

УДК 629.783:004.27:621.3.051.025

doi: 10.15421/aktt.2025.6.12

І. Б. ТУРКІН, О. В. ЛЮБІМОВ, Л. О. ВОЛОБУЄВА

*Національний аерокосмічний університет**«Харківський авіаційний інститут», Харків, Україна*

МОДЕЛЬ РЕАКТИВНОСТІ ВБУДОВАНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ НАНОСУПУТНИКІВ І БПЛА

Вбудований обчислювач є ключовим елементом авіоники як космічних апаратів, так і безпілотних літальних апаратів (БПЛА). Його реактивність визначає здатність системи гарантувати, що час реакції на зовнішні події відповідає заданим обмеженням реального часу. **Об'єкт дослідження** – реактивність мікроконтролера вбудованої обчислювальної системи як складової загальної технічної ефективності апаратно-програмного комплексу. **Предметом дослідження** є математична модель реактивності, а також методи, інструменти й технологія експериментального дослідження реактивності операційної системи реального часу. **Мета роботи** – розробити та верифікувати модель реактивності програмного забезпечення вбудованого обчислювача для застосування в наносупутниках і БПЛА. У рамках дослідження обґрунтовано необхідність створення теоретико-множинної моделі реактивності вбудованих систем авіоники для малих космічних апаратів і безпілотних платформ, виконано аналіз обмежень (Worst-Case Execution Time, WCET) для типових бортових завдань обчислювача та сформульовано вимоги до моделі, розроблено її математичний опис і програмну реалізацію. Методами чисельного моделювання перевірено відповідність результатів симуляції фактичним процесам у системі. **Результати.** Запропонована математична модель реактивності враховує стохастичні характеристики потоків подій та двоетапну обробку переривань, що дає змогу оцінювати ймовірність своєчасної реакції системи на події різних типів. Експериментальна перевірка підтвердила коректність моделі та її придатність для аналізу платформи Борівітер / Falco під керуванням операційної системи FreeRTOS. **Наукова новизна** полягає в розробці нової концепції оптимізації роботи планувальника операційної системи реального часу бортового обчислювача БПЛА та наносупутників, яка заснована на комплексній математичній моделі енергетичних витрат, обчислювальної продуктивності та реактивності БО, що на відміну від механізмів керування енергоспоживанням у процесорах із динамічним регулюванням частоти або напруги, спеціально орієнтована на мікроконтролери БО, у яких частота та напруга фіксовані, а оптимізація можлива лише за рахунок правильного вибору політики керування енергозберігаючими режимами. **Висновки.** Розроблена модель може бути використана для оцінювання та підвищення реактивності обчислювальних систем як університетського наносупутника ХАІ-ІКА, так і безпілотних авіаційних систем. Подальші дослідження спрямовані на створення інструментального програмного забезпечення аналізу реактивності для підтримки інженерії програмного забезпечення нових поколінь аерокосмічних і безпілотних платформ високого рівня автономності з урахування впровадження засобів AI/ML.

Ключові слова: наносупутник; БПЛА; бортовий обчислювач (БО); програмне забезпечення; реактивність; модель реактивності.

1. Вступ

Бортовий обчислювач є вбудованою системою, що відповідає за моніторинг, управління і контроль та являє собою одну з найважливіших складових авіоники як наносупутників так і безпілотних платформ.

Сучасні безпілотні літальні апарати здатні виконувати завдання високої складності комерційного, наукового, військового та промислового характеру з різним рівнем автономності. За даними Drones Market Report (MarketResearch.com) глобальний ринок UAVs (Fixed Wing, Rotary Wing, Hybrid) у 2024

році досяг \$30,7 млрд та за прогнозом на 2033 має досягти \$74,8 млрд з CARG у 10,41%. Для управління польотами БПЛА використовуються різні методи: дистанційно керовані (remotely piloted), напівавтономні (semi-autonomous) та повністю автономні (fully autonomous) [1]. Тоді як дистанційно контрольовані дрони потребують постійного зв'язку з наземною станцією, яка контролює курс, швидкість та висоту, напівавтономні та автономні дрони вказані параметри контролюють самостійно з використанням вбудованих сенсорів поєднуючи бортові обчислення та зв'язок у режимі реального часу (РЧ).



[Creative Commons Attribution
NonCommercial 4.0 International](https://creativecommons.org/licenses/by-nc/4.0/)

За даними дослідження Drone manufacturing market (www.futuredatastats.com) ключові тенденції ринку виробництва дронів включають інтеграцію засобів АІ, автоматизацію розширених можливостей та розробку гібридних дронів. Так, на теперішній час для підтримки вимог щодо автономності БПЛА рівня 4 та вище програмне забезпечення для польотів часто використовує гібридну архітектуру, що дозволяє поєднати можливості, які надають сучасні досягнення АІ з попередніми напрацюваннями в галузі.

Наносупутники – це клас малих космічних апаратів (КА) масою від 1 до 10 кг, які набули широкого поширення завдяки низькій вартості розробки та запуску. Найпоширенішим стандартом наносупутників є CubeSat [2, 3] – модульні платформи з базовою одиницею $10 \times 10 \times 10$ см (1U) вагою до 1,33 кг. Наносупутники використовуються для дистанційного зондування Землі, технологічних демонстрацій, наукових досліджень та освітніх місій. Їх перевагами є короткий цикл розробки (1-3 роки), можливість групового запуску та використання комерційних компонентів (Commercial-off-the-shelf, COTS [4]). Національний аерокосмічний університет «Харківський авіаційний інститут» розробляє власний наносупутник ХАІ-ІКА стандарту CubeSat для відпрацювання технологій побудови супутників, розробки програмного забезпечення (ПЗ) для бортових обчислювачів та розробки антени дистанційного зондування Землі як корисного навантаження. Додатковою метою проекту зі створення ХАІ-ІКА є розбудова освітніх курсів щодо розробки бортових систем керування космічних апаратів.

У статті розглядається реактивність бортових обчислювачів для безпілотних платформ та космічних апаратів (КА), як одна з найважливіших характеристик керуючих систем авіоники. Реактивність є критично важливою характеристикою для авіоники наносупутників та БПЛА, оскільки вони працюють в умовах обмежених обчислювальних ресурсів та енергоспоживання, де для досягнення ефективної роботи є потреба економії енергії шляхом знеструмлення або переведення в специфічні режими енергозбереження та «сну» бортової авіоники. У той самий час існує необхідність швидкої реакції системи на зовнішні події, такі як команди з наземної станції або зміни телеметричних параметрів, що безпосередньо впливає на ефективність виконання місії. У наносупутниках та БПЛА затримки в обробці даних можуть призвести до втрати всього літального апарата.

Системи реального часу з предиктивною поведінкою забезпечують детерміновані часові характеристики, що особливо важливо для задач орієнтації та стабілізації КА та БПЛА. Використання операційних систем реального часу (ОСРЧ) дозволяє досягти

необхідного рівня реактивності при мінімальних апаратних витратах [5]. Будь-які обчислення, від корисних, ті що виконують бізнес-завдання об'єкта керування (наприклад керування двигуном), до системних, як то діагностування стану БО та підключених сенсорів і виконуючих механізмів, виконуються програмним забезпеченням на центральному процесорі (ЦП) або мікроконтролері (МК).

Реактивність визначається як здатність БО своєчасно реагувати на зовнішні події з урахуванням жорстких обмежень на затримку (latency) та варіацію затримки (jitter), що накладаються специфікою апаратної платформи та завдань місії. Завдання місії включають і звичайне функціональне виконання (наприклад, у циклах систем управління орієнтацією, циклах формування телеметричних кадрів, обробки команд тощо), і аварійне реагування на відмови (FDIR – Fault Detection, Isolation and Recovery).

Таким чином часові характеристики ПЗ БО та інших засобів авіоники визначають не лише ефективність управління, але й життєздатність місії. Несвоєчасне виконання критичних функцій може призвести до втрати керування, що може бути як тимчасовою так і постійною. До типових проблем від втрат керування можна віднести наступні випадки:

- порушення замкнених контурів керування (наприклад, у системах орієнтації (ADCS – Attitude Determination and Control System), що викликає втрату точності орієнтації та зростання енергоспоживання на корекції;
- невчасне виявлення відмов в енергетичній підсистемі (Electrical Power System, EPS), що може спричинити глибокий розряд акумуляторів або тепловій аварії;
- порушення часової синхронізації між підсистемами, що призводить до помилок у телеметрії, некоректної роботи алгоритмів навігації та погіршення керованості;
- втрати даних корисного навантаження через недостатньо швидке зчитування, буферизацію чи обробку.

Усі ці фактори підкреслюють, що реактивність ПЗ БО має бути формалізована та гарантована системним дизайном ПЗ та апаратного забезпечення (АЗ) засобів авіоники КА і БПЛА у вигляді кількісних вимог та підтверджених властивостей й перевірена у процесі верифікації.

1.1. Мотивація і сучасний стан

Автори роботи [6] навели таксономію польотного програмного забезпечення (Flight Software, FSW) CubeSat місій та відмітили загальне зростання дослідницької зацікавленості щодо пов'язаних з роз-

робкою FSW CubeSat питань, дослідивши публікації в активності по кожній виділеній категорії у період з 2006 по 2024 роки. В роботі [6] виконано комплексний аналіз специфікацій та проблем реалізації FSW CubeSat з урахуванням апаратних обмежень.

