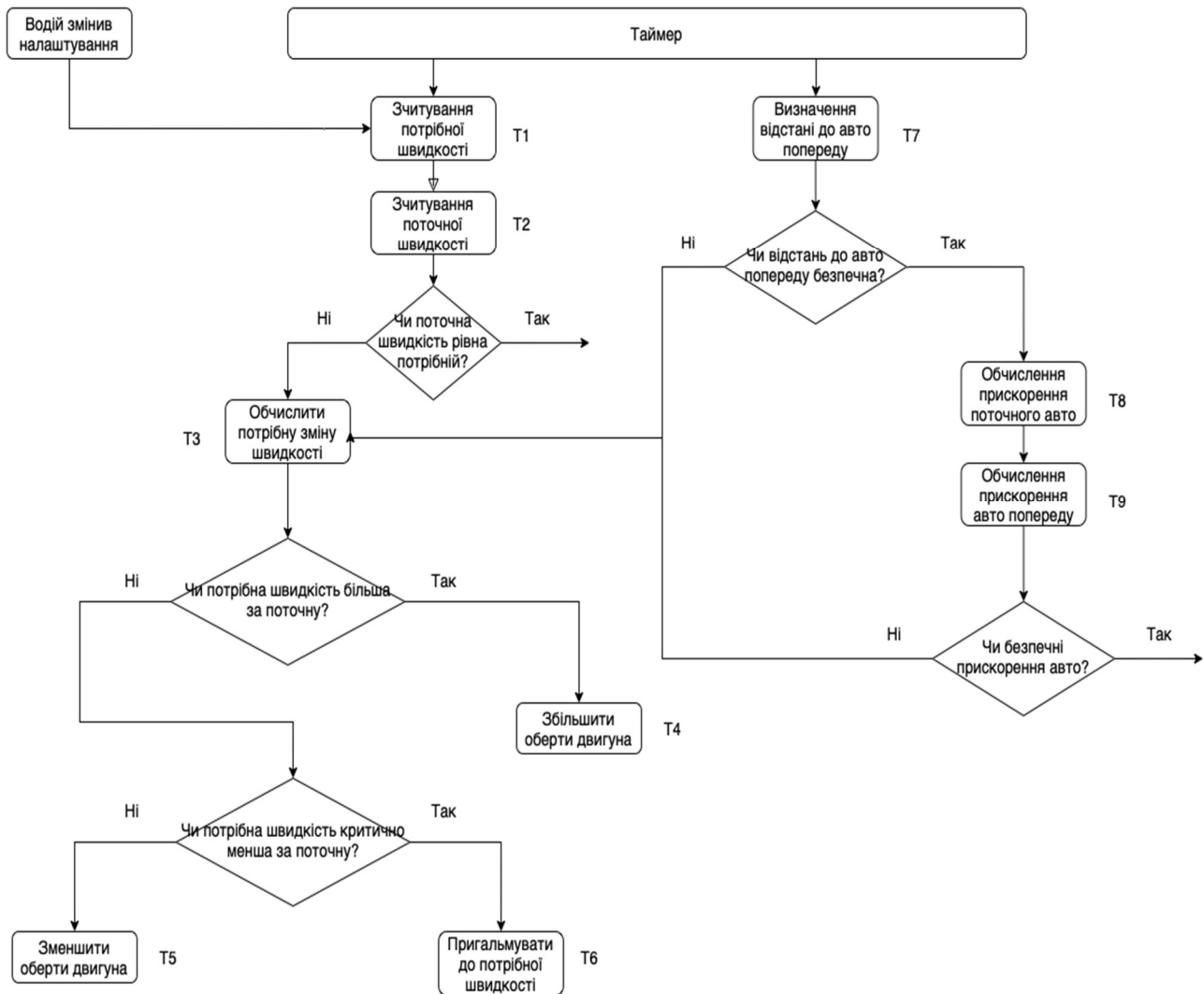


Рисунок 1 – Гібридна модель обробки даних для прогнозування й автоматизації просторових рішень у дизайні інтер'єру

### Висновок

Успішність роботи системи реального часу залежить не тільки від її логічної правильності, а й від часу, за який система видає результат. Проектуючи систему реального часу, важливим завданням є визначення часових характеристик, що використовуються для моделювання роботи системи. Для цього потрібно знати такі часові характеристики задач систем реального часу, як час появи задачі, час її виконання, а також кінцевий час виконання задачі. У цій роботі запропоновано метод для визначення вищенаведених часових характеристик. Суть методу полягає в тому, що потрібно розбити алгоритм роботи системи на задачі, зрозуміти їх послідовність та залежності. Після розбиття системи на задачі, пропонується створити блок-схему роботи

алгоритму, яка буде містити в собі задачі і їх послідовність. Після цього, використовуючи схему задач, можна наочно визначити часові характеристики задач. У подальшому визначені часові характеристики можуть бути використані для моделювання роботи системи відповідно до обраного алгоритму планувальника задач. Наприклад, одна задача може з'явитися тільки після того, як завершиться інша задача. Ця нова задача, час появи якої був до цього невідомий, отримує його відповідно до часу завершення попередньої задачі, тобто час завершення попередньої задачі стає часом появи нової задачі. Далі процедура роздачі часових квантів процесорного часу відбувається за звичайним порядком під управлінням планувальника, з урахуванням нової задачі.

