

Aldona KUŚMIŃSKA-FIJAŁKOWSKA¹
Zbigniew ŁUKASIK²

STRUKTURA, TWORZENIE I TESTOWANIE PROGRAMU ZINTEGROWANEGO ZROBOTYZOWANEGO SYSTEMU FANUC S- 420F

W artykule przedstawiono zintegrowany zrobotyzowany system w oparciu o robota S-420F , który daje możliwość przećwiczenia praktycznych aspektów programowania ruchów robota wykorzystywanego w procesach przemysłowych jak również transportowych.

PROGRAMMING INTEGRATED ROBOTISED SYSTEM FANUC S- 420F

Integrated robotised system was introduced in the article in the support about the robot S-420F , which the possibility of practicing gives the practical aspects of programming of the movements of the robot used in industrial processes as also forwarding.

1. WSTĘP

Rozwój robotów przemysłowych, ich popularność oraz zwiększające się możliwości, elastyczność, różnorodność i szeroka gama możliwych zastosowań sprawiły, że roboty są coraz częściej wybierane jako podstawowe urządzenia systemów transportowych w przemyśle[7]. Coraz bogatsza ich gama wymaga odpowiedniego ich doboru do wykonywanego przez nie zadania. Zadania te spoczywają na Integratorach. Są nimi ludzie, którzy posiadają wiedzę dotyczącą zagadnień automatyki przemysłowej, poszerzoną o dogłębną znajomość możliwości robotów przemysłowych, umiejętności ich doboru do danej aplikacji pod względem konstrukcji, zasięgu i transportowanego ciężaru.

Integrator powinien znać specyfikę budowy robota przemysłowego, przeznaczenie jego podzespołów oraz język programowania wraz z wbudowanymi procedurami i oferowanymi opcjami charakterystycznymi dla danego robota, by móc stworzyć efektywne i bezpieczne oprogramowanie robocze[2].

Integracja Zrobotyzowanego Systemu Przemysłowego polega na określeniu zadań budowanego systemu, dobraniu odpowiedniego ramienia roboczego, narzędzia roboczego,

¹ Politechnika Radomska, Wydział Transportu i Elektrotechniki; 26-600 Radom; ul. Malczewskiego 29.
tel: + 48 48 361-77-16, Fax: + 48 48 361-77-42, e-mail: a.kusminska@pr.radom.pl

² Politechnika Radomska, Wydział Transportu i Elektrotechniki; 26-600 Radom; ul. Malczewskiego 29.
tel: + 48 48 361-70-30, Fax: + 48 48 361-77-42, e-mail: z.lukasik@pr.radom.pl

układów bezpieczeństwa i innych elementów stanowiska oraz na stworzeniu oprogramowania pozwalającego na realizację wszelkich zakładanych zadań. Istotne etapy integracji to: konfigurowanie systemu, układów współrzędnych narzędzia i użytkownika oraz tworzenie oprogramowania wykonującego zadania przewidziane dla ramienia roboczego.

2. ZINTEGROWANY ZROBOTYZOWANY SYSTEM FANUC S- 420F

Prezentowany Zintegrowany Zrobotyzowany System (Rys. 1. 1) składał się z dwóch podsystemów :

- wykonawczego, w skład którego wchodzi: Robot FANUC S- 420F wyposażony w narzędzie robocze (pisak) oraz mobilnej tablicy o regulowanym kącie pochylecia.
- bezpieczeństwa, składającego się z: bariery, skanera (SICK S3000), krańcówki mechanicznej, czyli elementów gwarantujących bezpieczne użytkowanie systemu oraz zatrzymanie pracy robota i całego systemu w sytuacji zagrożenia.



