

УДК 681.3.016

О. М. Цимбал

ПОДАННЯ ЗНАТЬ У СИСТЕМАХ ПРИЙНЯТТЯ РІШЕНЬ ІНТЕЛЕКТУАЛЬНИХ РОБОТІВ

1. Вступ

У спеціальній літературі існують різні думки з приводу того, яким чином визначити поняття робота. Зокрема, навчальна література з робототехніки радянського періоду, наприклад [1], визначає робота як універсальний автомат для відтворення моторних та інтелектуальних функцій людини. Японські дослідники визначають робота багатьма способами, зокрема у [3] наводиться визначення робота як пристрою, здатного самостійно переміщуватися в просторі, вирішувати завдання аналізу сцен та розпізнавання образів, який має велику кількість ступенів рухливості, спроможний аналізувати середовище за допомогою зворотних зв'язків, а також прогнозувати ситуації, спираючись на власний досвід та наявну інформацію.

Наведені визначення є вірними, якщо врахувати потенційні можливості роботів та робототехніки як науки, здатної у технічний спосіб реалізувати зростаючі можливості таких суміжних галузей, як прикладна математика, штучний інтелект та системи зв'язку. А поки що спеціалісти в галузі баз знань [2] констатують, що робототехніка є інженерною наукою, яка не заперечує технології штучного інтелекту, але не готова до їх впровадження з різних причин. Отже, дослідження в галузі застосування методів штучного інтелекту в робототехніці є актуальними з теоретичної та практичної точки зору.

Метою запропонованої статті є аналіз загальних особливостей систем керування роботів, реалізації систем прийняття рішень, розгляд подання даних та знань у таких системах.

2. Особливості систем керування роботів

Як відомо, на цей час розрізняють три покоління роботів [1, 4]:

- роботи з жорсткою схемою керування, або програмовані маніпулятори, до яких належать майже всі сучасні промислові роботи;
- адаптивні роботи з сенсорними пристроями, повільно знаходять застосування у промисловості;
- інтелектуальні роботи (роботи, здатні до самоорганізації) — мета робототехніки.

Різниця між поколіннями роботів пролягає, головним чином, в організації системи керування маніпулятором, яка складається з декількох рівнів [4].

На рівні окремої робочої ланки робота забезпечується керування двигуном, який здійснює рух

однієї або декількох ступенів рухливості. За допомогою центрального процесора системи керування організується координована робота приводів маніпулятора. При цьому вхідною інформацією є послідовність положень затискача маніпулятора або пов'язаного з ним об'єкта. Траєкторія захоплювального пристрою зазвичай задається двома способами: шляхом безпосереднього завдання людиною-оператором у процесі навчання або за допомогою планування рухів на більш високому рівні керування, у тому числі за допомогою методів штучного інтелекту.

На рівні планування рухів передбачається, що відомими є кінцева мета руху та опис робочої сцени. Пристрій керування, який забезпечує впріщення завдань цього рівня, має спланувати рух таким чином, щоб із заданого початкового положення забезпечити досягнення мети маніпулятором і виконати додаткові умови на перенесення, наприклад оминути перешкоди на робочій сцені [4].

У багатьох випадках неможливо повністю запрограмувати рух маніпулятора. Прикладом є технологічна операція збирання. Неточне подання, встановлення деталей або їх дефектність може призвести до помилки у виконанні жорстко налаштованої програми. Щоправда, вказані особливості можна контролювати за допомогою засобу адаптації — системи технічного зору (СТЗ). Інформацію від СТЗ можна використати для корекції руху маніпулятора або для її прорахування. Таким чином, виникає ще один зовнішній контур керування, який забезпечує пристосування робота до змінюваних умов функціонування. Систему керування, що працює на вказаних принципах, можна віднести до рівня адаптивного керування [4].