## Список літератури

1. Tindell K. W. An extendible approach for analyzing fixed priority hard real-time tasks / K. W. Tindell, A. Burns, A. J. Wellings. *Real-Time Systems*. 1994. № 6. С. 133–151.
2. Motorola, Inc. MSCAN Block Guide [Електронний ресурс] / Motorola, Inc. 1998. URL: <https://www.rose-hulman.edu/class/ee/h Hoover/ece331/9S12C128%20PDF%20Documents/9S12%20Microcontroller%20Docs/S12MSCANV2.pdf>.
3. Punnekkat S. Response time analysis under errors for CAN / S. Punnekkat, H. Hansson, C. Norstrom. *IEEE Computer Society Press*. 2000. С. 258–265.
4. Nolte T. Minimizing CAN response-time analysis jitter by message manipulation / T. Nolte, H. Hansson, C. Norstrom. // *IEEE real-time and embedded technology and applications symposium*. – 2002. – С. 197–206.
5. Nolte T. Probabilistic worst-case response-time analysis for the Controller Area Network / T. Nolte, H. Hansson, C. Norstrom. *9th IEEE real-time and embedded technology and applications symposium*. 2003. С. 200–207.
6. Broster I. Probabilistic analysis of CAN with Faults / I. Broster, A. Burns, G. Rodríguez-Navas. *23rd IEEE real-time systems symposium*. 2002. С. 269–278.
7. Broster I. Timing analysis of real-time communication under electro- magnetic interference / I. Broster, A. Burns, G. Rodríguez-Navas. *Real-Time Systems*. 2005. № 30. С. 55–81.
8. Integrating reliability and timing analysis of CAN- based Systems / H. Hansson, T. Nolte, C. Norstrom, S. Punnekkat. *IEEE Transactions on Industrial Electronics*. 2002. № 49. С. 1240–1250.
9. Broster I. An analysable bus-guardian for event-triggered communication / I. Broster, A. Burns. *IEEE Computer Society Press*. 2003. № 24. С. 410–419.
10. Volcano – a revolution in on-board communica- tions [Електронний ресурс] / L. Casparsson, A. Rajnak, K. Tindell, P. Malmberg *Volvo Technology Report*. 1998. URL: <https://theeshadow.com/files/volvo/can/Casparsson-Volcano%20in%20Volvo%20Tech%20Report%201998.pdf>. Lukkien. *Springer Science + Business Media*. 2007. № 35. С. 239–272.
11. Audsley N. Optimal Priority Assignment And Feasibility Of Static Priority Tasks With Arbitrary Start Times [Електронний ресурс] / Audsley *Department of Computer Science, University of York*. 1991. URL: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=59911c0872b2e598ebc4fe8eefdb75d39f0e3ff01>.
12. Зайцев В. Г., Цибаєв Є. І. Оцінка часових характеристик у комп'ютерних системах реального часу з використанням сіток Петрі. *Управління розвитком складних систем*. 2023. № 54. С. 48–62.
13. Зайцев В. Г., Цибаєв Є. І. Модель оцінки часових характеристик у комп'ютерних системах реального часу з використанням сіток. *Управління розвитком складних систем*. 2019. № 40. С. 76 – 86; [dx.doi.org/10.6084/m9.figshare.11969013](https://doi.org/10.6084/m9.figshare.11969013).
14. Зайцев В. Г., Цибаєв Є. І. Оцінка часових характеристик задач в багатопроцесорних системах реального часу з використанням сіток Петрі. *Управління розвитком складних систем*. 2020. № 42. С. 43 – 50.

Стаття надійшла до редколегії 03.09.2024

**Yurovych Ivan**

Postgraduate of the Department of System Programming and Specialized Computer Systems, Kyiv Polytechnic Institute, <https://orcid.org/0009-0005-9575-065X>

National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv

**Zaitsev Vladimir**

DSc (Eng.), Professor, Professor of the Department of System Programming and Specialized Computer Systems, Kyiv Polytechnic Institute, <https://orcid.org/0009-0009-9917-5364>