Rys. 1. 1. Zintegrowany Zrobotyzowany System a) Stanowisko robocze; b) Kontroler R-H, Teach Pendant; c) Skaner SICK S3000, (Laboratorium Zakładu Automatykacji Procesów na Wydziale Transportu i Elektrotechniki Politechniki Radomskiej)

3. PROGRAMOWANIE ROBOTA FANUC S-420F

Najistotniejszym etapem integracji robota w systemie przemysłowym lub transportowym jest tworzenie oprogramowania użytkowego realizującego wszelkie zadania, dla których dany system został zaprojektowany. Każdy program powinien składać się z trzech wyraźnie wydzielonych części[3]:

1. Wstępnej. W tej części program powinien sprawdzać i ustawiać wszelkie zmienne systemowe niezbędne do prawidłowego wykonania roboczej części programu. Tu powinny być umieszczane deklaracje, definicje i obliczenia wstępnych wartości zmiennych programowych oraz kodu, który ma za zadanie sprawdzenie poprawności ustawiania wszelkich zewnętrznych urządzeń systemu takich jak: osłony i obrabiane komponenty oraz sprawdzenie aktualnej pozycji ramienia robota jako wyjściowej dla kolejnych jego ruchów. W przypadku niezgodności program powinien to zakomunikować operatorowi. Wykonywanie części roboczej programu powinno odbywać się tylko w przypadku poprawnej i bezpiecznej konfiguracji warunków wstępnych. Pierwszym ruchem ramienia roboczego w dobrze napisanym programie powinno być ustawienie go w pozycji wyjściowej zwanej HOME.

2. Roboczej. Część roboczą tworzy seria potrójnych faz ruchu, z których buduje się kolejne etapy wykonywanego procesu. Są to odpowiednio:

- Faza ustawiania wstępnego parametrów narzędzia roboczego i przemieszczenia go z pozycji bieżącej do pozycji roboczej
- Faza wykonywania zakładanych czynności przez ustawianie odpowiednich stanów narzędzia roboczego oraz poruszanie nim po zadanej trajektorii dla uzyskania zakładanego efektu.
- Faza ustawiania końcowego parametrów narzędzia i przemieszczenia go do pozycji bezpiecznej.

3. Końcowej. Część końcowa ma za zadanie ustawić wszelkie urządzenia zewnętrzne by można było bezproblemowo wykonać ponownie część wstępną programu. Program powinien przenieść ramię robocze do pozycji wyjściowej, spoczynkowej zwanej HOME i ustawić stan wyjściowy jego narzędzia roboczego tak, by umożliwić ponowne, bezpieczne wykonanie programu.

4. PORUSZANIE ROBOTEM FANUC S-420F

Poruszanie ramieniem robota jest jedną z podstawowych czynności w trakcie codziennej obsługi zrobotyzowanego systemu, uczenia punktów i ścieżek programu, serwisowaniu robota oraz ustawianiu ramienia w określonych punktach koniecznych do prowadzenia bezpiecznych prac naprawczych lub uruchamiania programu w trybie automatycznym.

W systemie KAREL ruch ramienia robota może być wywołany przez kilka urządzeń wchodzących w jego skład [6]:

- Teach Pendant – gdy potrzeba jest ręcznego sterowania ruchami ramienia przy uczeniu lub serwisowaniu, a także testowania punktów, ścieżek punktów oraz programów[2].
Rys. 1.1
- Program – gdy kontroler pracuje w trybie automatycznym, program, który jest aktualnie uruchomiony ma kontrolę nad ruchami robota i wykonuje instrukcję ruchu.
- Panel operatora – przez wciśnięcie odpowiedniego przycisku, a przy odpowiedniej konfiguracji zmiennych systemowych nawet załączenie zasilania może spowodować uruchomienie odpowiedniego programu, który może zawierać instrukcje ruchu.

