

Gabriel Grzegorz KOST

Politechnika Śląska

JĘZYK TEKSTOWEGO PROGRAMOWANIA OFF-LINE ROBOTÓW PRZEMYSŁOWYCH

Streszczenie. Artykuł przedstawia język tekstowego programowania robotów przemysłowych pod nazwą SPMT-R/PC. Omawia przeznaczenie języka, sposób formułowania w nim zadań robotowych i jego gramatykę.

1. Wstęp

Słowane obecnie coraz powszechniej obrabiarkowe elastyczne systemy produkcyjne (ESP) oraz ich ogniwa w postaci elastycznych gniazd wytwórczych (EGW), konfigurowane najczęściej z obrabiarek sterowanych numerycznie (OSN), bardzo często wymagają zastosowania robotów przemysłowych (RP) do transportu materiałów między stanowiskami obróbki. Od RP wykorzystywanych w tych systemach wymaga się odpowiednio dużej elastyczności w sensie możliwości współdziałania z OSN zastosowanymi w EGW czy ESP, jak i w sensie łatwości ich programowania dla zmieniających się warunków technologicznych. Właśnie ta dążność do łatwości programowania RP prowadząca do wzrostu wydajności ESP wskutek skrócenia czasów ich przestoju na programowanie, leży u podstaw opracowywania systemów tekstowego (komputerowego) programowania RP wykorzystujących specjalne języki programowania wyższego rzędu, pozwalające na programowanie off-line RP.

Posiadamy obecnie w kraju szereg systemów komputerowego programowania OSN, ale nie posiadamy takiego systemu dla produkowanych w kraju RP. Czynione w tym kierunku działania zacierają z jednej strony do rozwiązań bardziej ogólnych i uniwersalnych, tworzonych w ośrodkach naukowych i bazujących na zastosowaniach sztucznej inteligencji, z drugiej do rozwiązań prostszych, pozwalających na programowanie tylko elementarnych czynności - ruchów RP.

Pierwsze z nich wymagają interakcji między elementami środowiska roboczego (sceny) RP, realizowanej poprzez "zmysły robota" (czujniki taktylne, wizyjne itp.) zbierające i obrabiające informacje o środowisku, tworząc jego model, oraz realizujące sprzężenia zwrotne.

Drugie zaś w oparciu o wcześniej przygotowany program, który określa warunki przemieszczenia ogniwa roboczego RP w przestrzeni roboczej opisanej układem współrzędnych kartezjańskich, przetwarzają go powodując wygenerowanie tzw. układu współrzędnych ruchu robota, które przekazane do układu sterowania RP (USD) interpretowane są jako współrzędne ruchu każdego ogniwa łańcucha kinematycznego RP. W rezultacie robot osiąga zadane położenie w przestrzeni. Ta grupa systemów tekstowego programowania RP, w postaci

systemów wąsko wyspecjalizowanych, przeznaczonych dla konkretnych zastosowań przemysłowych i jednego typu RP, jak dotąd jedyna znalazła szersze zastosowanie przemysłowe. Twórcami ich są bowiem firmy produkujące RP, dla których są one dodatkowym wyposażeniem ułatwiającym jego programowanie (MAL f-my OLIVETTI dla robota SUPERSIGMA, HELP11 f-my DEA dla robota PRAGMA 3000a, ALFA f-my GTEL dla robota VERSATRAN [1, 2, 3]).

Ze względu na duże zalety programowania interaktywnego, które wiąże się, niestety, z wielką pracochłonnością i wysokimi kosztami opracowań, oraz na efektywność zastosowań systemów drugiej grupy, część twórców skłania się do opracowania systemów programowania RP pośredniego poziomu, które pozwoliłyby programować ruchy RP wykorzystując do tego celu uproszczony model środowiska roboczego. Systemy te pozwalają na niepełne "programowanie inteligentne", które ogranicza się najczęściej do gromadzenia określonych informacji o środowisku (pomiar odległości końcówki od przeszkody, wykorzystanie resolwerów, chwytaków mogących rozpoznać ułożenie przedmiotu, stosowanie systemów wizyjnych do rozpoznawania warunków sceny itd.). Do tej grupy systemów zaliczyć można np.: RAPT, LAMA, LAMA-S, AUTOPASS i inne [1, 2, 3].