Існує головне технічне протиріччя – зменшення енергетичних витрат погіршує обчислювальну продуктивність та реактивність системи. Зокрема, споживання енергії зростає квадратично зі збільшенням напруги живлення, що робить цей параметр найважливішим для оцінки енергетичних витрат, але відхилення напруги від рекомендованого виробником мікроконтролера БО значення погіршує показники надійності. Підвищення тактової частоти МК прискорює обчислення (збільшує обчислювальну продуктивність) та покращує реактивність, але водночас збільшує споживання енергії. Збільшення частоти системного таймера ОСРЧ (частоти системного тикуну) – це одночасно і збільшення енергетичних витрат на службові операції операційної системи, і покращення реактивності. Використання режимів енергозбереження МК суттєво знижує споживання енергії, але помітно погіршує продуктивність та збільшує час реакції (погіршує реактивність) через затримки при відновленні активної роботи МК.

Розроблення комплексної математичної моделі споживання енергії, обчислювальної продуктивності та реактивності БО має вирішальне значення для ефективності всієї місії як для наносупутників, так і у разі використання безпілотних платформ.

По-перше, така модель дозволяє точно оцінити баланс між обмеженими енергетичними ресурсами та обчислювальними потребами, що є критично важливим в умовах жорстких енергетичних обмежень. По-друге, вона створює теоретичну основу для оптимізації розподілу обчислювальних навантажень, забезпечуючи стабільну роботу системи за різних режимів. По-третє, модель дає змогу аналізувати реактивність системи, гарантувавши здатність бортового комп'ютера своєчасно реагувати на події, важливі для місії. Модель також підтримує процеси планування роботи всієї авіоники КА або БПЛА, надаючи прогностичні дані про використання ресурсів, що покращує планування розкладу комунікаційних сеансів та роботи корисного навантаження.

Однією з ключових метрик, що описують реактивність, є найгірший гарантований час виконання (Worst-Case Execution Time, WCET) [7, 8]. Це максимальний час виконання завдання у найгіршому сценарії з урахуванням усіх можливих затримок. Для бортових систем наносупутників та БПЛА, WCET критичних функцій має бути істотно меншим за період відповідного розрахункового системного обчислювального циклу, аби забезпечити часовий запас для непередбачених асинхронних системних переривань та

варіацій навантаження.

Аналіз публікацій та досвіду реалізації ПЗ у місіях типу CubeSat [6, 9, 10] показує, що орієнтовні обмеження WCET для типових бортових завдань БО знаходяться у діапазоні:

- зчитування сенсорних даних ADCS не більше, ніж 0.5 мс з періодичністю 1–10 мс;
- обчислення закону керування орієнтацією не більше, ніж 0.75 мс з періодичністю 5–20 мс;
- передача команд виконавчим пристроям не більше, ніж 0.15 мс при затримці ≤ 1 мс після розрахунку;
- агрегація телеметрії не більше, ніж 2 мс з періодичністю 0.1–1 с.

Такі межі забезпечують виконання критичних задач без пропуску часових обмежень (дедлайнів) і з достатнім запасом для функцій діагностики та FDIR.

Також, надважливим чинником що впливає на вимоги до реактивності, є здатність системи своєчасно реагувати на події що виявлені під час FMEA/FMEDA/FMECA [10, 11] аналізу, та позначені як критичні, оскільки призводять до невідновлюваного аварійного стану. Аналіз видів та наслідків відмов (FMEA), його розширені версії з урахуванням діагностичного покриття (FMEDA) та критичності (FMECA - Failure mode effects and criticality analysis) широко застосовуються у проектуванні космічних апаратів та БПЛА для формування вимог до FDIR.

Досвід таких місій, як Masat-1 [12] та Foresail-1p [13], показує, що підхід з аналізу FMECA дозволяє для кожного сценарію відмови визначити час до небезпечного ефекту (time-to-hazard) та сформулювати гранично допустимі інтервали реакції:

- час виявлення (T_{detect}) не повинен перевищувати $\frac{1}{4}$ часу time-to-hazard;
- час ізоляції (T_{isolate}) не повинен перевищувати половину часу time-to-hazard;
- час відновлення (T_{recover}) не повинен перевищувати час time-to-hazard.

Приклад: для відмови у вигляді перенавантаження шини живлення енергосистеми, коли time-to-hazard $\approx 0.2\text{--}0.5$ с, дедлайни будуть:

- $T_{\text{detect}} \leq 50$ мс;
- $T_{\text{isolate}} \leq 100$ мс;
- $T_{\text{recover}} \leq 500$ мс.

Аналогічно, для неконтрольованого розгону реакційного колеса (складова системи орієнтації) з часом до втрати керованості у сотні мілісекунд необхідно забезпечити виявлення за 1–2 цикли ADCS (10–20 мс) та ізоляцію ≤ 100 мс.

1.2. Мета та завдання

Мета роботи. Розробити та практично верифікувати модель реактивності вбудованого обчислю-

вача для застосування в наносупутниках і безпілотних платформах на прикладі авторського бортового обчислювача Борівітер 0.1/Falco [14] (рис. 1).

На етапі проектування модель допомагає обмежити простір параметрів майбутньої платформи, дозволяючи інженерам заздалегідь оцінити енергоспоживання, продуктивність та інші характеристики. Під час виконання місії її використання відкриває можливості для адаптивного керування режимами в реальному часі завдяки порівнянню фактичних показників із закладеними у модель. У перспективі така модель сприяє підвищенню відмовостійкості та масштабованості системи, що важливо для розвитку наступних поколінь наносупутників та безпілотних платформ.

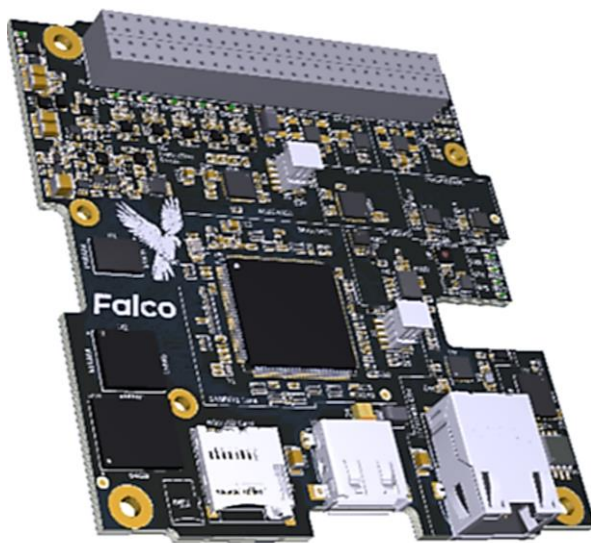


Рис. 1. Авторський бортовий обчислювач – Борівітер 0.1/Falco [14]

Таким чином, модель є універсальним інструментом, що поєднує аналіз зовнішніх впливів, апаратних конфігурацій, програмних налаштувань і динамічних політик управління роботою мікроконтролера.

Для досягнення поставленої мети в межах дослідження необхідно виконати наступні завдання:

- виконати аналіз технічних вимог і практичного досвіду та сформулювати обмеження для типових завдань бортового обчислювача у складі авіоніки для малих космічних апаратів і безпілотних платформ (розділ 2);
- визначити комплексну модель цільових показників ефективності БО (розділ 3);
- побудувати модель реактивності програмного забезпечення бортових обчислювачів наносупутників та БПЛА (розділ 4);
- виконати чисельний експеримент для перевірки запропонованої моделі реактивності (розділ 5).

2. Формування обмежень для типових завдань БО у складі авіоніки для малих космічних апаратів і безпілотних платформ

WCET визначає максимальну тривалість виконання окремої задачі, тоді як реактивність в FDIR описує часові рамки послідовності "виявлення–ізоляція–відновлення". Для забезпечення відповідності FDIR-вимогам, WCET критичних задач діагностики та обробки відмов має бути кратно меншим за часові обмеження (дедлайн) відповідної фази реагування.

Наприклад, якщо T_{detect} для перегріву акумулятора встановлено на рівні 1 с, то задачі моніторингу температури повинні мати WCET не більше десятків мілісекунд, аби зберегти стабільний період опитування та своєчасність реагування.

Аналіз технічних вимог і практичного досвіду (детально розглянуто у розділі 1.1) дозволяє окреслити такі орієнтовні межі реактивності для наносупутникових місій що є також релевантними для місій на основі використання БПЛА:

- WCET критичних функцій керування: 0.5–1 мс при циклах 5–20 мс;
- виявлення критичних відмов в електроживленні та динамічних підсистемах: 10–100 мс;
- реакція на термічні відмови: 1–5 с (залежно від теплової інерції вузла);
- точність абсолютного часу: десятки мікросекунд (наприклад, ± 35 мкс у місії Hitomi [15, 16]).

Ці значення не є універсальними, але відображають реалістичні проєктні цілі для більшості наносупутників класу CubeSat та БПЛА, що працюють на платформах із обмеженими ресурсами та під керуванням ОСРЧ.

Таким чином, реактивність ПЗ бортових обчислювальних засобів наносупутників та БПЛА є критичним фактором, який безпосередньо впливає на:

- стабільність та точність замкнених контурів керування;
- надійність виявлення та усунення відмов;
- безпеку та життєздатність місії в цілому.

