

# Паралельні інформаційні процеси динамічного сприйняття оточення автономною рухомою платформою

*О.М. Сухоручкіна, С.О. Ільїн, М.В. Прогонний*

Міжнародний науково-навчальний центр інформаційних технологій та систем  
Національної академії наук України і Міністерства освіти і науки України, Україна  
sukhoru@irtc.org.ua

## Анотація

Для управляючої системи автономної рухомої платформи запропоновано методику організації паралельних обчислювань, які обслуговують паралельні інформаційні процеси, у тому числі процеси формування та розпізнавання образів у середовищі функціонування платформи.

## 1. Вступ

Інформаційні процеси – збір, обробка, передача, використання та зберігання інформації – важливі процеси функціонування складних технічних систем, які спроможні автономно виконувати різні завдання у нестационарному середовищі. При розробці рухомих технічних систем з інтелектуальними можливостями, виникають питання практичної реалізації в їх управляючих системах (УС) інформаційних процесів, які протікають паралельно у часі, відповідних ним обчислювань та звернень до структур пам'яті.

Теоретичні аспекти систем зі штучним інтелектом [наприклад, 1, 2] вказують на суттєву роль у механізмах прийняття рішень взаємозв'язку процесів аналізу поточної інформації з пошуком аналогів у накопленому раніше досвіді, якій потрібно представити у деякому формалізованому вигляді. Дана робота спирається на результати розробки УС з інтелектуальними можливостями для багатофункціональної рухомої платформи (БРП), де передбачено два типи обов'язкових паралельних інформаційних процесів, а саме:

1) отримання, аналіз та використання фактичної поточної інформації від фізичних модулів платформи – сенсорів, відеокамери, ефекторів;

2) формування та використання як прогнозу очікуваних станів БРП інформації по кожному з спостережуваних параметрів УС, спираючись на знання розробника про об'єкт управління та характер його оточення, представлені у вигляді формалізованих моделей.

Кожний з двох типів інформаційних процесів сам по собі також неоднорідний та може складатись з багатьох незалежних або взаємозв'язаних паралельно функціонуючих процесів, що обслуговують різні сенсорні канали та керовані модулі БРП. УС відповідає також за інтеграцію інформації у контексті поточного стану для прийняття рішень про доцільну поведінку БРП у цілому. Ця задача представляє окремий інформаційний процес.

Прагнення до ефективної практичної реалізації множинної паралельності різнорідних інформаційних процесів при розробці інтелектуального управління БРП породжує відповідні вимоги до обчислювальних ресурсів БРП та методики програмування.

У загальноприйнятому підході розробники технічних систем як об'єктів управління розміщують в одному послідовному циклі активізацію його базових блоків – отримання даних від сенсорів, аналізу інформації, прийняття рішень про управляючі впливи, мінімізації розузгодження спостережуваних параметрів з очікуваними за цільовою функцією та інші [2]. Такі системи представляють певну практичну цінність, але строго обмежені за функціональними можливостями.

Зазначені вище два типи паралельних інформаційних процесів УС оперують даними різного походження. В першому випадку джерелом є дані про фізичні властивості реального світу, що реєструють різні пристрої – датчики, сенсори. Таки дані мають різні час появи, точність, шуми та взагалі мають випадковий характер, що потребує відповідних рішень для отримання з них корисної інформації та зведення сукупної оцінки стану об'єкта управління до єдиної часової шкали.

Для другого типу інформаційних процесів джерелом даних є математичні моделі. Ці процеси більш передбачувані за потрібними обчислювальними ресурсами та їх розподілом у часі. Формування даних відбувається з передбачуваними розробником математичних моделей частотою, точністю та корисністю.

Сучасні багатоядерні процесори персональних комп'ютерів (ПК) та принципи багатопотокового програмування, що підтримують операційні системи ПК, дають принципову змогу виконання паралельних та умовно паралельних обчислювань на борту БРП. Однак можливості інтелектуальної поведінки БРП суттєво залежать від якості виконання обчислювальним середовищем взаємозв'язку всіх інформаційних процесів на верхньому рівні управління.