На всіх рівнях керування відбувається взаємодія між роботом та людиною-оператором. В одних випадках допускається безпосередня участь людини у процесі керування. Тобто робот може функціонувати й у автономному режимі, але за певних умов роботи необхідна корекція рухів, супроводжувана аналізом ситуації, нарешті, необхідним є досвід людини — керування може бути передано оператору робота. Саме до цієї групи належать системи дистанційного керування для виконання різноманітних робіт в екстремальних умовах — за підвищеного рівня радіації, під водою, у космосі. Такі системи відносять до ергатичних (людино-машинних) маніпуляційних систем.

До другої групи належать автономні робототехнічні системи, здатні певний час після навчання функціонувати без участі оператора. При цьому останній виконує функції програмування попереднього навчання робота руху за заданою траєкторією або функції керування затискачем у реальному часі (у напівавтоматичних системах). В адаптивній системі керування робота оператора завершується до початку функціонування системи. Він визначає можливі варіанти змін умов роботи, установлює відповідні датчики й складає алгоритми адаптації, які дозволять системі функціонувати автоматично в межах визначених допусків на параметри.

Натомість під час роботи з інтелектуальними робототехнічними системами можливості оператора суттєво збільшуються. Оператор може використовувати наявний (і не тільки власний) досвід під час попереднього планування дії робота, застосовуючи програмне забезпечення. Таким чином, мова може йти про використання методів експертних систем. Експерти можуть залучатися для розв'язання завдань опису зовнішнього світу у випадках невизначеності, а також для завдань розпізнавання образів. Указані методи можуть застосовуватися і для керування роботами в реальному часі.

Використання методів експертних систем передбачає залучення зберігання знань у відповідності до моделей. Реалізація моделей визначатиме характер роботи системи керування робота як структурному, так і в процедурному плані.

3. Реалізація систем прийняття рішень роботів

З практичної точки зору, реалізація систем керування інтелектуальними роботами в багатьох аспектах виходить з робіт щодо проекту STRIPS (Stanford Research Institute Problem Solver) — системи прийняття рішень для замкненого світу робота, що взаємодіяла із виконавчою системою PlanEx (plan execution — виконання планів).

Для розв'язання практичних завдань прийняття рішень роботами необхідно автоматично формувати абстрактні простори різних рівнів з базового простору об'єктів та подій, у якому функціонує система.

У STRIPS-подібних системах [5] абстрактні простори визначаються рівнем деталізації умов застосування операторів. Такий підхід дозволяє:

- залишати незмінною модель світу — немає необхідності викреслювати з неї несуттєві (для даного рівня абстракції) деталі та не враховувати їх;
- задіяти незмінні операторні схеми.

Модель світу подається у вигляді набору правильно побудованих формул (ППФ) логіки предикатів першого порядку, що відображають собою

факти (наприклад, $ATR(a)$ — робот знаходиться у пункті a) та закони, наприклад:

$$(\forall y)(\forall z)\{[ATR(y) \& (y \neq z)] \supset \overline{ATR}(z)\},$$

тобто робот не може знаходитись одночасно в пунктах y та z .

Операторна схема визначається найменуванням, списком параметрів та описами у вигляді ППФ мови логіки предикатів першого порядку умов застосування дії та результату дії. Останній, у свою чергу, містить список викреслювань та список додавання. Оператори породжують різні моделі світу шляхом генерування нових фактів. Мета системи також подається у вигляді ППФ тієї ж логіки, тобто вона є бажаною для ППФ системи.

У системі STRIPS пошук починається зі спроби вивести цільову формулу G_0 з вихідної моделі світу M_0 . Для цього програма доведення теорем здійснює пошук протиріччя в множині диз'юнктивів $\{M_0 \cup \overline{G}_0\}$. Якщо таке протиріччя знайдене (вводиться порожній диз'юнктив), то вихідне завдання вирішується на цьому кроці тривіальним чином, тобто вихідна модель M_0 задовольняє меті G_0 . Якщо ж вказане протиріччя не погоджується, формується так званий незавершений вивід D_0 . Цей вивід подається набором диз'юнктивів, відповідних запереченню формули мети (у даному випадку \overline{G}_0), плюс усі їх похідні, якщо такі є, мінус ті диз'юнктиви, що виключаються завдяки застосуванню обмежувачих стратегій (наприклад, стратегії заміщення та оцінки предикатів).