Формалізація та верифікація вимог до WCET та FDIR-реактивності на основі FMECA забезпечує не лише технічну відповідність стандартам ECSS (European Cooperation for Space Standardization - Європейська кооперація зі стандартизації в галузі космічної техніки), але й дає обґрунтовану впевненість у здатності системи працювати в умовах обмежених ресурсів та непередбачуваних зовнішніх впливів. Таким чином, розгляд реактивності ПЗ є невід'ємним елементом наукового та інженерного підходу до розробки бортових керуючих систем як наносупутників так і безпілотних платформ.

3. Комплексна модель цільових показників ефективності бортового обчислювача

Модель призначена для розрахунку залежностей цільових показників ефективності бортового обчислювача від налаштувань апаратно-програмної частини та характеристик завдань, що вирішує обчислювач. Узагальнено в нотації теорії множин комплексна математична модель споживання енергії (Energy), обчислювальної продуктивності (Productivity) та реактивності (Reactivity) бортового обчислювача – це система відображень:

$$\begin{cases} \text{Reactivity} : \{F_{\text{clk}}, F_{\text{timer}}\} \times \text{Modes} \times \text{Tasks} \rightarrow R; \\ \text{Energy} : \{F_{\text{clk}}, F_{\text{timer}}, V_{\text{dd}}\} \times \text{Modes} \times \text{Tasks} \rightarrow E; \\ \text{Productivity} : \{F_{\text{clk}}, F_{\text{timer}}\} \times \text{Modes} \times \text{Tasks} \rightarrow P, \end{cases} \quad (1)$$

де: $\{F_{\text{clk}}, F_{\text{timer}}, V_{\text{dd}}\}$ – множина системних параметрів, що характеризує налаштування апаратної частини: F_{clk} – тактова частота, V_{dd} – напруга

живлення, F_{timer} – частота переривань від системного таймеру;

Modes – множина можливих політик та відповідних програмних реалізацій керування режимами енергозбереження;

$\text{Tasks} = \{E^H, E^S, E^{bw}\}$ – множина завдань, яка містить опис вирішуваних системою завдань трьох видів: E^H – обробка подій жорсткого РЧ, E^S – обробка подій м'якого РЧ, E^{bw} – фонові завдання;

R – множина показників реактивності програмного забезпечення бортових обчислювачів;

E – середня споживана електрична потужність;

P – обчислювальна продуктивність, представлена як кількість виконуваних корисних операцій в одиницю часу.

Мікроконтролер бортового обчислювача може знаходитись в одному з доступних режимів:

$$\text{Modes} = \{\text{Active}, \text{Wait}, \text{Sleep}, \text{Backup}\}. \quad (2)$$

Діаграма станів скінченного автомата МК показана на рис. 2.

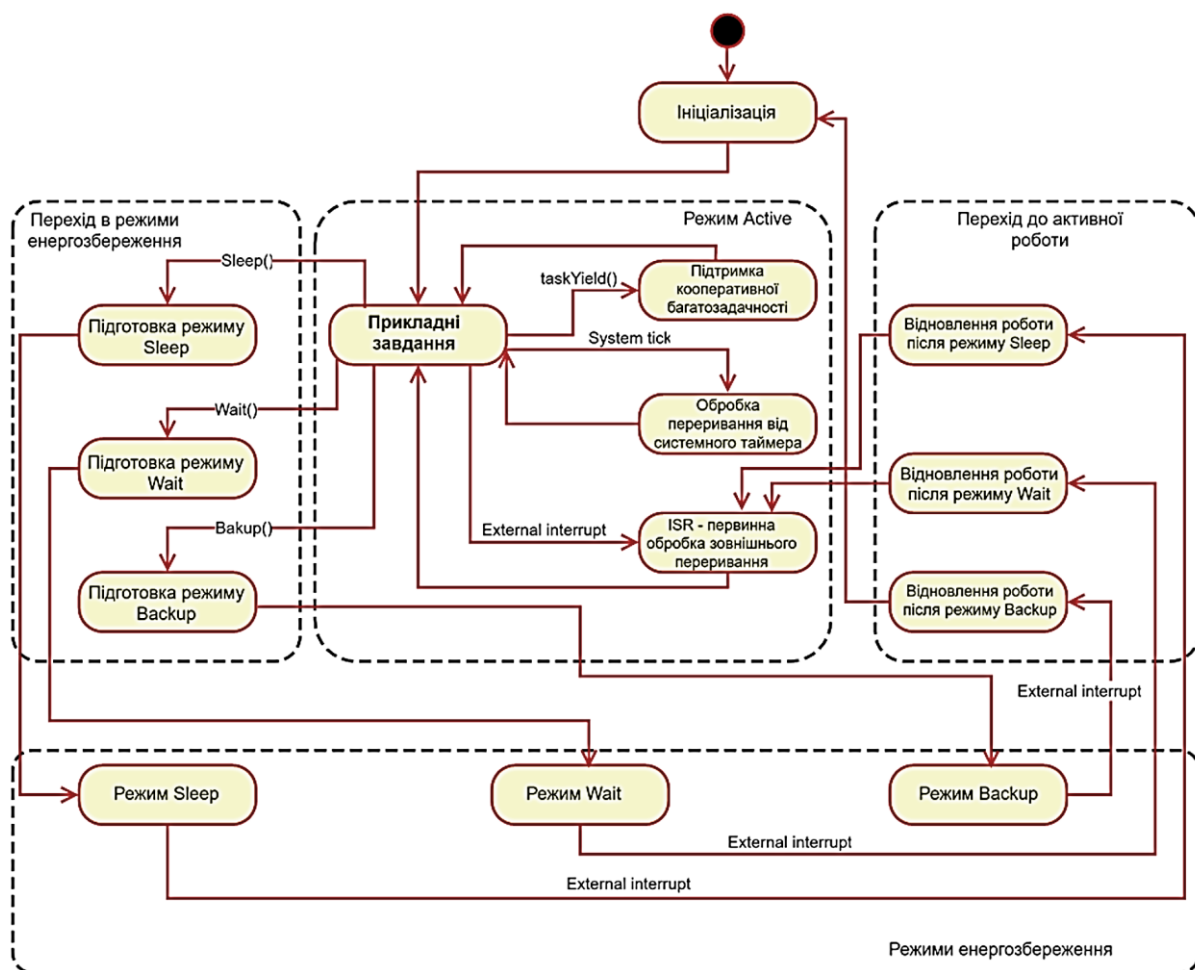


Рис. 2. Діаграма станів (state diagram) скінченного автомата МК

В активному режимі (Active) мікроконтролер працює на повній тактовій частоті, виконує всі обчислення, працює з периферією (UART, SPI, ADC тощо). Працюють периферійні модулі. Споживання енергії – номінальне, найвище серед усіх режимів.

В режимі очікування (Wait) модуль процесорного ядра (Core) зупинений, але периферійні модулі продовжують роботу (наприклад, таймери, UART). Процесор очікує надходження подій від периферії (переривання від таймера, завершення передачі даних, прихід сигналу по комунікаційному інтерфейсу тощо). Споживання енергії – нижче, ніж у Active, оскільки Core не виконує інструкцій.

В режимі сну (Sleep) зупиняється тактування більшості блоків МК, Core і більшість периферійних модулів вимикаються. Деякі низькоспоживні модулі (такі як модуль годинника реального часу (Real-Time Clock, RTC) або сторожовий таймер – watchdog) можуть залишатися активними. Вихід із режиму – через зовнішнє чи внутрішнє переривання. Споживання енергії – суттєво нижче, ніж у Wait, оскільки зупинено більшість периферійних модулів.

В режимі глибокого сну (Backup) майже вся система вимкнена, живляться тільки мінімально необхідні блоки, зазвичай це – RTC, регістри процесорного ядра та логіка пробудження. Режим застосовується для дуже тривалого очікування подій, при цьому зберігаються лише критично важливі дані і можливість пробудження по зовнішньому сигналу чи таймеру RTC (наприклад, датчики з автономним

живленням). Споживання енергії – найнижче з можливих.

4 Модель реактивності програмного забезпечення бортових обчислювачів наносупутників та БПЛА

Реактивність бортового обчислювача – це складова загальної ефективності апаратно-програмного забезпечення БО, що характеризує середню або гарантовану частку подій, на які система встигає відреагувати за одиницю часу, у відношенні до сумарної інтенсивності надходження подій. Надалі будемо розрізняти три типи процесів, які визначають реактивність, як складову загальної ефективності апаратно-програмного забезпечення бортового обчислювача (рис. 3):

- фонові процеси;
- процеси м'якого реального часу;
- процеси жорсткого реального часу.

Процеси м'якого реального часу (soft real-time) та жорсткого реального часу (hard real-time) ініціюються зовнішніми подіями. Події потрапляють в систему для подальшої обробки через механізм переривань. Переривання – це системний сигнал, що може використовуватись як засіб комунікації між апаратними складовими (як зовнішніми, так і вбудованими у МК) та програмними складовими.

