

Zastosowanie robotów typu Fanuc w aplikacjach  
(Optymalizacja oprogramowania)

Paletyzacja



Przedstawiony poniżej program został zrealizowany na jednym z robotów typu Fanuc ze sterowaniem RJ2 w fabryce Opla w Niemczech.

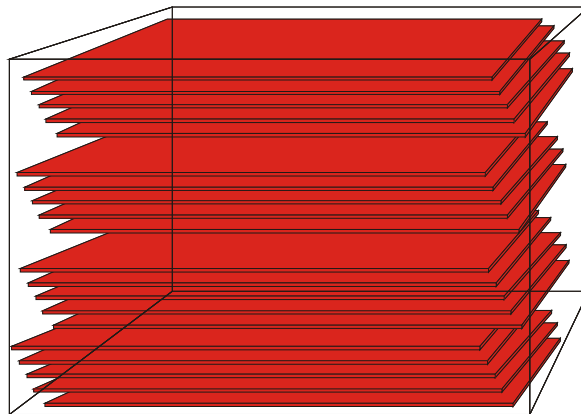
W obecnych czasach od robotów przemysłowych pracujących w różnych branżach przemysłu oczekuje się, aby były: uniwersalne, łatwo przezbrajane oraz łatwo programowalne. Większość różnorodnych zadań (programów) można zrealizować na wiele sposobów.

Przykładowy program został opracowany dla specjalnego rozwiązania na stanowisku zrobotyzowanym. Konstruktorzy zaprojektowali dla naszego robota osiem różnych stacji. Robot przenosi część nadwozia samochodu w dwóch typach. Dla każdego typu ma stację do pobierania części, dwie stacje do odkładania oraz jedną stację, która służy jako bufor dla utrzymania ciągłości produkcji. Do tego bufora robot może odkładać blachy lub je z niego pobierać w zależności od warunków zewnętrznego sterowania. Bufor składa się z dwudziestu pól odkładczych, które pogrupowane są: 4 x 5 pól. Każde pole przesunięte jest dodatkowo w osi Z i osi X tej stacji o stałą wartość. Każde z 4 grup pól odkładczych przesunięte jest również w osi Z o stałą wartość.

Patrząc na tak sformułowane zadanie, pierwszym rozwiązaniem, jakie przychodzi nam na myśl jest napisanie dla każdego pola osobnego programu do odkładania i pobierania części. Jest to oczywiście rozwiązanie problemu, lecz należy wziąć pod uwagę następujące okoliczności:

- Musimy dla każdego pola osobno nauczyć wszystkie punkty trajektorii,
- Stworzyć osobny program logiczny do selekcji pola oraz wyboru funkcji: odkładaj lub pobierz.

W takiej sytuacji należy sobie zadać pytanie:, co się stanie, jeżeli robot będzie miał w trakcie normalnej produkcji zderzenie, chwytak straci swoje wymiary lub konstruktor stwierdzi, że trzeba tutaj coś zmienić. Koszty przestoju z powodu poprawy 40 programów tylko dla jednej stacji będą niewyobrażalne.



Rys. 1 Szkic stacji (bufor)

Poniższy program pozwala w łatwy i przyjemny sposób wybrnąć z takiej opresji. W pierwszej kolejności odpowiednio zmierzono opracowywaną stację. Od tej pory robot będzie poruszał się dokładnie we współrzędnych pomierzonej stacji. Następnie zapamiętano tylko 2 pozycje: pozycja nad stacją oraz pozycja pierwszej części (półka nr 1 na samym dole). Kolejnym krokiem było zmierzenie odpowiednich przesunięć pomiędzy poszczególnymi półkami oraz zapisanie ich w odpowiednich rejestrach pozycji. Następnie wykonano kilka obliczeń matematycznych:

```
L PR[7:Pozycja ponad stacją] 1000mm/sec FINE

R[12:Licznik stacji]= R[12: Licznik stacji] + 1
PR[6:Pozycja aktualna]= PR[9:1.Część]

IF R[12: Licznik stacji] > 1 AND R[12: Licznik stacji] < 21 , JMP LBL[1]
R[12: Licznik stacji]= 1
JMP LBL[2]

LBL[1]
R[9:Rejestr pomocniczy]= R[12: Licznik stacji] MOD 5
IF R[9: Rejestr pomocniczy] <> 0 , JMP LBL[77]
R[9: Rejestr pomocniczy]= 5
LBL[77]
R[10: Rejestr pomocniczy 2]= R[9: Rejestr pomocniczy] * R[6:Przesunięcie X]
R[11: Rejestr pomocniczy.3]= R[10: Rejestr pomocniczy.2] - R[6: Przesunięcie X]
PR[6,1: Pozycja aktualna]= PR[6,1: Pozycja aktualna] + R[11: Rejestr pomocniczy.3]
```

Powyższa część programu spełnia następujące zadanie:

1. Sprowadza robota do pozycji bezpośrednio nad stacją

**L PR[7:Pozycja ponad stacją] 1000mm/sec FINE**

2. Inkrementuje licznik obecności części

**R[12:Licznik stacji]= R[12: Licznik stacji] + 1**

3. Przypisuje do rejestru pozycji współrzędne najniższej półki jako wartość aktualną (na podstawie tej wartości obliczane jest odpowiednie przesunięcie)

**PR[6:Pozycja aktualna]= PR[9:1.Część]**

4. Sprawdza czy bufor przypadkiem nie jest już pełny (zabezpieczenie w przypadku awarii czujników itp.)

**IF R[12: Licznik stacji] > 1 AND R[12: Licznik stacji] < 21 , JMP LBL[1]**

5. Sprawdza numer wolnej półki do odłożenia części

**R[9:Rejestr pomocniczy]= R[12: Licznik stacji] MOD 5**

**IF R[9: Rejestr pomocniczy] <> 0 , JMP LBL[77]**

**R[9: Rejestr pomocniczy]= 5**

6. Oblicza odpowiednie przesunięcie w osi X (stacji) oraz zapamiętuje obliczoną pozycję jako pozycję aktualną

**R[10: Rejestr pomocniczy 2]= R[9: Rejestr pomocniczy] \* R[6:Przesunięcie X]**

**R[11: Rejestr pomocniczy.3]= R[10: Rejestr pomocniczy.2] - R[6: Przesunięcie X]**

**PR[6,1: Pozycja aktualna]= PR[6,1: Pozycja aktualna] + R[11: Rejestr pomocniczy.3]**

Następnie należy dokonać podobnych obliczeń dla pozostałych dwóch przesunięć. Zapisać obliczenia jako aktualizację pozycji do osiągnięcia w odpowiednim rejestrze pozycji oraz zadać robotowi dane koordynaty w parametrach ruchu do punktu odłożenia części

**L PR[6:Aktualna pozycja] 600mm/sec FINE**

Podobny program tworzymy dla pobierania części z jedną modyfikacją – dekrementacja licznika.

Powyższe rozwiązanie problemu zostało przedstawione za pomocą kilku rejestrów pozycji i danych pochodzących z pomiarów bufora. Przeprogramowanie stacji zajmuje w tym przypadku około 30 minut i znacznie ułatwiło pracę obsługi tego stanowiska zrobotyzowanego. Przykład pokazuje, że w pierwszej kolejności trzeba się zastanowić, co należy zrobić a w drugiej jak to zrobić optymalnie dla siebie i użytkownika końcowego.

Bartłomiej Siuda