Przy tworzeniu języków programowania RP w ramach danego systemu istnieją dwa równoległe podejścia [1, 2, 3]:

- wykorzystanie istniejącego już języka programowania maszyn cyfrowych (np. FORTRAN, PASCAL, ADA) lub OSN (APT), polegające na dodaniu do standardu języka nowych specjalizowanych procedur pozwalających na sterowanie robotem;
- opracowanie nowego specjalizowanego języka sterowania robotem i przetwarzania informacji pochodzących od RP, którego zaletą jest lepsze dostosowanie go dla potrzeb użytkownika i brak procedur językowych, które nie są wykorzystywane w programowaniu robota.

Jednym z języków programowania off-line RP pod nazwą SPMT-R/PC jest przedstawiony poniżej język, który powstał w Instytucie Budowy Maszyn Politechniki Śląskiej w Gliwicach.

2. Język programowania robotów SPMT-R/PC

Język ten pozwala na symboliczny zapis czynności robota zajmującego się obsługą OSN obróbki ubytkowej zgrupowanych w elastyczne systemy wytwórcze, przy czym nie wyklucza się możliwości rozbudowy i wykorzystania go do programowania RP wykorzystywanych do innych zadań wytwórczych, np. w montażu. Tworzą go kody będące mnemotechnicznymi skrótami słów języka polskiego.

2.1. Zasady definiowania języka SPMT-R/PC

Przy opisie języka SPMT-R/PC przyjęto pewne reguły, z których wyjątki przedstawiono poniżej [4]:

- a) litery duże i znaki specjalne (znaki interpunkcyjne, operatory, nawiasy itp.) użyte we wzorach definicyjnych reprezentują same siebie;

- b) małe litery oznaczają zastępowane przez nie obiekty, którymi mogą być oddzielnie definiowane konstrukcje składniowe lub liczby;
- c) nawiasy kwadratowe "[]" zawierają jednostki składniowe opcjonalna, które nie muszą wystąpić,
- d) nawiasy klamrowe "()" zawierają składniki wariantowe, z których jeden musi wystąpić;
- e) wielokropek "... " wskazuje, że poprzedzająca go jednostka składniowa może być powtórzona wielokrotnie;
- f) znak "&" oznacza kontynuację, a znak "!" komentarz.

Język systemu SPMT-R/PC dopuszcza stosowanie w programie źródłowym następujących znaków:

- litery alfabetu łacińskiego od A do Z i litery małe z tego zakresu, które system traktuje jak litery duże,
- cyfry od 0 do 9,
- znaki specjalne mieszczące się w zakresie od 32 do 127 kodu ASCII,
- NL, CR znak "nowa linia", "powrót karetki".

2.2. Program źródłowy

Zapisane symbolicznie zadanie RP w języku SPMT-R/PC tworzy dane wejściowe dla systemu SPMT-R/PC programowania RP, noszące nazwę programu źródłowego. Program ten jest zbudowany z kilku części, z których każda zawiera inną grupę danych:

- dane nagłówkowe identyfikujące program,
- dane geometryczne - deklaracje geometrii przestrzennej zawierające definicje obiektów geometrycznych wykorzystywanych w programie robota (część geometryczna),
- dane technologiczne dotyczące opisu pracy robota i jego US, na które składają się (część technologiczna):
 - dane początkowe (deklaracje):
 - dane o zadaniach technologicznych wykonywanych przez robot i US.

2.2.1. Dane nagłówkowe

Składają się na nie następujące informacje:

Program nr.....: numer programu, ...
 Scena.....: nazwa sceny, ...
 Nazwa części.....: nazwa części, ...
 Rysunek nr.....: numer rysunku części, ...
 Ciężar.....: ciężar części, ...
 Data opracowania programu.....: data opracowania programu, ...
 Opracował.....: dane opracowującego program, ...
 Manipulator.....: nazwa (typ) robota dla
 którego opracowano program, ...;

Adresową część wiersza można skracać do jednej litery (podkreślonej w

opisie postaci nagłówka).

2.2.2. Dane geometryczne

Geometryczna część programu źródłowego przeznaczona jest do definiowania tzw. pozycji referencyjnych sceny (PR), wykorzystywanych przy opisie jego zadań technologicznych. PR określają położenie przedmiotu manipulowanego, obiektów sceny lub części manipulacyjnej samego robota. Definiowanie ich dokonywane jest w układzie kartezjańskim nazywanym układem podstawowym (sceny). Są one określane przez definiowanie tzw. elementów geometrycznych (EG).