В даній роботі розглянуто питання практичної реалізації множинної паралельності інформаційних процесів експериментального зразка УС БРП, у тому числі процесів ідентифікації об'єктів нестационарного оточення.

Робота виконується за ДНТП України „Образний комп'ютер”.

## 2. Склад та структура УС БРП

При створенні експериментального зразка БРП з інтелектуальними властивостями нами розроблені структура УС та відповідне програмне забезпечення (ПЗ), які підтримують множинну паралельність інформаційних процесів та орієнтовані на обчислювальні ресурси стандартних персональних комп'ютерів (ПК) з операційною системою (ОС) MS Windows, як ОС широкого застосування [3–5].

Також запропонована двохмодульна УС верхнього рівня [5], де кожен модуль є окреме Windows-прикладення, що забезпечують відповідно перший та другий тип згаданих вище інформаційних процесів. Дану УС обслуговує розподілене обчислювальне середовище із двох ПК з двох ядерними процесорами.

Розроблено програмні механізми взаємодії та синхронізації паралельних обчислювальних процесів з роздільними ресурсами двох модулів УС.

До складу УС входять база декларативних знань про сам об'єкт управління та його оточення у вигляді математичного опису просторових геометричних моделей твердих тіл, їх кольору, кінематичних та складених взаємозв'язків, положення у просторі, а також вербальних ідентифікаторів.

Процедурні знання УС складають три рівня бібліотеки правил поведінки (БПП), а саме, команди, поведінки та завдання. Всі складові БПП мають власні екземпляри у кожному модулі УС. На першому рівні БПП – командному ці екземпляри суттєво відрізняється. Той модуль, що відповідає за фактичні процеси БПП, містить програмно реалізовану взаємодію з фізичними складовими платформи у вигляді команд керування двигунами, опиту датчиків, сенсорів та відповідні алгоритми обробки та інтерпретації їх сигналів. Другий модуль УС на рівні команд містить математичні моделі всіх функціональних компонентів БПП, які формують моделі спостережуваних параметрів об'єкта управління.

На другому та третьому рівнях екземпляри БПП у модулях УС принципово не відрізняються та містять відповідні алгоритми умовних реакцій у вигляді команд на зовнішні зміни (рівень поведінок) та алгоритми досягнення заданої цілі (рівень завдань) як логічні зв'язки поведінок.

Розроблені методики програмування у вигляді поточкових класів вказаних структур компонентів БПП. Функціонування БПП під час виконання завдань супроводжується двома головними паралельними інформаційними процесами, які забезпечують два ПК, та необхідним числом паралельних та умовно паралельних інформаційних процесів у кожному модулі УС на кожному з двох ПК, де обчислення виконують декілька потоків. Таким чином в межах кожного модуля УС використані переваги єдиного адресного простору як загальної пам'яті однорідних інформаційних процесів.

Автономне функціонування БПП для досягнення цільового стану зазначеного в завданні користувача відбувається завдяки роботі базової функції верхнього рівня УС по адаптуванню системи до наявних умов. Цю функцію складають три механізми:

- реєстратор поточної активності в модулях УС компонентів БПП;
- компаратор для порівняння фактично спостережуваних параметрів стану системи з очікуваними за моделями та базою знань;
- диспетчер для прийняття рішень про доцільну діяльність БПП у випадках розбігу фактичних та очікуваних параметрів стану системи або при невизначеності параметрів її цілі.

Для синхронізації паралельних процесів двох модулів УС розроблено спеціальний програмний механізм.

Взаємозв'язки складових базової функції верхнього рівня УС зображено на рис. 1.