Należy pamiętać, że tylko w układzie JOINT możliwe jest poruszanie robotem gdy robot nie jest skalibrowany lub przekroczone zostały limity ruchu osi[5]. Dla komfortowego poruszania ramieniem robota istotna jest też możliwość regulacji prędkości ruchu oraz

możliwość podglądu bieżącej pozycji ramienia. Podczas poruszania ramieniem robota należy:

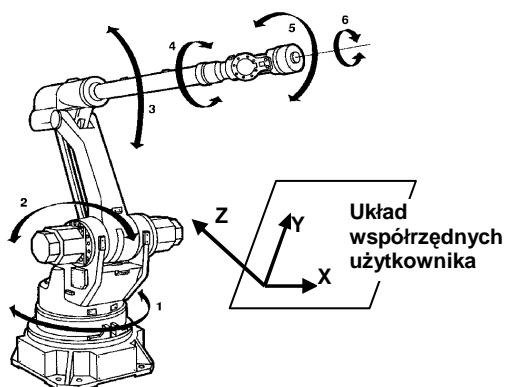
- przewidywać wszelkie jego ruchy zanim się je wykona pamiętając przy tym o układzie współrzędnych w jakim robot aktualnie pracuje,
- obserwować strefę roboczą ramienia przed i w trakcie ruchu sprawdzając, czy nie spowoduje się kolizji z innym elementem systemu.

4.1. Układy współrzędnych

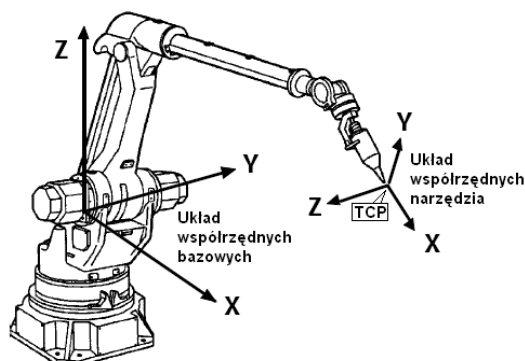
Robot FANUC S-420F z kontrolerem R-H umożliwia programiście używanie kilku typów układów współrzędnych co ułatwia programowanie, wielokrotne stosowanie kodu oraz łatwe i szybkie konfigurowanie narzędzi w razie konieczności ich wymiany bez wpływu na wcześniej zapamiętane punkty programowe. Ramię robocze może być przemieszczane przez zmianę położenia wybranej osi i jednoczesną zmianę położenia grupy osi ruchem zwanym Joint. Dotyczy to zarówno ruchu pojedynczej osi jak i połączonego ruchu kilku bądź wszystkich osi. Zmiana położenia ramienia roboczego może też być wynikiem zmiany położenia lub orientacji punktu TCP (Tool Center Point – punkt centralny narzędzia) narzędzia roboczego, w którymś z dostępnych, kartezjańskich układów współrzędnych zgodnie z zastosowaną interpolacją. Kontroler R-H oferuje wszelkie niezbędne układy współrzędnych i umożliwia łatwą ich konfigurację i definiowanie z poziomu operatora. Zmiany, definiowanie i wybór może odbywać się z poziomu uruchomionego programu co daje programiście niezwykle elastyczne narzędzie w procesie integracji robota FANUC S-420F[3].

Kontroler R-H umożliwia definiowanie, poruszanie ramienia i stosowanie w tworzonych programach następujących układów współrzędnych i ruchów:

- **Joint** Rys. 4. 1– służy do poruszania pojedynczymi osiami lub ich grupami.
- **Base Frame** (World Frame) – bazowego układu współrzędnych robota.
- **User Frame** – układu współrzędnych skonfigurowanego zgodnie z geometrią zewnętrznego np. obrabianego przedmiotu.
- **Tool Frame** – układu współrzędnych zgodnego z geometrią narzędzia roboczego.