Незавершений вивід D_0 приймається у якості різниці між M_0 та G_0 , що зв'язується з данім вузлом (M_0, G_0) . Далі відбувається пошук операторів, які підходять для зменшення отриманої різниці. Це такі оператори, дія яких на модель середовища дозволяє продовжувати доведення. У процесі пошуку найбільш придатного оператора значення його параметрів піддаються частковому або повному перебору.

Пошук оператора складається з двох кроків. На першому кроці складається упорядкований список операторів-кандидатів. Вибір операторів-кандидатів оснований на простому порівнянні предикатів з різницею D_0 з предикатами зі списку доповнення оператора. Другий крок полягає у застосуванні програми доведення теорем для визначення того, чи є у вказаному списку доповнень диз'юнктив, що могли б продовжити вивід після застосування цього оператора. Якщо цей крок видався успішним, оператор-кандидат із відповідними значеннями параметрів розглядається як придатний для зменшення різниці D_0 .

Коли таким чином оператор-кандидат знайдений, умови його застосування приймаються у якості нових підцілей системи. Нехай перед роботом поставлене завдання — перейти в точку b . Тоді $G_0 = ATR(b)$, й до

тих пір, поки робот не опиниться в точці b , спроби знайти необхідний доказ виявляться марними. Вочевидь, що окремий випадок $goto(m, b)$ оператора $goto(m, n)$ — перейти з пункту m у пункт n — підходить для зменшення різниці $D_0 = \bar{G}_0$, оскільки результат — $ATR(b)$ дозволяє продовжити вивід G_0 (у даному випадку — й закінчити його). Роль нової підцілі G_1 грає відповідна схема ППФ — умова застосування, скажімо, $ATR(m)$.

Система STRIPS чинить з підціллю G_1 так само, як і з метою. Вона знову використовує доведення теорем для виводу G_1 з M_0 . Тут важливі два випадки. Якщо система не знаходить доказ, вона аналогічним чином формує різницю D_1 між M_0 та G_1 і встановлює підцілі, відповідні формулам умов застосування відповідних операторів-кандидатів. Якщо теорема виводить G_1 з M_0 , то відповідний окремий випадок оператора використовується для перетворення моделі M_0 у нову модель M_1 . У вищезгаданому простому прикладі підціллю є $G_1 = ATR(m)$. Якщо $ATR(a) \in M_0$, окремий випадок G_1 , а саме $ATR(a)$, може бути виведений з M_0 . У цьому випадку оператор $goto(a, b)$ застосовується до моделі M_0 і формує модель M_1 , що включає $ATR(b)$. Потім STRIPS продовжує спроби вивести G_0 з M_1 . У нашому прикладі G_0 тривіально випливає з M_1 . Однак, якщо вивід G_0 не вдається, встановлюються відповідні підцілі й процедура починається знову [5]. Описаний вище підхід реалізує логічну модель подання знань.

Проблемне середовище системи планування містить модель світу, набір операторних схем та мету (або набір цілей) системи. У випадку постановки задачі для мобільного робота в замкненому просторі світ робота повинен відповідати схемі розташування об'єктів у цьому просторі, наприклад рис. 1.