W aktualnej wersji systemu wyróżnia się elementy geometryczne typu:

- punkt - P, określany na płaszczyźni
- pozycja - PZ, określa położenie PR w przestrzeni trójwymiarowej,
- układ wtórny U - układ prostokątny związany z dowolnym obiektem sceny, pozwala na definiowanie w nim EG, które transponowane są do układu podstawowego.

2.2.3 Opis pracy robota

Przewidziano dwie grupy danych opisujących pracę RP:

- dane początkowe (deklaracje),
- dane o zadaniach technologicznych realizowanych przez robot (instrukcje czynne).

2.2.3.1 Dane początkowe

Zawierają informacje dotyczące całego programowanego zadania technologicznego wykonywanego przez RP. Tworzą je:

- instrukcja T: jest pierwszą instrukcją części technologicznej programu źródłowego. Od niej procesor technologiczny rozpoczyna analizę programu źródłowego w jego części opisującej zadanie RP,
- instrukcja B: położenie przestrzeni roboczej RP w środowisku jego pracy definiuje się przez określenie tzw. układu programowania robota względem układu podstawowego. Położenie to musi być ściśle określone przy opracowywaniu programów wykonawczych robota,
- instrukcja ADA: służy do definiowania tzw. adaptera (elementu służącego do mocowania chwytaków, narzędzi itp.) powodującego przedłużenie łańcucha kinematycznego robota,
- instrukcja CHW: definiuje w programie źródłowym chwytak (pojedynczy lub podwójny),
- instrukcja PM: służy do definiowania przedmiotu manipulowanego tzn. jego parametrów geometrycznych (konstrukcyjnych), pozycji chwytu oraz pozycji odłożenia.

2.2.3.2. Instrukcje czynne robota

Instrukcje te zawierają całość informacji dotyczącej sposobu wykonywania przez robot programowanego zadania technologicznego. Są to instrukcje ruchu samego robota, czyli jego przejazdów (pozycjonowań) do zadanych PR, jak też instrukcje opisujące działania US:

- instrukcja POZ-"POZYCJONOWANIE": służy do realizacji pozycjonowań, tzn. generowania ruchów robota między dowolnymi pozycjami PZ. Może być realizowane jako pozycjonowanie: liniowe, zgrubne lub dokładne,
- instrukcja WYJ-"WYJŚCIE": sprawdza stan wyjścia lub wskaźnika US, które mogą być związane informatycznie z samym robotem i dowolnym obiektem sceny (OSN, magazyn we/wy itp.),
- instrukcja CZW/B-"CZEKANIE": realizuje czekanie robota na spełnienie warunku, jakim jest stan wejścia (wskaźnika), po spełnieniu którego następuje dalsza realizacja programu - " CZW ", lub czekanie bezwarunkowe w określonym czasie - " CZB ",
- instrukcja SKW/B-"SKOK": realizuje skok warunkowy wg. wejścia lub wskaźnika oraz skok bezwarunkowy do instrukcji programu źródłowego o określonym numerze. Powoduje ona przerwanie sekwencyjnego ciągu realizowania instrukcji programu, następnie przeszukiwanie programu źródłowego, aż do napotkania instrukcji o zadany numerze. SKW realizuje skok do instrukcji wg. warunku, jakim jest odpowiedni stan wejścia (wejścia), zaś SKB realizuje skok bezwarunkowy, tzn. skok nieobwarowany żadnym warunkiem,
- instrukcja PWT-"POWTÓRZENIE": służy do wyodrębnienia tych części programu źródłowego, które mają być powtarzane - czyli do definiowania pętli ograniczonej instrukcjami "Powtórzenie" i "Koniec powtarzania",
- instrukcja SKP-"SKOK PROGRAMU": realizuje skoki do jednego z programów znajdujących się w pamięci US, zawierających typowe sekwencje ruchów RP (pobranie chwytaka, detalu itp.),
- instrukcja E-"KONIEC": kończy ona zapis programu źródłowego oraz program pracy robota; generuje instrukcję US robota "Koniec programu". Musi wystąpić na końcu programu źródłowego,
- instrukcja WOT/PM-"UAKTYWNIENIE": powoduje przedłużenie (założenie) lub skrócenie (odłożenie) łańcucha kinematycznego robota o elementy określone w instrukcji (adapter, chwytak lub przedmiot manipulowany) oraz zadziałanie danego ogniwa w tym łańcuchu.