Rys. 4. 1. Ruchy JOINT poszczególnych osi Robota Fanuc S-420F[1]



Rys. 4. 2. Układy współrzędnych Robota Fanuc S-420F [4]

4. 2. Sposoby interpolacji

Język KAREL daje nam do dyspozycji cztery typy interpolacji, czyli sposobu przemieszczania TCP (Tool Centre Point – punkt bazowy narzędzia)[5]:

- ruch pojedynczej osi,
- ruch **JOINT** złożony ruch wielu osi równocześnie,
- interpolacja typu **LINEAR**, liniowa interpolacja przemieszczająca TCP po linii prostej,
- oraz interpolacją kołową **CIRCULAR** dla wykonywania ruchów po łuku.

By wykonać ruch interpolacją **JOINT**, **CIRCULAR** lub **LINEAR** program musi poinformować system obliczania trajektorii ruchu o tym fakcie przez ustawienie zmiennej **\$MOTYPE** zgodnie z zakładaną interpolacją

. Po tym można użyć odpowiedniej instrukcji ruchu by przemieścić TCP do punktu docelowego.

Tabela 4. 1 Typy interpolacji

Predefiniowana	liczbowa
JOINT	6
LINEAR	7
CIRCULAR	8

5. LISTING PROGRAMU ZINTEGROWANEGO ZROBOTYZOWANEGO SYSTEM W OPARCIU O ROBOTA FANUC

Napisany listing programu lab 0 dla Zintegrowanego Zrobotyzowanego Systemu ma na celu zapoznanie z budową przykładowego programu, sposobami jego testowania oraz metodami wprowadzania zmian w położenia punktów programu[3].

PROGRAM lab0

```

-- ***** --
-- **           Laboratorium 0           ** --
-- ***** --

VAR                -- deklaracja zmiennych
  base_tool      ,  -- narzedzie bazowe - czolo montazowe
  base_frame     ,  -- bazowy ukklad wspolrzecznych
  table_frame    ,  -- ukklad wspolrzecznych tablicy
  lab_tool       ,  -- ukklad wspolrzecznych narzedzia
  home           ,  -- punkt wyjsciowy - spoczynkowy
  start          ,  -- pozycja przejsciowa - bezpieczna
  clean_tool     ,  -- pozycja czyszczenia narzedzia
  pick          : POSITION  -- pozycja pobrania detalu

  TEST_PATH     : PATH

ROUTINE ResetFrames
BEGIN
  $UFRAME      = base_frame
  $UTOOL       = base_tool
END ResetFrames

-- ===== M A I N   P R O G R A M   ===== --

BEGIN

  WRITE (CR,'POCZATEK PROGRAMU LAB0')
  base_frame = POS(0,0,0,0,0,0,'N')
  base_tool  = POS(0,0,0,0,0,0,'N')
  table_frame = POS(1860,-1140,290,-180,-82.2,-49,'N')
  lab_tool    = POS(145,-55,295,-8,38,-27,'N')

  $SPEED      = 150
  $TERMTYPE   = NODECEL
  $MOTYPE     = JOINT

  ResetFrames

  WRITE (CR,CR,CR,'UWAGA !!!')
  WRITE (CR,'Nacisniecie F1 ruch do poz. HOME')
  WRITE TPMENU (CR,'TO HOME           ')
  WAIT FOR PIN[1]
  WRITE (CR,'Robot sie porusza')

  WITH $TERMTYPE=FINE,
  MOVE TO home
  WRITE (CR,'Robot w pozycji HOME')

  $UTOOL      = lab_tool
  $UFRAME     = table_frame

  WRITE (CR,CR,'UWAGA !!!')
  WRITE (CR,'Nacisniecie F1 powoduje wykonanie programu')
  WRITE TPMENU (CR,'START           ')