4. Подання даних та знань у системах прийняття рішень

Реалізуючи на практиці вказану вище схему, можна описати всі її компоненти: модель світу, операторні схеми та мету системи. Зокрема, мовою

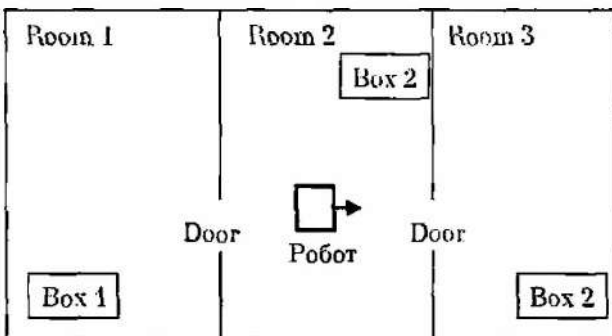


Рис. 1. Приклад схеми розташування об'єктів для системи планування робота

Prolog [6] модель світу, відповідна рис. 1, може бути описана набором тверджень:

```
is_a(room1,room,always)
is_a(room2,room,always)
is_a(room3,room,always)
is_a(door12,door,always)
is_a(door23,door,always)
is_a(box1,object,always)
is_a(box2,object,always)
is_a(box3,object,always)
connects(door12,room1,room2,always)
connects(door23,room2,room3,always)
stands(door12,opened,now)
stands(door23,opened,now)
is_in(box1,room1,now)
is_in(box2,room2,now)
is_in(box3,room3,now)
is_in(robot,room2,now)
is_at(robot,door12,now)
```

Неважко зрозуміти зміст бази даних. Вона формується відношеннями: is_a — приналежність об'єкта до класу, $connects$ — поєднання двох об'єктів третім (двох кімнат дверима), $stands$ — стан об'єкта (дверей), is_in — місцезнаходження об'єкта (ящика або робота в кімнаті), is_at — взаємне розташування одного об'єкта біля іншого. Слід зауважити, що такий опис повинен виключати протиріччя, для чого необхідно забезпечити спеціальну процедуру.

З точки зору реалізації бази знань робота, вказаний набір фактів поділяється на дві основні частини — статичну й динамічну. Факти статичної частини містять ключове слово «always» і залишаються незмінними в процесі роботи, факти ж динамічної частини можуть додаватися та вилучатися в процесі виконання програм і містять ключове слово «now».

Операторні схеми систем планування представляють описи дій, що можуть виконуватися в межах моделі світу. Для мобільного робота це можуть бути дії: відкрити (закрити) двері, перейти у кімнату N , перейти до об'єкта M , пересунути об'єкт P до об'єкта Q тощо. Кількість описів залежить від ступеня складності завдань робота.

Кожна операторна схема практично містить у собі:

- 1) твердження про результат реалізації операторної схеми — події, яку забезпечує схема;
- 2) тест виконання — перевірку відсутності реалізованого цільового факту;
- 3) тест валідності — перевірку придатності операторної схеми для забезпечення мети завдання;
- 4) список передумов — постановку (і виконання) умов, що передують виконанню дії;
- 5) список видалення — список подій, що застаріють на момент виконання схеми;

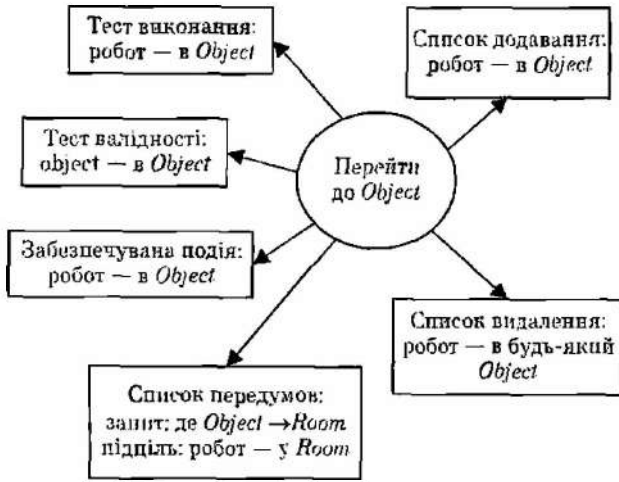


Рис. 2. Структура опису дії у системі планування мобільного робота

б) список додавання — список подій, що вносять-ся операторною схемою.