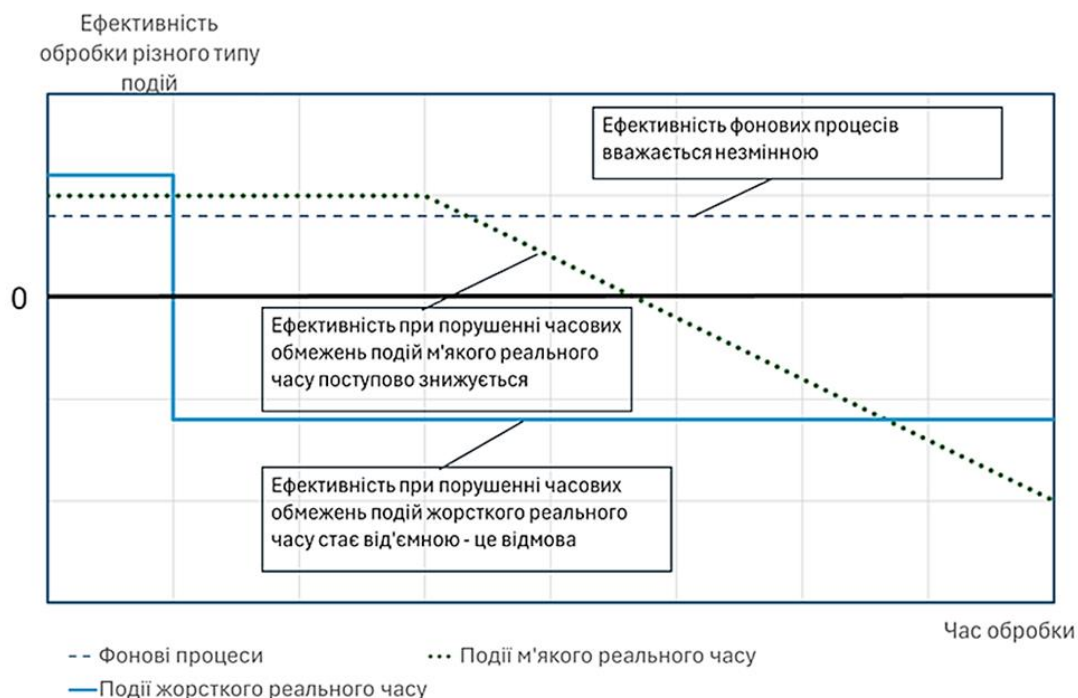


Рис. 3. Змістовна інтерпретація реактивності бортового обчислювача як складової загальної ефективності апаратно-програмного забезпечення БО

Надалі в статті розглянуто варіант, в якому обробка переривання (події) здійснюється в 2 етапи: первинним та вторинним обробниками.

Первинна обробка переривання здійснюється операційною системою в «процедурі обробника переривання» (Interrupt Service Routine, ISR) та полягає в активації заздалегідь створеного та ініціалізованого об'єкта ядра ОС, який дозволяє роботу вторинного обробника переривання, наприклад формування системної події `xTaskNotifyFromISR()` у ОСРЧ FreeRTOS. При цьому операційна система бере на себе відповідальність за такі дії, як переключення контексту з поточної функції до вторинного обробника переривання з урахуванням пріоритету вторинного обробника.

Вторинний обробник переривання – це користувацька «бізнес» функція, що починає свою роботу після отримання сигналу (перехід об'єкта ядра ОС в сигнальний стан) про виникнення конкретного переривання у системі. Час реакції системи на переривання надалі визначається як інтервал часу між створенням апаратною частиною сигналу переривання та завершенням обробки цього сигналу відповідним вторинним обробником.

При обробці подій жорсткого реального часу реактивність – це максимальна кількість зовнішніх подій, що надходять у вигляді переривань на одиницю часу, на які система здатна гарантовано відреагувати так, що час реакції на кожну подію не перевищує визначений дедлайн (часові обмеження), враховуючи фіксовані та змінні часові витрати на обробку події (переривання). Якщо мова йде про обробку подій м'якого реального часу [18], то реактивність БО – це характеристика здатності системи обробляти зовнішні події, визначена через середню кількість подій, які система гарантовано може обробити за одиницю часу, враховуючи затримки первинної та вторинної обробки, накладні витрати ОС і обмеження планувальника. Зазвичай бортовий обчислювач забезпечує обробку обох класів зовнішніх подій. Крім завдань обробки зовнішніх подій БО витрачає процесорний час на фонові процеси – background work, які не мають обмежень на час їх виконання.

Надалі дотримуємось наступного визначення: реактивність бортового обчислювача – це залежність ймовірності коректної обробки зовнішніх подій м'якого реального часу, яка визначає частку подій, оброблених за одиницю часу, із врахуванням стохастичних характеристик потоку подій, часу обслуговування та накладних витрат на обробку. Гарантована обробка подій жорсткого реального часу в цій постановці завдання є обмеженням – тобто кожна така подія має бути обов'язково оброблена.

4.1. Обмеження та припущення

Для формалізації та синтезу моделі реактивності БО під керуванням ОСРЧ, розглянемо припущення та обмеження що викладені нижче.

1. Реактивність системи залежить від тактової частоти процесора F_{clk} та характеристик потоку зовнішніх подій $E = \{E_i\}$.

2. Потік подій розбито на 2 класи: жорсткого – E^H та м'якого реального часу – E^S :

$$E = E^H \cup E^S, E^H \cap E^S = \emptyset. \quad (3)$$

3. Кожен тип події представлено кортежем, який залежить від типу події:

$$\forall E_i \in E^H, E_i = \langle f_i^{\max}, T_i, D_i \rangle, \quad (4)$$

$$\forall E_i \in E^S, E_i = \langle \lambda_i, T_i, D_i \rangle, \quad (5)$$

де: $f_i^{\max}$ – максимальна інтенсивність виникнення події жорсткого реального часу (подій/с);

λ_i – середня частота виникнення події м'якого реального часу (подій/с);

T_i – максимальний час вторинної обробки WCET, с;

D_i – обмеження на час обробки події (дедлайн), с.

4. Інтервали між подіями м'якого реального часу статистично моделюються як експоненційні (пуассонівський потік для кожного класу подій).

5. Обробка переривання (події) здійснюється в 2 етапи: первинним (перший етап) та вторинним (другий етап) обробниками.

6. Первинний обробник має найбільший пріоритет, його робота не переривається наступною подією, або примусово забороняється на рівні так званої «заборони вкладених переривань».

7. Час роботи первинного обробника є фіксованим для всіх типів подій та залежить від тактової частоти процесора F_{clk} :

$$T_{IRQ} = C_{IRQ} / F_{clk}, \quad (6)$$

де C_{IRQ} – кількість тактів процесора, необхідних первинному обробнику.

8. Первинний обробник ініціює роботу вторинного обробника через механізми операційної системи (об'єкти ядра). Вторинні обробники виконуються в одиночному сервері (single server), відповідно до механізму планування – EDF (Earliest Deadline First)

[19], допускається витиснення (preemption) вторинного обробника.

9. Час роботи вторинного обробника залежить від кількості тактів процесорного ядра МК, необхідних вторинному обробнику C_i , та тактової частоти процесорного ядра F_{clk} :

$$T_i = C_i / F_{clk} . \quad (7)$$

10. Буфер подій вважається необмеженим.

11. Фонова завантаженість процесора (background work) – це обчислювальна робота, яку МК виконує за умови відсутності необроблених подій. Вважаємо, що максимальний об'єм цієї роботи є відомим, МК її повинен виконати без обмежень реального часу, але при цьому знижується середній час, що залишається для обробки подій м'якого РЧ на наступну величину:

$$\Delta T_{bw} = C_{bw} / F_{clk} , \quad (8)$$

де C_{bw} – кількість тактів процесора, необхідних мікроконтролеру для відпрацювання фонові роботи.

12. Накладні витрати часу ОС T_{Core} на виконання таких операцій як переключення контексту, роботу з об'єктами ядра є параметрами моделі, залежать від кількості тактів процесора C_{Core} , необхідних для реалізації таких функцій та тактової частоти процесорного ядра МК F_{clk} :

$$T_{Core} = C_{Core} / F_{clk} . \quad (9)$$

13. До обмежень моделі також відноситься те, що вона не розглядає затримки, які можуть навмисно формуватись програмним забезпеченням, щоби оптимізувати споживану енергію [17]. До таких затримок відносяться:

– затримки прокрастинації (відкладання на потім – procrastination delays): алгоритми навмисно відкладають виконання деяких завдань, щоби збільшити тривалість періоду простою та ефективніше використовувати режими низького енергоспоживання. Ці затримки ретельно розраховуються, щоби не порушувати часові обмеження завдань;

– затримки планування: час, необхідний для прийняття рішень про зміну режиму енергоспоживання та перепланування завдань. Затримки планування, пов'язані з обчисленням оптимальних моментів для переходу в режим сну та пробудження: певні алгоритми виконують складні обчислення для визначення цих моментів, що може призвести до додаткових затримок;

– затримки передчасного завершення: ці затримки створюють додатковий простір для енергозбереження, але вимагають динамічного перепланування.