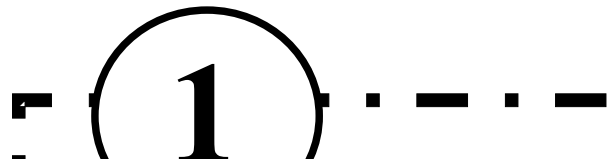


Рис. 1. Схема взаємовідношень компонентів та механізмів базової функції автономного управління БПП. Тут 1 – модуль УС по підтримці фізичних процесів; 2 – модуль УС по підтримці модельних процесів; С – механізм синхронізації процесів; Р – реєстратор; Д – диспетчер; К – компаратор

Завдяки роботі компаратора УС має можливість детектувати моменти виникнення критичних розбіжностей в сприйнятті БПП спроектованої за модельним представленням та реального зовнішнього середовища та приймати ті чи інші рішення для їх коригування.

Взаємодію користувача з БПП реалізовано за принципом супервізорного управління. Одиницею мови людино-машинної взаємодії є завдання БПП. Формулювання завдання користувач може виконати як через графічний інтерфейс так і голосовим спілкуванням з УС. Інтегрування мовних комп'ютерних технологій до інтерфейсів систем управління – розпізнавання команд, генерації тексту та синтезу голосових повідомлень, аналізу текстового опису просторових сцен та інших задач – є фактично обов'язковим для сучасних технічних систем.

Людино-машинний інтерфейс з БПП – це окремий інформаційний процес, який вимагає рішення різномірних задач логічного, математичного та програмного характеру, безпосередньо пов'язаних з базою знань УС. Тому за ці задачі відповідає другий модуль УС.

Природна наявність багатьох різних за швидкістю та потрібними алгоритмічними і обчислювальними ресурсами паралельних інформаційних процесів сприйняття оточення, прийняття рішень про доцільну поведінку автономної БПП та спілкування з користувачем або іншими технічними системами ставить важливі питання відповідної синхронізації та розподілу ресурсного забезпечення цих процесів.

### 3. Синхронізація паралельних процесів УС

З точки зору архітектури ПЗ УС рухомої платформи є комплекс з двох взаємодіючих багатопоточних прикладень – модулів УС, які обслуговують два ПК. Дані ПК пов'язані локальною радіомережею Wi-Fi. Взаємодія двох модулів УС реалізована за технологією клієнт-сервер. Інформаційний обмін між програмами складає TCP/IP здійснює програмний мережевий компонент УС, як її додатковий модуль.

Слід відмітити, що сучасні ПК мають багатоядерні процесори з апаратною підтримкою багатопотокових

АКТИВНІ КО  
БПП  
Факти  
спостере  
параметр

обчислювань, що ідеально співпадає з запропонованою авторами програмної архітектури УС.

УС БРП має два типи паралельних процесів:

1) процеси в межах різних модулів УС, для синхронізації яких потрібен спеціальний механізм;

2) процеси в межах одного модулю УС, синхронізація яких забезпечена стандартними засобами ОС.

Традиційно використання багатопотоковості переслідує такі цілі:

– дозволити прикладенню паралельно відпрацьовувати декілька відносно незалежних задач;

– підвищувати продуктивність прикладення завдяки декількох процесорів або їх ядер.

Але використання багатопотоковості для УС БРП має іншу головну мету, а саме, забезпечення природної паралельності процесів сприйняття різними каналами інформації, її інтерпретації та інтеграції для прийняття рішень про доцільну поведінку автономної БРП. При чому, програмні рішення повинні враховувати навантаження на ОС по обробці критичних секцій, як базової методики програмування поточкових прикладень. Оцінки показують, що при реалізації УС традиційними рішеннями звернення до критичних секцій досягає 100 разів за секунду, що перевантажує ОС транзакціями та приводить до неоптимальних витрат процесорного часу та відвертанню програмного забезпечення від його безпосередніх задач по обробці інформації та управлінню.

Тому запропоновано використати технологію публікації-підписки, завдяки чому перевірка різнорідних сенсорних даних рознесена по контекстам відповідних потоків та знижено потребу їх синхронізації, оскільки при цьому відповідні реакції УС реалізовані шляхом постановки умов до черги та перевірки їх теми потоками, що володіють необхідними даними, що не потребує використання критичних секцій та між потокової синхронізації.