Схематично опис операторної схеми дії *Перейти до Object* представлений на рис. 2.

З точки зору баз знань, операторна схема подібного типу відповідає фреймовій моделі подання знань і фактично є фреймом-сценарієм. Особливістю фрейму-сценарію є визначеність послідовності виконання такого фрейму як набору полів фрейму — слотів (приклад наведено на рис. 3). Слід також вказати на наявність у кожного слота такого фрейму приєднаних функцій (у термінології фреймів — демонів), виконання яких означає заповнення відповідного слота фрейму значенням.

Таким чином, система прийняття рішень робота складатиметься з бази даних, набору фреймів-сценаріїв, які реалізують операторні схеми, механізму прийняття рішень та мети (або набору цілей) програми. Отримані в ході прийняття рішень результати можуть демонструватися за допомогою дво- або тривимірного комп'ютерного моделювання та бути виконані після відповідного тестування безпосередньо системою керування маніпулятором промислового робота. Схема підсистеми прийняття рішень зображена на рис. 4.

Назва фрейму дії	Перейти до <i>Object</i>
Тест виконання	Робот — в <i>Object</i>
Тест валідності	<i>Object</i> — <i>Object</i>
Забезпечувана подія	Робот — в <i>Object</i>
Виконання списку передумов	Запит: де <i>Object</i> → <i>Room</i> Підціль: робот — у <i>Room</i>
Список додавання	Робот — в <i>Object</i>
Список видалення	Робот — у будь-якій <i>Object</i>

Рис. 3. Приклад подання операторної схеми за допомогою фрейму

Мета системи формулюється у вигляді бажаного факту, до якого система повинна перейти, наприклад:

```
plan (is_at (box1, box2, now)),
```

де *plan* — предикат, що забезпечує досягнення поставленої мети — знаходження поруч об'єктів *box1* та *box2*.

Окрім вказаних компонентів проблемної області, у системі планування необхідно реалізувати власне вирішувач. Вирішувачем може бути предикат або набір предикатів — у випадку реалізації мовою Prolog [5], або набір відповідних функцій — у випадку реалізації на C++ (автором реалізовані обидві схеми).

Як вже вказувалося, завдання у системі формулюється в якості бажаного факту стану системи. Якщо такий факт не знаходиться у базі даних (тест виконання), викликається вирішувач, що реалізує схему пошуку рішення. Пошук рішення передбачає:

1) пошук предикатної схеми, яка відповідає поставленому завданню (у тому числі перевірку валідності);

2) виконання списку передумов;

3) виконання списку викреслення;

4) виконання списку додавання.

Слід урахувати, що при виконанні списку передумов (звичайно, якщо такий визначено) досягнення підцільей проводиться за допомогою рекурсивного виклику вирішувача, що вимагає чітких умов виходу з рекурсії.

Іншим рівнем розробки систем керування інтелектуальних роботів є розробка підсистем адаптації промислових роботів, зокрема систем технічного зору. Тут дослідників зустрічає досить великий набір проблем, пов'язаний із початковою обробкою інформації, її розпізнаванням та ідентифікацією. Однак,