4.2. Функціональні залежності моделі

Коефіцієнт завантаження процесорного ядра МК (коефіцієнт утилізації) обробниками переривань залежить від сумарної інтенсивності вхідного потоку подій двох класів: жорсткого та м'якого РЧ:

$$U = \sum_{\forall E_i \in E^H} f_{E_i}^{\max} \cdot (T_{IRQ} + T_{E_i} + T_{Core}) + \sum_{\forall E_j \in E^S} \lambda_{E_j} \cdot (T_{IRQ} + T_{E_j} + T_{Core}) . \quad (10)$$

Для гарантованої обробки потоку подій жорсткого РЧ E^H необхідно, щоби була виконана робота всіх первинних обробників переривань з урахуванням накладних витрат часу на роботу ОС T_{Core} . Наступне обмеження сформульовано в припущенні, що вторинна обробка подій м'якого РЧ може бути відкладена на певний час:

$$U - \sum_{\forall E_j \in E^S} \lambda_{E_j} \cdot T_{E_j} \leq 1 . \quad (11)$$

Час реакції на подію типу i буде визначатись двома складовими: часом її обробки та часом перебування події в черзі:

$$T_{resp,i} = W_i + (T_{IRQ} + T_i + T_{Core}) , \quad (12)$$

де W_i – час очікування у черзі для події типу i (секунди) що залежить від режиму планування та стану системи в момент приходу події.

Модель Поллачека–Хінчина (Pollaczek–Khinchine) [20], яка є фундаментальною в теорії черг, дає можливість розрахунку середнього часу очікування:

$$E[W] = \frac{\Lambda \cdot E[S^2]}{2 \cdot (1 - \rho)} , \quad (13)$$

де: $E[W]$ – середній час очікування у черзі (секунди), усереднений за всіма класами;

Λ – інтенсивність подій;

$E[S^2] = \text{Var}[S] + (E[S])^2$ – математичне очікування квадрату часу очікування у черзі (c^2);

$\rho = \Lambda \cdot E[S]$ – завантаження процесорного ядра МК.

Ймовірність того, що подія типу i буде повністю оброблена до встановленого дедлайну (ймовірність своєчасної обробки події):

$$P_{\text{Success}, i} = \Pr(T_{\text{resp}, i} \leq D_i), \quad (14)$$

де D_i – обмеження на час обробки події (дедлайн).

Відповідно до встановлених припущень та обмежень, ймовірність неприйнятної затримки для події типу i :

$$P_{\text{loss}, i} = 1 - P_{\text{Success}, i}. \quad (15)$$

Модель Поллачека–Хінчина дає змогу оцінювати середні затримки та час очікування при довільному розподілі часу обслуговування, знаючи лише його середнє та дисперсію, але ця модель непридатна в ситуації коли вхідні події поділяються на два класи: жорсткого та м'якого РЧ та обробка цих подій здійснюється одним сервером з використанням трьох черг, що мають впорядковані пріоритети (від найбільшого до найменшого):

- черга первинного обробника переривань;
- черга вторинного обробника переривань (подій) жорсткого реального часу;
- черга вторинного обробника переривань (подій) м'якого реального часу.

Замість аналітичного рішення представлена в статті модель реактивності формально описує здатність обчислювача обробляти вхідний потік подій двох класів у межах заданих часових обмежень. Модель допускає два режими аналізу: детерміністичний (аналіз за найгіршим випадком, WCET) і стохастичний (модель черги). Для практичного використання розробленої моделі необхідно експериментально визначити наступні параметри:

- час обробки первинного переривання T_{IRQ} , мкс. Він є фіксованим для всіх переривань в системі;
- накладні витрати T_{Core} на реалізацію таких операцій ОС як переключення контексту та роботу з об'єктами ядра.

5. Результати експериментів та обговорення

Для експериментального визначення необхідних параметрів моделі, що були задекларовані в попередньому розділі, був складений експериментальний стенд, розроблене авторське експериментальне

ПЗ, проведено експеримент. Експеримент був багатоцільовим, в ньому крім витрат часу на певні операції а активному режимі роботи МК досліджувались також залежності енергоспоживання в різних режимах від частоти та час, необхідний МК для переходу між різними режимами. Докладний опис експерименту та його результатів можна знайти в статті [17].

Для перевірки запропонованої моделі реактивності виконано чисельний експеримент, в ході якого імітувалась ситуація, коли в МК поступає два потоки подій (переривань) з детерміністичними часами обслуговування:

- пуассонівський потік подій (переривань) м'якого реального часу з середньою інтенсивністю λ_i ;
- потік подій (переривань) жорсткого реального часу, частота яких фіксована $f_{E_i}^{\text{max}}$, що відповідає концепції WCET.

Як вхідні дані для чисельного експерименту використано результати натурного експерименту [17], що зведено в Таблицю 1.

Таблиця 1

Залежність витрат часу на службові операції під управлінням FreeRTOS на апаратній платформі «Борівітер-0.1» від тактової частоти МК

Тактова частота МК, МГц	Витрати часу на роботу первинного обробника переривань T_{IRQ} , мкс	Витрати часу на роботу з об'єктом ядра, T_{Core} , мкс
30	6,42	7,73
100	1,95	2,34
300	0,65	0,78

Завдяки специфіці обчислювальної архітектури ARM Cortex-M що є архітектурою МК БО «Борівітер», яка має два незалежних лічильника інструкцій: повний Program Counters (PC), та «емульований» Link Register (LR) – БО демонструє достатньо високі показники реактивності. ОСРЧ FreeRTOS ефективно підтримує архітектуру ARM Cortex-M, оскільки не розрізняє функцій обробки переривань та звичайних функцій, тобто не підтримує класичних засобів обробки переривань як-то Linux на архітектурі x86 або amd64. Отримані експериментальні результати підтверджують, що використання ОСРЧ не погіршує реактивність системи в порівнянні з програмним рішенням, заснованим на «чистому залізі» (Bare Metal).

Правила моделювання в розробленому авторському ПЗ для чисельного експерименту наступні:

- пріоритети впорядковано наступним чином: найвищий пріоритет має первинна обробка перери-

вань, середній – вторинна обробка переривань жорсткого РЧ, найнижчий – вторинна обробка переривань м'якого РЧ;

- якщо при надходженні події (переривання) обчислювач зайнятий первинною обробкою – подія ставиться в первинну чергу;

- після завершення первинної обробки для тієї ж події створюється "повідомлення" у одну з двох вторинних черг для подій м'якого та жорсткого РЧ;

- вторинна обробка може початися тільки коли первинна черга порожня;

- вторинна обробка переривається і зберігає залишок часу, якщо в системі з'являється нова подія, яка вимагає первинного обслуговування;

- обслуговування не переривається під час первинної стадії.

Для моделювання використано подієвий (event-driven) підхід:

- події жорсткого реального часу (зовнішні переривання) генеруються з постійною частотою, що відповідає ситуації WCET;

- події м'якого РЧ (зовнішні переривання) генеруються з інтервалами відповідно до експоненційного закону розподілення випадкової величини;

- по кожному завданню для подальшого аналізу реєструються наступні відмітки часу: arrival – час появи переривання, primary_done – виконано первинну обробку, secondary_done_HRT – виконано вторинну обробку події жорсткого реального часу, secondary_done_SRT – виконано вторинну обробку події м'якого реального часу.

Результати розрахунку показано на рис. 4. Чисельні експерименти, результати яких використовувались для перевірки розробленої моделі реактивності,

проводились при наступних вхідних параметрах:

- імітація обробки 10 000 подій (переривань);
- середня інтенсивність потоку випадкових подій м'якого РЧ – 1000, 1/с, частота подій жорсткого РЧ фіксована – 2000 Гц;

- тривалість первинної обробки переривань фіксована та відповідає результатам натурного експерименту 0.002 мс;

- тривалість вторинної обробки подій м'якого РЧ є змінною величиною в діапазоні [0.3, 0.6], мс;

- тривалість вторинної обробки подій жорсткого РЧ фіксована – 0.1 мс.

Після обробки вказаної кількості подій програмне забезпечення для чисельного експерименту розраховує наступні показники.

1. Середній та максимальний часи для:

- очікування первинної обробки;
- вторинної обробки для подій м'якого РЧ та подій жорсткого РЧ;

- повного часу перебування в системі події м'якого РЧ та подій жорсткого РЧ.

2. Часове середнє значення довжин черг первинного обробника та вторинних обробників для подій м'якого РЧ та подій жорсткого РЧ.

3. Імовірності порушення часових обмежень (дедлайну) для подій м'якого РЧ та жорсткого РЧ.

Середній час очікування в чергах подій м'якого реального часу при наближенні до повної завантаженості процесора (коефіцієнт утилізації дорівнює 0.97) досягає 10 мс (рис. 4.а). Якщо необхідно, щоби час виконання хоча-б половини вирішуваних завдань м'якого РЧ не перевищував 1 мс, потрібно не завантажувати процесор мікроконтролера більше, ніж на 0.8.