```

```
WAIT FOR PIN[1]          -- czeka na klawisz F1 na TP w User Menu

WRITE (CR,'Robot sie porusza')
MOVE TO start            -- przesun do punktu: start
WRITE (CR,'Robot w pozycji START')

WRITE (CR,'Robot sie porusza')
WITH $SPEED=800,        -- chwilowa zmiana predkosci na 500
MOVE NEAR clean_tool BY 50 -- przesun do punktu: clean_tool
WRITE (CR,'Robot 50 mm od pozycji clean_tool')

WRITE (CR,'Robot sie porusza')
WITH $TERMTYPE=FINE,
    $MOTYPE=LINEAR,
MOVE TO clean_tool      -- przesun do punktu: clean_tool
WRITE (CR,'Robot w pozycji clean_tool')

WRITE (CR,CR,'UWAGA !!!')
WRITE (CR,'Nacisniecie F1 powoduje kontynuowanie programu')
WRITE TPMENU (CR,'CONT          ')
WAIT FOR PIN[1]          -- czeka na klawisz F1 na TP w User Menu

WRITE (CR,'Robot sie porusza')
WITH $MOTYPE=LINEAR,
MOVE AWAY 100
WRITE (CR,'Robot 100 mm od pozycji clean_tool')

WRITE (CR,'Robot sie porusza')
WITH $SPEED=800,        -- chwilowa zmiana predkosci na 500
MOVE NEAR pick BY 50    -- dojedz do punktu: pick na odleglosc 50
WRITE (CR,'Robot 50 mm od pozycji pick')

WRITE (CR,'Robot sie porusza')
$MOTYPE = LINEAR        -- wymuszenie interpolacji liniowej
WITH $TERMTYPE=FINE,
MOVE TO pick            -- dojedz do pozycji ruchem liniowym
WRITE (CR,'Robot w pozycji pick')

WRITE (CR,CR,'UWAGA !!!')
WRITE (CR,'Nacisniecie F1 powoduje kontynuowanie programu')
WRITE TPMENU (CR,'CONT          ')
WAIT FOR PIN[1]          -- czeka na klawisz F1 na TP w User Menu

WRITE (CR,'Robot sie porusza')
MOVE AWAY 20            -- odsun od biezacego punktu o 20
WRITE (CR,'Robot 20 mm od pozycji pick')

ResetFrames              -- wywołanie procedury ResetFrames

WRITE (CR,'Robot sie porusza')
$MOTYPE = JOINT
WITH $SPEED=800,        -- chwilowa zmiana predkosci na 500
MOVE NEAR HOME BY 100
WRITE (CR,'Robot 100 mm od pozycji HOME')

WRITE (CR,'Robot sie porusza')
WITH $TERMTYPE=FINE,
```

```
$MOTYPE=LINEAR,  
MOVE TO home          -- powrot do pozycji spoczynkowej  
WRITE (CR,'Robot w pozycji HOME')  
WRITE (CR,'KONIEC PROGRAMU LAB0',CR)  
  
END lab0
```

6. WNIOSKI

Główny cel stworzenia Zintegrowanego Zrobotyzowanego Stanowiska w oparciu o robota przemysłowego FANUC S-420F to poznanie specyfiki zrobotyzowanych systemów przemysłowych. Jednocześnie stwarza ono możliwość przećwiczenia w warunkach laboratoryjnych praktycznych aspektów programowania ruchów robotach przygotowując w przyszłości do pracy w roli integratorów zrobotyzowanych systemów.

7. BIBLIOGRAFIA

- [1] Enhanced KAREL Operations Manual; ©GMFanuc Robotics Corporation, 1991
- [2] Kuśmińska-Fijałkowska, Zbigniew Łukasik „Zrobotyzowany laboratoryjny system w oparciu o robota FANUC S- 420F”. konferencja TRANSCOMP Zakopane 2010
- [3] Królikowski Jarosław „Praca inżynierska” promotor Zbigniew Łukasik Politechnika Radomska, 2008
- [4] R-J3 Controller Sealing Tool Operator's Manula ©FANUC LTD, 1998
- [5] R-J3 Controller Software Reference Manual ©FANUC LTD, 1998
- [6] <http://www.fanucrobotics.com>
- [7] <http://www.robotyka.com>