3. Wyciąg ze składni języka SPMT-R/PC

Pp = współrzędne punktu na płaszczyźnie;

PZpz = współrzędne kartezjańskie, orientacja przestrzenna;

Uu = współrzędne kartezjańskie, orientacja przestrzenna

Bb = współrzędne kartezjańskie, orientacja przestrzenna

T [!komentarz]

ADAn = { identyfikator [, długość elementu] [, promień obrotu] [, masa] &

```

<[,identyfikator elementu współpracującego]...&
<położenie i orientacja elementu chwytanego(roboczego)...)
CHWn = <identyfikator [,długość ramion][,promień obrotu][,masa] &
[,siła działań<[identyfikator el. współpracującego]...&
<położenie i orientacja elementu chwytanego(roboczego)...)
PMpm = <identyfikator [,masa][,promień walca opisującego &
przedmiot][,szerokość przedmiotu w płaszczyźnie &
uchwycenia][,pozycja uchwycenia]...<[pozycja &
odłożenia]...)
POZpoz = <<(rodzaj pozycjonowania <[pozycja][,pozycja uchwycenia] &
[,pozycja odłożenia]...<[czas pozycjonowania][,szybkość] &
[[,czekanie warunkowe][,czekanie bezwarunkowe]])... &
<pozycjonowanie)...)
WYJ = <[stan wyjścia][,stan wejścia][,stan wskaźnika][,czekanie &
w czasie]...)
CZ([W][B]) = <[[stan wejścia][,stan wyjścia][,stan wskaźnika][czas]]
SK([W][B]) = <[[wyjście][,wskaźnik][,nr. instrukcji skoku] &
[ nr. instrukcji skoku ]]
PWT = ilość powtórzeń (iteracji), ostatnia instrukcja w pętli &
powtórzeń
SKP = nr. programu do którego na zostać zrealizowany skok
E [!komentarz]
W([OT][PM]) = <<[[założenie][,odłożenie]]<[adaptera][,chwytaka] [,czas &
zadziałania chwytaka<[nr. elementu roboczego chwytaka &
adaptera][,([szczęki zwarte][,szczęki rozwarte]])]... &
[,przedmiot manipulowany]>

```

8 Zakończenie

Opisany język SPMT-R/PC został zastosowany w komputerowym systemie programowania robotów przemysłowych o tej samej nazwie. W aktualnej wersji system ten jest przeznaczony do programowania robotów IRb pracujących w elastycznych ESP, z możliwością zastosowania go do innych typów RP w innych zastosowaniach wytwórczych. Jest to możliwe dzięki dużej uniwersalności opracowanego języka. Jego twórcy w kolejnych etapach rozwoju systemu SPMT-R/PC przewidują rozwinięcie go o cały szereg nowych EG ułatwiających określenie tak PR, jak i obiektów sceny, z którymi robot współpracuje (np. linia, okrąg, kula, walec, prostopadłościan, zbiór punktów, zbiór PR itp.).

LITERATURA

- [1] Blume Ch., Dillman R.: Struktur und Programmierung von Industrierobotern. VDI-2 nr. 6, 1980, s. 231-239.
- [2] Zieliński C.: Roboty cz. 3,4, Informatyka nr. 8,9, 1984.

- [3] Blume Ch., Jakob W.: Programming Languages for Industrial Robots. Springer-Verlag, London, 1986.
- [4] Wójcikowski J. i inni: Instrukcja użytkownika mikrokomputerowego systemu SPMT-R/PC programowania robotów przemysłowych. IBM Gliwice, 1987, nie publikowane.

Recenzent: Prof.dr h.inż.W.Niewczas

Wpłynęło do Redakcji do 1990-04-30.

OFF-LINE PROGRAMMING LANGUAGE OF INDUSTRIAL ROBOT

Summary

The paper presents SPMT-R/PC off-line programming language of industrial robots. It discusses the language destination, its basic structures, types of a geometrically-interpreted objects, motoric and control commands, syntax definition.

АВТОНОМНЫЙ ЯЗЫК ПРОГРАММИРОВАНИЯ ДЛЯ ПРОМЫШЛЕННЫХ РОБОТОВ

Резюме

В статье представлено сокращенное описание языка SPMT-R/PC текстового программирования промышленных роботов с помощью ЭВМ. Рассмотрено предназначение языка, его структура, способ представления геометрических объектов, инструкции движения и управления программой.