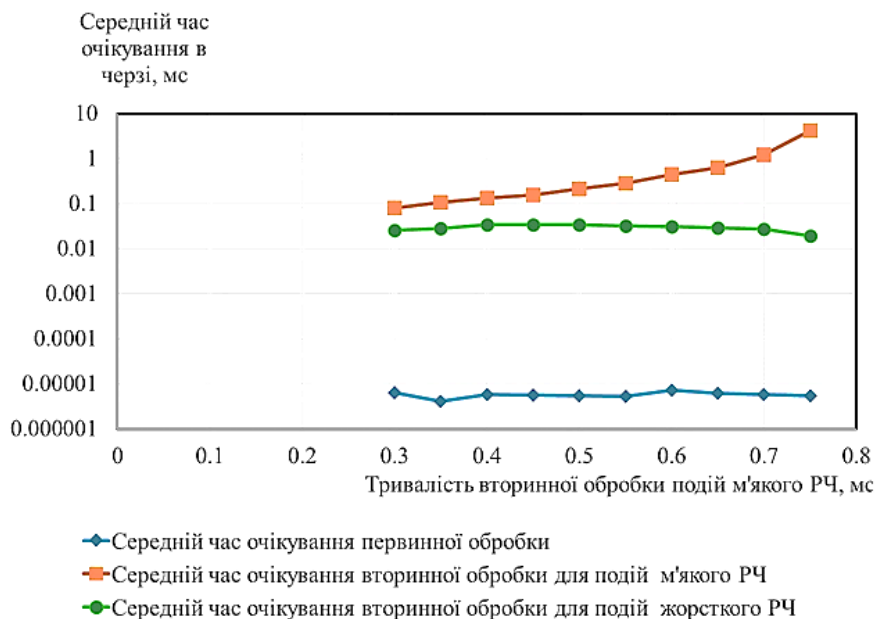
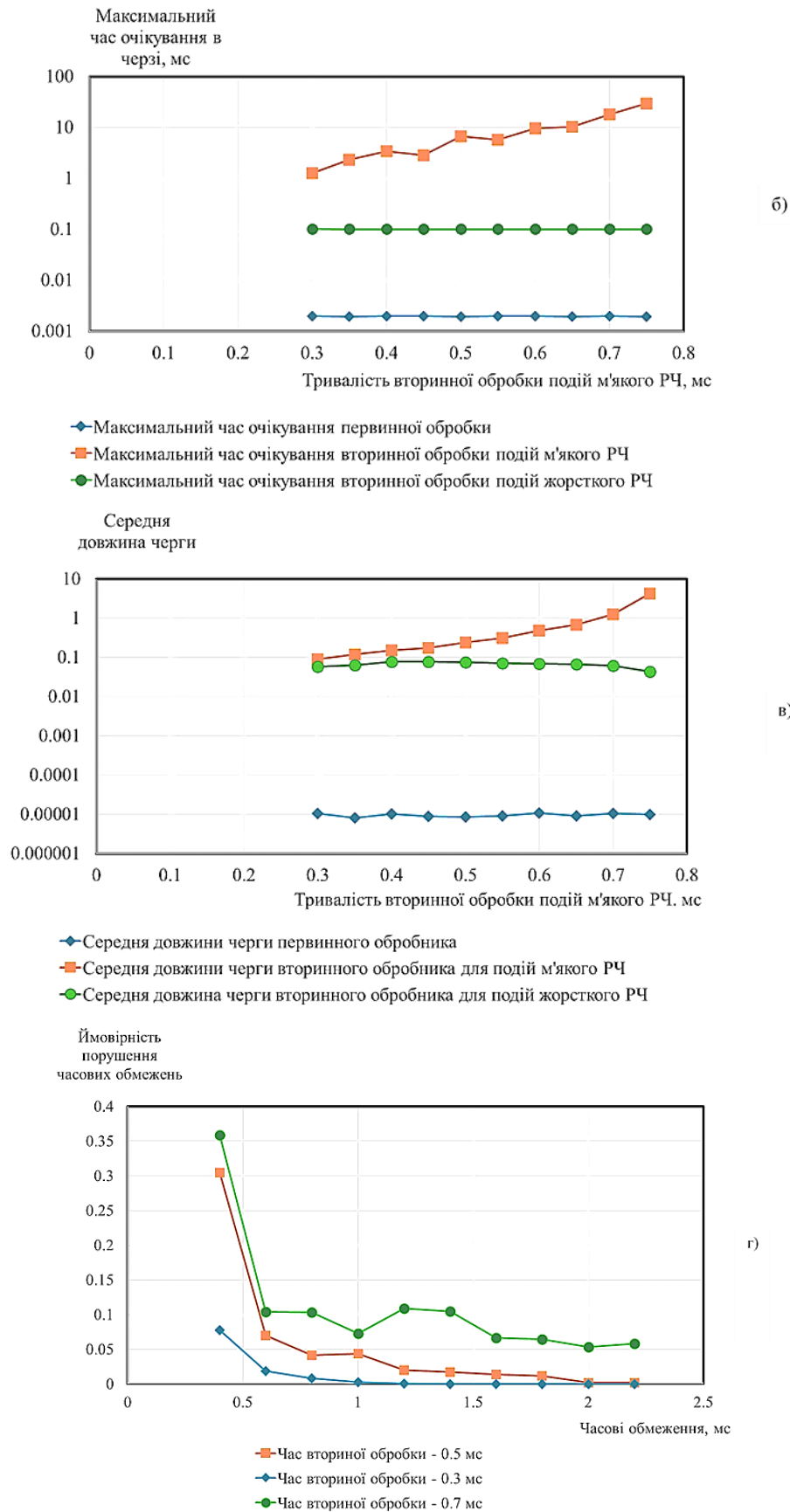


Рис. 4. Результати розрахунку:

а) середній час очікування в чергах та середній час від переривання до завершення обробки



Продовження рис. 4. Результати розрахунку: б) максимальний час очікування в чергах та максимальний час від переривання до завершення обробки, в) середні довжини черг, г) залежності ймовірності порушення часових обмежень від величини цих часових обмежень

Максимальний час очікування в чергах подій м'якого РЧ при наближенні до повної завантаженості процесора (коефіцієнт утилізації дорівнює 0.97) досягає 30 мс (рис. 4.б). При дотриманні обмеження в 10 мс на максимальний час очікування в чергах подій м'якого РЧ, необхідно встановити коефіцієнт утилізації процесора не більше, ніж 0.6. Середні довжини черги (рис. 4.в) визначають витрати оперативної пам'яті для підтримки запланованих черг. Відповідно до результатів моделювання при повному завантаженні процесора необхідно розраховувати на середню довжину черги вторинного обробника подій м'якого реального часу до 10 елементів. На рис. 4.г представлено результати розрахунку залежності ймовірності порушення часових обмежень від величини цих часових обмежень. Для розглянутих варіантів (для демонстрації обрано варіанти, коли вторинна обробка вимагає 0.3, 0.5, 0.7 мс одиниць часу) ймовірність порушення часових обмежень не буде перевищувати 0.1, якщо обмеження складає 0.6 мс.

Таким чином, результати чисельного експерименту не суперечать гіпотезі про коректність запропонованої математичної моделі реактивності.

Модель побудована в припущенні двоетапної процедури обробки переривань та враховує як роботу первинних обробників (ISR), які реалізуються через механізми операційної системи, так і вторинних, розроблених користувачем. В основу запропонованої моделі реактивності покладено гіпотезу, що за умови гарантованої обробки подій жорсткого реального часу в визначені терміни реактивність БО можна розрахувати як ймовірність своєчасної обробки подій м'якого РЧ. Відповідно модель вирішує два основних завдання: перевіряє на вказаному часовому інтервалі гарантованість обробки подій жорсткого реального часу та визначає ймовірність своєчасної обробки подій м'якого реального часу. Події жорсткого РЧ поступають на вхід системи з максимальної (WCET) частотою, події м'якого РЧ – з інтервалами відповідно до експоненційного закону розподілення випадкової величини. Адаптивність запропонованої моделі забезпечується тим, що оцінка тривалості майбутніх idle-періодів та очікуваної відповідно моделі енергетичної вигоди постійно оновлюється за результатами попередніх циклів роботи. Це дозволяє планувальнику динамічно підлаштовуватися під зміну характеру навантаження, частоти переривань і поведінки периферійних пристроїв.

6. Висновки

Стаття присвячена питанням оцінки та аналізу реактивності мікроконтролера вбудованої обчислювальної системи як складової загальної технічної ефективності апаратно-програмного комплексу КА та БПЛА. В результаті досліджень:

- на основі аналізу технічних вимог і практичного досвіду сформовано обмеження (Worst-Case Execution Time, WCET) для типових бортових завдань обчислювача та сформульовано вимоги до моделі;

- в нотаціях теорії множин описано комплексну модель цільових показників ефективності БО, яка призначена для розрахунку залежностей цільових показників від налаштувань апаратно-програмної частини та характеристик завдань БО;

- запропонована математична модель реактивності, яка визначає здатність бортового обчислювача своєчасно обробляти зовнішні події двох різних класів реального часу - жорсткого (hard real-time) та м'якого (soft real-time) з урахуванням стохастичних характеристик потоку подій та часових обмежень;

- виконано чисельний експеримент для перевірки запропонованої моделі реактивності. Результати чисельного експерименту не суперечать гіпотезі про коректність запропонованої математичної моделі реактивності. Оцінка реактивності платформи БО «Борівітер» під керуванням FreeRTOS підтверджує можливість використання платформи для завдань з швидко-змінюваним обчислювальним контекстом та зовнішніми чинниками. В якості конкретних числових характеристик витрат часу на службові операції ОС використано результати попередніх досліджень авторів [17].

У подальших дослідженнях заплановано розробку інструментального ПЗ для чисельного розрахунку реактивності МК з можливістю аналізу реактивності системи з урахуванням високого рівня автономності та можливостей впровадження засобів AI/ML. ПЗ надасть можливість імітації множин завдань жорсткого та м'якого РЧ з індивідуальними параметрами кожного завдання. Таке інструментальне програмне забезпечення необхідно і для підтримки розробки безпілотних платформ, і для реалізації наносупутника ХАІІ-КА, а також для методичного наповнення навчальної дисципліни «Інженерія програмного забезпечення наносупутників та безпілотних повітряних літальних апаратів», що започаткована в Національному аерокосмічному університеті.