Принциповою відмінністю обчислювань за потребою двох модулів УС БРП від класичної теорії паралельних обчислювань [7] є те, що алгоритми, які виконують два модулі УС ресурсами двох ПК, інформаційно залежні один від одного. Причому, не завжди ця залежність є очевидною. Тому не можна провести попередньо структурне розподілення обчислювань між двома ПК.

Як сказано вище, важливо забезпечити скоординоване функціонування паралельних процесів двох модулів УС БРП на двох ПК. В ОС немає готових засобів такої синхронізації, а відомі підходи до рішення подібних задач з використанням стандарту паралельних обчислювань MPI (Message-Passing Interface) [6], дуже громіздкі для УС БРП. Тому в середовищі програмування Delphi розроблено спеціалізований механізм між модульної синхронізації, яка виконує дві основні функції: погоджує виконання модулями коду та забезпечує обмін даними між модулями.

Суть цієї синхронізації є у тому, що кожен з модулів УС у заздалегідь оговорених точках коду очікує виходу другого модуля до однойменної точки, та при необхідності приймає від нього дані. При цьому, очікування подій та обмін даними суміщені в одній операції.

Розробка між модульної синхронізації виконана на основі стандартних засобів ОС синхронізації потоків мережевого компоненту та оригінальної розробки, якій забезпечує взаємодію модулів УС.

Як особливість цієї синхронізації треба назвати автоматичну прив'язку точок синхронізації до контекстів

завдань – складових БПП: розробник коду відповідає лише за те, щоб на рівні логіки алгоритмів у командах, поведінках та завданнях БПП поіменувати відповідні точки синхронізації, а сама УС присвоює точкам синхронізації внутрішні унікальні ідентифікатори, виходячи з того, у контексті якої поведінки активізовано конкретну команду, та у контексті якого завдання активізовано конкретну поведінку. Такий підхід суттєво полегшив розробку системи та виключив можливі помилки синхронізації однакових команд та поведінок, коли їх використовують різні завдання.

Крім того, механізм синхронізації, що розроблено, доповнено можливістю між модульного обміну асинхронними повідомленнями. Ще однією особливістю цього механізму є реалізація зупинок у двох модульної УС у сенсі програмних переривань з очисткою пам'яті, збереженням даних та узгодженістю модулів при перериванні.

Оскільки алгоритми БПП розподілені між модулями заздалегідь, задача розподілу обчислювального навантаження між модулями УС не розглядається. Але дуже важливим є питання розподілу обчислювальних навантажень між компонентами у межах кожного з модулів УС БРП.

#### 4. Про проблеми розподілу обчислювальних навантажень між компонентами УС

Програмна обробка інформації необхідної для функціонування автономної рухомої платформи на практиці виливається в організацію спільної роботи потоків обчислення [4].

Зрозуміло що через принципово різну природу вхідних даних вони неоднорідно навантажені та їм потрібна різна обчислювальна потужність. Так потік обробки відео даних вимагає значних ресурсів оперативної пам'яті системи (ОП) та центрального процесора (ЦП), потоки ж обробки сенсорних даних – на порядки менших. З іншого боку, деякі об'єкти УС можуть значно змінювати свої потреби в системних ресурсах на протязі своєї роботи. Об'єкт компаратор, який слідкує за знаходженням параметрів функціонування автономної рухомої платформи в допустимих межах [5], для фоновий моніторингу досить невибагливий щодо обчислювальних потужностей, але у випадку детектування тієї чи іншої нештатної ситуації для прийняття рішень йому часто потрібно проводити досить значні обчислення.

Такі ситуації є типовими і досить гарно відпрацьованими при проектуванні прикладень для операційних систем реального часу (ОСРЧ) [2, 8], в нашому ж випадку для ОС MS Windows аналогічні рекомендації хоч і не дозволяють повністю позбавитись проблем нерівномірності навантаження і зростанні часу відгуку УС на зовнішні сигнали, але значно зменшують імовірність помилок в роботі чи відмову у обробці даних в критичні моменти часу.