National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv

**METHOD OF DETERMINING TIMING PARAMETERS OF REAL TIME SYSTEMS**

**Abstract.** *Real-time systems are widely used and actively implemented in advanced developments across many industries, from medicine to aerospace. Research and modelling of real-time systems are crucial tasks at the design stage, as they help in the determination of whether the system being modelled meets the specified timing characteristics and, accordingly, assess the system's ability to satisfy timing requirements. After all, the success of a real-time system depends not only on their logical correctness but also on the time it takes for the system to generate a result. Considering the various types of tasks, such as synchronous and asynchronous, parallel and sequential, determining their timing characteristics at the design stage is quite a complex problem. It is also essential to consider the sequence of tasks and their arrival times since there are cases where one task depends on the result of another task, and thus, the arrival time of a new task to the scheduler's queue depends on the completion time of the previous task. Previous studies have investigated methods for evaluating the timing characteristics of tasks in real-time systems by analyzing data obtained from modeling the distribution of processor time among tasks according to selected scheduler algorithms using Petri net models in both single-processor and multi-processor systems. These methods ensured the acquisition of task timing characteristics when choosing a specific type of processor and scheduler, which is necessary for the initial technical design of a real-time system. However, the methods have not considered the dynamic nature of task yielding, which is an essential component*

of modern real-time systems. This paper proposes a method for determining the timing characteristics of real-time systems at the design stage. The proposed method helps to identify the arrival time of tasks, taking into account various types of tasks and their dependencies. The identified timing characteristics can subsequently be used for modelling the system's operation and determining the optimal task scheduling algorithm.

**Keywords:** real time system (RTS); model; timing parameters; task scheduler; Petri net; adaptive cruise control

#### References

1. Tindell, K. W., Burns, A., Wellings, A. J. (1994). An extendible approach for analyzing fixed priority hard real-time tasks. *Real-Time Systems*, 6, 133–151.
2. Motorola, Inc. (1998). MSCAN Block Guide. URL: <https://www.rose-hulman.edu/class/ee/h Hoover/ece331/9S12C128%20PDF%20Documents/9S12%20Microcontroller%20Docs/S12MSCANV2.pdf>.
3. Punnekkat, S., Hansson, H., Norstrom, C. (2000). Response time analysis under errors for CAN. *IEEE Computer Society Press*, 258–265.
4. Nolte, T., Hansson, H., Norstrom, C. (2002). Minimizing CAN response-time analysis jitter by message manipulation. *IEEE real-time and embedded technology and applications symposium*, 197–206.
5. Nolte, T., Hansson, H., Norstrom, C. (2003). Probabilistic worst-case response-time analysis for the Controller Area Network. *9th IEEE real-time and embedded technology and applications symposium*, 200–207.
6. Broster, I., Burns, A., Rodríguez-Navas, G. (2002). Probabilistic analysis of CAN with Faults. *23rd IEEE real-time systems symposium*, PP. 269–278.
7. Broster, I., Burns, A., Rodríguez-Navas, G. (2005). Timing analysis of real-time communication under electromagnetic interference. *Real-Time Systems*, 30, 55–81.
8. Hansson, H., Nolte, T., Norstrom, C., Punnekkat, S. (2002). Integrating reliability and timing analysis of CAN-based Systems. *IEEE Transactions on Industrial Electronics*, 49, 1240–1250.
9. Broster, I., Burns, A. (2003). An analysable bus-guardian for event-triggered communication. *IEEE Computer Society Press*, 24, 410–419.
10. Casparsson, L., Rajnak, A., Tindell, K., Malmberg, P. (1998). Volcano – a revolution in on-board communications. URL: <https://theeshadow.com/files/volvo/can/Casparsson-Volcano%20in%20Volvo%20Tech%20Report%201998.pdf>.
11. Davis, R., Burns, A., Bril, R., Lukkien, J. (2007). Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised. *Springer Science + Business Media*, 35, 239–272.
12. Audsley, N. (1991). Optimal Priority Assignment And Feasibility Of Static Priority Tasks With Arbitrary Start Times. URL: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=59911c0872b2e598ebc4fe8eefdb75d39f0e3f01>.
13. Zaitsev, V. G., Tsybaev, E. I. (2023). Estimation of timing characteristics in real-time computer systems using petri nets. *Management of development of complex systems*, 40, 48–62.
14. Zaitsev, V. G., Tsybaev, E. I. (2019). A model for estimating time characteristics in real-time computer systems using Petri nets. *Management of development of complex systems*, 40, 76–86.
15. Zaitsev, V. G., Tsybaev, E. I. (2020). Evaluation of time characteristics of problems in multiprocessor real-time systems using petri networks. *Management of development of complex systems*, 42, 43–50.

#### Посилання на публікацію

APA Yurovych, I. & Zaitsev, V. (2024). Method of determining timing parameters of real time systems. *Management of Development of Complex Systems*, 59, 148–154, dx.doi.org/10.32347/2412-9933.2024.59.148-154.

ДСТУ Юрович І. В., Зайцев В. Г. Метод визначення часових характеристик систем реального часу. *Управління розвитком складних систем*. Київ, 2024. № 59. С. 148 – 154, dx.doi.org/10.32347/2412-9933.2024.59.148-154.