Внесок авторів: концепція моделі реактивності, верифікація отриманих результатів – **Ігор Туркін**; розробка математичної моделі реактивності, виконання експериментів – **Олександр Любимов**; розробка програмної реалізації моделі реактивності, підготовка графічних матеріалів – **Ліна Волобуєва**.

Конфлікт інтересів

Автори заявляють, що немає конфлікту інтересів щодо цього дослідження, фінансового, особистого, авторського чи іншого, який міг би вплинути на дослідження та його результати, представлені в цій статті.

Фінансування

Дослідження проведено в рамках проекту 2023.04/0143 «Експериментальне відпрацювання бортового обчислювача безпілотної літальної апарату подвійного призначення» за фінансової підтримки Національного фонду досліджень України.

Доступність даних

Рукопис не має пов'язаних даних.

Використання засобів штучного інтелекту

Автори використовували технології штучного інтелекту в допустимих межах для надання власних перевірених даних, як описано в розділі методології дослідження.

Подяки

Автори висловлюють вдячність інжиніринговій компанії ТОВ «ЕКТОС-УКРАЇНА» за надання до проведення експерименту апаратних платформ та допомогу у налаштуванні та налагодженні інструментів для портування. Для детальнішої інформації <https://dev.ektos.net>.

Усі автори прочитали та погодилися з опублікованою версією цього рукопису.

Література

- Butt, M. Z. A review of perception sensors, techniques, and hardware architectures for autonomous low-altitude UAVs in non-cooperative local obstacle avoidance [Text] / M. Z. Butt, N. Nasir, & R. A. Rashid // *Robotics and Autonomous System*. - 2024. - Vol. 173. - Article no. 104629. DOI: 10.1016/j.robot.2024.104629.
- CubeSat Design Specification (1U – 12U) Rev 14.1 [Electronic Resource] / CubeSat Program, CAL POLY San Luis Obispo, 2022. – Access mode: https://static1.squarespace.com/static/5418c831e4b0fa4ecac1bacd/t/62193b7fc9e72e0053f00910/1645820809779/CDS+REV14_1+2022-02-09.pdf (accessed 02.11. 2025).
- On-Board Computer for CubeSats: State-of-the-Art and Future Trends [Text] / A. Cratere, L. Gagliardi, G. A. Sanca, F. Golmar, & F. Dell'Olio // *IEEE Access*. – 2024. - Vol. 12. - P. 99537-99569. DOI: 10.1109/ACCESS.2024.3428388.
- Liubimov, O. Use Of Open-Source Cots/Mots Hardware And Software Platforms For The Build Up Of The Cubesat Nanosatellites [Text] / O. Liubimov, & M. Liubimov // *Journal of Rocket-Space Technology*. – 2023. – Vol. 31, no. 4. – P. 138–147. DOI: 10.15421/452318.
- Russ, S. H. Power-Efficient Software Architecture for a CubeSat System [Text] / S. H. Russ, E. Spencer, T. Scroggins, & S. Latif // *SoutheastCon*. - Raleigh, 2020. - P. 1-6. DOI: 10.1109/SoutheastCon44009.2020.9249713.
- CubeSat Flight Software: Insights and a Case Study [Text] / M. Eshaq, M. S. Zitouni, Sh. Atalla, S. Al-Mansoori, & M. Macdonald // *Journal of Spacecraft and Rockets*. – 2025. – Vol. 62, no. 4. – P. 1328-1345. DOI: 10.2514/1.A35882.
- Probabilistic Safe WCET Estimation for Weakly Hard Real-time Systems at Design Stages [Text] / J. Lee, S. Y. Shin, L. C. Briand, & Sh. Nejati // *ACM Transactions on Software Engineering and Methodology*. - 2023. - Vol. 33, no. 2. - Article no. 32. DOI: 10.1145/3617176.
- Worst-Case Execution Time Analysis of a Real-Time System based on Arduino in CAN Network [Text] / A. Merazga, D. Rahem, F. Moulahcene, D. Kebbal, R. Khelaf, & I. Benacer // *Engineering, Technology & Applied Science Research*. - 2025. – Vol. 15, no. 3. – P. 23467–23478. DOI: 10.48084/etasr.10990.
- Kulu, E. CubeSats & Nanosatellites - 2024 Statistics, Forecast and Reliability [Text] / E. Kulu // *31st IAA Symposium on Small Satellite Missions / International Astronautical Federation (IAF)*. - Milan, 2024. – P. 601-614. DOI: 10.52202/078365-0069.
- Timperley, L. Reliability Analysis and Failure Mitigation Strategies for the PROVE Pathfinder CubeSat Payload [Text] / L. Timperley, & L. Berthoud // *IEEE Aerospace Conference (AERO)*. – Big Sky, 2022. – P. 1-18. DOI: 10.1109/AERO53065.2022.9843234.
- Reusable and Reliable Flight-Control Software for a Fail-Safe and Cost-Efficient Cubesat Mission: Design and Implementation [Text] / I. Latachi, T. Rachidi, M. Karim, & A. Hanafi // *Aerospace*. – 2020. – Vol. 7, no. 10. – Article no. 146. DOI: 10.3390/aerospace7100146.
- Masat-1 Mission Passport [Electronic Resource]. – Access mode: <https://www.eoportal.org/satellite-missions/masat-1#spacecraft>. (accessed: 02.11.2025).
- Foresail-1p Spacecraft Database Entry at Nanosats Database [Electronic Resource]. – Access mode: <https://www.nanosats.eu/sat/foresail-1>. (accessed 02.11.2025).
- Сторінка кафедри інженерії програмного забезпечення Національного аерокосмічного університету, присвячена проекту «Експериментальне відпрацювання бортового обчислювача безпілотної літальної апарату подвійного призначення» [Електронний ресурс]. – Режим доступу: <https://se.khai.edu/grant-nfdu/>. (дата звернення: 02.11.2025).
- Hitomi NASA Spacecraft Registry [Electronic Resource]. – Access mode: <https://heasarc.gsfc.nasa.gov/docs/hitomi/>. (accessed 02.11.2025).
- Design and performance of Soft Gamma-ray Detector onboard the Hitomi (ASTRO-H) satellite [Text] / H. Tajima, Sh. Watanabe, Ya. Fukazawa, R. D. Blandford, T. Enoto, A. Goldwurm, K. Hagino, K. Hayashi, Yu. Ichinohe, J. Kataoka, Ju. Katsuta, T. Kitauchi, M. Kokubun, Ph. Laurent, F. Lebrun, O. Limousin, G. M. Madejski, K. Makishima, T. Mizuno, K. Mori, T. Nakamori, T. Nakano, K. Nakazawa, H. Noda, H. Odaka, M. Ohno, M. Ohta, Sh. Saito, G. Sato, R. Sato, Sh. Takeda, H. Takahashi, T. Takahashi, T. Tanaka, Ya. Tanaka, Yu. Terada, H. Uchiyama, Ya. Uchiyama, K. Yamaoka, Y. Yatsu, D. Yonetoku, & T. Yuasa // *Journal of Astronomical Telescopes, Instruments, and Systems*. – 2018. - Vol.

4. – Article no. 02141. DOI: 10.1117/1.JATIS.4.2.021411.

17. UAV Mission Computer Operation Mode Optimization Focusing on Computational Energy Efficiency and System Responsiveness [Text] / O. Liubimov, I. Turkin, V. Cheranovskyi, & L. Volobuyeva // *Computation* – 2024. – Vol. 12, no. 12, - Article no. 235. DOI: 10.3390/computation12120235.

18. Bernat, G. Weakly hard real-time systems [Text] / G. Bernat, A. Burns, & A. Liamosi // *IEEE Transactions on Computers*. – 2001. – Vol. 50, no. 4. – P. 308-321. DOI: 10.1109/12.919277.

19. Zhang, F. Schedulability Analysis for Real-Time Systems with EDF Scheduling [Text] / F. Zhang, & A. Burns // *IEEE Transactions on Computers*. – 2009. – Vol. 58, no. 9. – P. 1250-1258. DOI: 10.1109/tc.2009.58.

20. Ложковський, А. Г. Теорія масового обслуговування в телекомунікаціях [Текст] / А. Г. Ложковський. – Одеса: ОНАЗ ім. О.С. Попова, 2010. – 112 с.

References

1. Butt, M. Z., Nasir, N., & Rashid, R. A. A review of perception sensors, techniques, and hardware architectures for autonomous low-altitude UAVs in non-cooperative local obstacle avoidance. *Robotics and Autonomous Systems*, 2024, vol. 173, article no. 104629. DOI: 10.1016/j.robot.2024.104629.

2. CubeSat Design Specification (1U – 12U) Rev 14.1. CubeSat Program, CAL POLY San Luis Obispo, 2022. Available at: https://static1.squarespace.com/static/5418c831e4b0fa4ecac1bacd/t/62193b7fc9e72e0053f00910/1645820809779/CDS+REV14_1+2022-02-09.pdf. (accessed 02.11.2025).

3. Cratere, A., Gagliardi, L., Sanca, G. A., Golmar, F., & Dell'Olio, F. On-Board Computer for CubeSats: State-of-the-Art and Future Trends. *IEEE Access*, 2024, vol. 12, pp. 99537-99569. DOI: 10.1109/ACCESS.2024.3428388.