Для цього на етапі проектування визначається множина усіх можливих потоків обчислення, що потенційно можуть працювати в УС паралельно. Експериментально визначаються суми необхідних їм обчислювальних ресурсів та накладається умова у вигляді

$$\sum_{i=1, N} Q_i + Q_{\max s} < Q_{\max} . \quad (1)$$

Де  $\sum_{i=1, N} Q_i$  – сума обчислювальних навантажень окремих  $N$  компонентів УС;  $Q_{\max s}$  – обчислювальні навантаження пов'язані з синхронізацією даних;  $Q_{\max}$  – максимально можлива обчислювальна потужність.

Якщо кожен з обчислювальних потоків володіє потрібними конфігураційними даними, тоді в ході виконання завдань автономною БРП можна динамічно визначати доцільність (принципову можливість) запуску додаткових обчислень, так щоб УС ніколи не втрачала здатність реагувати на динамічні зміни в середовищі функціонування БРП.

Не слід забувати також, що подібні величини є суто статистичними тому чим краще виконується нерівність (1) тим стабільніше працює УС БРП в цілому.

Окремих "природних" обмеженням програмних систем, які тісно пов'язані з керуванням механікою, є те що стан механічної системи динамічно змінюється за час, коли виконуються критичні для прийняття рішень обчислення. Отримана в їх результаті інформація втрачає актуальність або вимагатиме додаткових дій для подальшого її використання.

Для запобігання цьому, на етапі проектування потрібно закладати зв'язок між максимальним часом критичних обчислень та швидкістю зміни тих чи інших параметрів, руху, тощо автономної БРП.

## 5. Висновки

У даній роботі задачі інтелектуального управління розглянуті з точки зору організації необхідних паралельних інформаційних процесів функціонування автономної БРП дозволили розробити архітектуру УС верхнього рівня, відповідні структури представлення у її пам'яті інформації про взаємодію об'єкта управління з оточенням та формування його доцільної поведінки.

Вказані рішення практичної реалізації програмного забезпечення УС дозволили створити експериментальний зразок БРП, що здатен автономно виконувати складні завдання в умовах не повністю визначеного оточення, адаптувати свої дії до поточних ситуацій, звертаючись до відповідних методів інтерпретації інформації про оточення, та забезпечувати спілкування людини-користувача з технічною системою на рівні природно сприйманих динамічно синтезованих комп'ютерних просторових моделей та голосових повідомлень про стан системи.

## 6. Література

- [1] Хоккинс Дж., Блейкли С. Об интеллекте: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2007. – 240 с.
- [2] Жданов А.А. [Автономный искусственный интеллект](#). – М.: Бинوم. Лаборатория знаний, 2008. – 359 с.
- [3] Сухоручкина О.Н. Структуры функциональной организации интеллектуализированного управления мобильной системой // УСИМ, 2007. — № 3. — С. 26–33, 63.
- [4] Ільїн С.О. Особливості програмної реалізації системи керування мобільним роботом // УСИМ, 2007. – № 4. – С. 28–42.
- [5] Сухоручкина О.Н. Модели и структуры интеллектуального управления мобильным

роботом. – Экстремальная робототехника. Нано-, микро- и макророботы (ЭР-2009) // Материалы XX Международной научно-технической конференции. – Таганрог: Изд-во ТТИ ЮФУ, 2009. – С. 240–242

- [6] Gropp W., Lusk E., Skjellum A. Using MPI: Portable Parallel Programming with the Message-Passing Interface, 2nd Edn., MIT Press, 1999.
- [7] Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. — СПб.: БХВ-Петербург, 2002. – 608 с.
- [8] Ослэндер Д. М., Риджли Дж. Р., Рингенберг Дж. Д. Управляющие программы для механических систем: Объектно-ориентированное проектирование систем реального времени. — М.: Бинум. Лаборатория знаний, 2004. — 416 с.