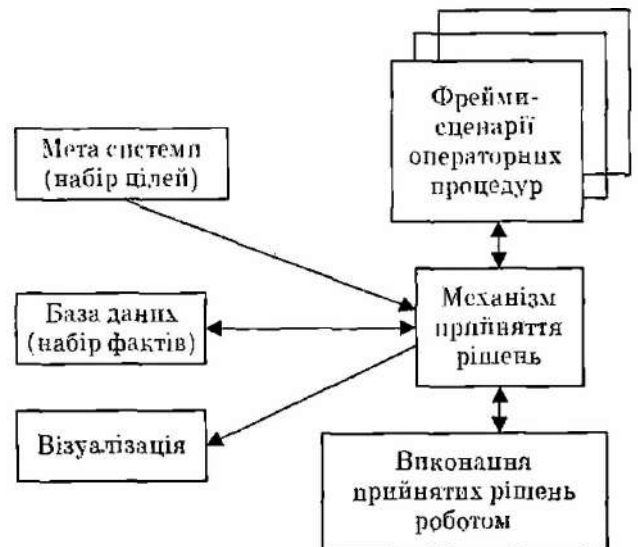


Рис. 4. Варіант реалізації підсистеми прийняття рішень інтелектуального робота

з точки зору побудови систем прийняття рішень, і тут необхідно виходити на рівень подання знань про стан робочої зони робота. При цьому слід виходити з наявності певних ознак об'єктів у робочій зоні, наприклад:

- належність об'єкта робочої зони до певного класу;
- стан об'єкта;
- абсолютні або відносні координати об'єкта;
- рівень пріоритету об'єкта;
- можливість перенесення системи координат до певного об'єкта.

Таким чином, із кожним об'єктом можна пов'язати фрейм із описом, що має вводити до бази даних системи прийняття рішень та використовується у стандартному режимі.

Як висновок, з точки зору реалізації системи прийняття рішень, пропонується:

- у загальній схемі прийняття рішень інтелектуального робота користуватися логічною моделлю подання знань;
- для опису операторних процедур системи прийняття рішень формувати структуру, відповідні фреймовій моделі;
- формування бази даних про робоче середовище робота вести також на основі фреймової моделі;
- вести проектування програмного забезпечення засобами високого рівня, зокрема за допомогою мови програмування Prolog.

5. Перспективні напрямки досліджень

До перспективних напрямків розвитку описаної вище системи прийняття рішень (планування) інтелектуального робота слід віднести, по-перше, розширення проблемної області. Таке розширення призведе до зростання операторних схем та їх

ускладнення. У випадках, коли одна мета може бути досягнута декількома шляхами, виникне необхідність попередньої оцінки складності операторної схеми. При цьому кожній схемі з набору, що забезпечує однакові цілі, відповідатиме певний коефіцієнт, а сам набір буде нечіткою множиною. По-друге, функціонування роботів у реальному часі вимагатиме поліпшення процедур знаходження адекватних операторних схем, у тому числі з урахуванням алгоритмів «відкату», відновлення попереднього ступа предметної області. По-третє, розширення проблемної області визначатиметься розширенням кола операцій, виконуваних роботом, а послідовність виконання операцій (технологічний процес) визначатиме стратегію дій робота.

У межах тематики статті ведеться розробка системи прийняття рішень промислового робота РМ-01 на операціях транспортування та збирання. Керування роботом здійснюється від ПЕОМ, ідентифікація об'єктів у робочій зоні проводиться системою технічного зору робота.

Список літератури: 1. Воробьев Е. И., Попов С. А., Шевелева Г. И. Механика промышленных роботов: В 3 кн. Кн. 1: Кинематика и динамика. — М.: Высшая школа, 1988. — 304 с. 2. Гаврилова Т. А., Хоросhevский В. Ф. Базы знаний интеллектуальных систем. — СПб.: Питер, 2001. — 384 с. 3. Вакано Э. Введение в робототехнику. — М.: Мир, 1988. — 334 с. 4. Зенкевич С. Л., Юценко А. С. Основы управления манипуляционными роботами. — М.: Изд. во МГТУ, 2004. — 480 с. 5. Люгер Дж. Ф. Искусственный интеллект: стратегии и методы решения сложных проблем. — М.: ИД «Вильямс», 2003. — 894 с. 6. Малласс Дж. Реляционный язык Пролог и его применение. — М.: Наука, 1990. — 464 с.

Надійшла до редакції 08.09.2006