4. Liubimov, O., & Liubimov, M. Use of open-source Cots/Mots hardware and software platforms for the build up of the CubeSat nanosatellites. *Journal of Rocket-Space Technology*, 2023, vol. 31, no. 4, pp. 138-147. DOI: 10.15421/452318.

5. Russ, S. H., Spencer, E., Scroggins, T., & Latif, S. Power-Efficient Software Architecture for a CubeSat System. *Proceedings of the SoutheastCon*, Raleigh, NC, USA, IEEE, 2020, pp. 1-6, DOI: 10.1109/SoutheastCon44009.2020.9249713.

6. Eshaq, M., Zitouni, M. S., Atalla Sh., Al-Mansoori, S., & Macdonald, M. CubeSat Flight Software: Insights and a Case Study. *Journal of Spacecraft and Rockets*, 2025, vol. 62, no. 4, pp. 1328-1345. DOI: 10.2514/1.A35882.

7. Lee, J., Shin, S. Y., Briand, L. C., & Nejati, Sh. Probabilistic Safe WCET Estimation for Weakly Hard Real-time Systems at Design Stages. *ACM Transactions on Software Engineering and Methodology*, 2023, vol. 33, no. 2, article no. 32. DOI: 10.1145/3617176.

8. Merazga, A., Rahem, D., Moulahcene, F., Kebbal, D., Khelaf, R., & Benacer, I. Worst-Case Execution Time Analysis of a Real-Time System based on Arduino in CAN Network. *Engineering, Technology & Applied Science Research*, 2025, vol. 15, no. 3, pp. 23467–23478. DOI: 10.48084/etasr.10990.

9. Kulu, E. CubeSats & Nanosatellites - 2024 Statistics, Forecast and Reliability. *Proceedings of the 31st IAA Symposium on Small Satellite Missions*. International Astronautical Federation (IAF), Milan, Italy, 2024, pp.601-614. DOI: 10.52202/078365-0069.

10. Timperley, L. & Berthoud, L. Reliability Analysis and Failure Mitigation Strategies for the PROVE Pathfinder CubeSat Payload. *Proceedings of the 2022 IEEE Aerospace Conference (AERO)*. Big Sky, MT, USA, 2022, pp. 1-18. DOI: 10.1109/AERO53065.2022.9843234.

11. Latachi, I., Rachidi, T., Karim, M., & Hanafi, A. Reusable and Reliable Flight-Control Software for a Fail-Safe and Cost-Efficient Cubesat Mission: Design and Implementation. *Aerospace*, 2020, vol. 7, article no. 146. DOI: 10.3390/aerospace7100146.

12. Masat-1 Mission Passport. Available at: <https://www.eoportal.org/satellite-missions/masat-1#spacecraft> (accessed 02.11.2025).

13. Foresail-1p Spacecraft Database Entry at Nanosats Database. Available at: <https://www.nanosats.eu/sat/foresail-1> (accessed 02.11.2025).

14. Page of the Department of Software Engineering of the National Aerospace University, dedicated to the project "Experimental development of an on-board computer of a dual-purpose unmanned aerial vehicle". Available at: <https://se.khai.edu/grant-nfdu/> (accessed 02.11.2025). (In Ukrainian).

15. Hitomi NASA Spacecraft Registry. Available at: <https://heasarc.gsfc.nasa.gov/docs/hitomi/> (accessed 02.11.2025).

16. Tajima, H., Watanabe, Sh., Fukazawa, Ya., Blandford, R. D., Enoto, T., Goldwurm, A., Hagino, K., Hayashi, K., Ichinohe, Yu., Kataoka, J., Katsuta, Ju., Kitaguchi, T., Kokubun, M., Laurent, Ph., Lebrun, F., Limousin, O., Madejski, G. M., Makishima, K., Mizuno, T., Mori, K., Nakamori, T., Nakano, T., Nakazawa, K., Noda, H., Odaka, H., Ohno, M., Ohta, M., Saito, Sh., Sato, G., Sato, R., Takeda, Sh., Takahashi, H., Takahashi, T., Tanaka, T., Tanaka, Ya., Terada, Yu., Uchiyama, H., Uchiyama, Ya., Yamaoka, K., Yatsu, Y., Yonetoku, D., & Yuasa, T. Design and performance of Soft Gamma-ray Detector onboard the Hitomi (ASTRO-H) satellite. *Journal of Astronomical Telescopes, Instruments, and Systems*, 2018, vol. 4, no. 2, article no. 021411. DOI: 10.1117/1.JATIS.4.2.021411.

17. Liubimov, O., Turkin, I., Cheranovskiy, V., & Volobuyeva, L. UAV Mission Computer Operation Mode Optimization Focusing on Computational Energy Efficiency and System Responsiveness. *Computation*, 2024, vol. 12, article no. 235. DOI: 10.3390/computation12120235.

18. Bernat, G., Burns, A., & Liamosi, A. Weakly hard real-time systems. *IEEE Transactions on Computers*, 2001, vol. 50, no. 4, pp. 308-321. DOI: 10.1109/12.919277.

19. Zhang, F., & Burns, A. Schedulability Analysis for Real-Time Systems with EDF Scheduling. *IEEE*

Transactions on Computers, 2009, vol. 58, no. 9, pp. 1250-1258. DOI: 10.1109/tc.2009.58.

20. Lozhkovskiy, A.H. *Queueing theory in telecommunications*. Odesa, Publ. ONAZ, 2010. 112 p. (In Ukrainian).

Отримано 10.09.2025, отримано у доопрацьованому вигляді 24.09.2025

Дата ухвалення 17.11.2025, дата публікації 08.12.2025

REACTIVITY MODEL OF NANOSATELLITE AND UAVS EMBEDDED COMPUTING SYSTEMS

*Ihor Turkin, Oleksandr Liubimov,
Lina Volobuieva*

The embedded computer is a key element of the avionics of both spacecraft and unmanned aerial vehicles (UAVs). Its reactivity determines the system's ability to ensure that the reaction time to external events meets the specified real-time constraints. **This study aims to investigate** the reactivity of the microcontroller of the embedded computing system as a component of the overall technical efficiency of the hardware and software complex. The mathematical model of reactivity, as well as methods, tools, and technology for experimental research of the reactivity of a real-time operating system, are **the subjects of this study**. **This study aims** to develop and verify a model of the reactivity of embedded computer software for use in nanosatellites and unmanned aerial vehicles. This study substantiated the need to create a set-theoretic model of the reactivity of embedded avionics systems for small spacecraft and unmanned platforms. A constraint analysis (Worst-Case Execution Time, WCET) was performed for typical onboard computer tasks. The requirements for the model were formulated, and its mathematical description and software implementation were developed. Numerical modeling methods were used to verify the simulation results' compliance in the actual processes in the system. **Results**. The proposed mathematical model of reactivity considers the stochastic characteristics of event flows and two-stage interrupt processing, allowing the probability of a timely system response to various types of events to be assessed. Experimental verification confirmed the model's correctness and suitability for analyzing the Boryviter / Falco platform under the control of the FreeRTOS operating system. **The scientific novelty** lies in the development of a new concept for optimizing the work of the real-time operating system scheduler of UAV and nanosatellites OBC, which is based on a complex mathematical model of energy consumption, computational performance and reactivity of the OBC, which, unlike power consumption control mechanisms in processors with dynamic frequency or voltage regulation, is specifically focused on OBC microcontrollers, in which the frequency and voltage are fixed, and optimization is possible only through the correct choice of the policy for controlling energy-saving modes. **Conclusions**. The developed model can be used to assess and improve the reactivity of computing systems, including the university nanosatellite Khai-1KA and unmanned aviation systems. Further research is aimed at creating instrumental software for reactivity analysis to support software engineering of new generations of aerospace and unmanned platforms with a high level of autonomy, considering the implementation of AI/ML tools.

Keywords: Nanosatellite; UAV, Onboard Computer (OBC), Software; Reactivity; Reactivity Model.

Туркін Ігор Борисович – д-р техн. наук, проф., зав. каф. інженерії програмного забезпечення, Національний аерокосмічний університет «Харківський авіаційний інститут», Харків, Україна.

Любимов Олександр Вікторович – асп. каф. інженерії програмного забезпечення, Національний аерокосмічний університет «Харківський авіаційний інститут», Харків, Україна.

Волобуєва Ліна Олексіївна – канд. техн. наук, доц., доц. каф. інженерії програмного забезпечення, Національний аерокосмічний університет «Харківський авіаційний інститут», Харків, Україна.

Ihor Turkin – Doctor of Technical Science, Professor, Head at the Department of Software Engineering, National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine, e-mail: i.turkin@khai.edu, ORCID: 0000-0002-3986-4186, Scopus Author ID: 57203145725.

Oleksandr Liubimov – Ph.D. Student at the Department of Software Engineering, National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine, e-mail: oleksandr.liubimov@gmail.com, ORCID: 0000-0003-3636-6939, Scopus Author ID: 58099287400.

Lina Volobuieva – Candidate of Technical Sciences, Associate Professor, Associate Professor at the Department of Software Engineering, National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine, e-mail: l.volobuieva@khai.edu, ORCID: 0000-0002-3466-5743, Scopus Author ID: 57203147598.