

Obsługa i programowanie robota FANUC (R-30iA)

Zadania do wykonania

Jacek Jagodziński Katarzyna Zadarnowska

1 Wymagania wstępne

- Zapoznanie się z zasadami bezpieczeństwa pracy na stanowisku laboratoryjnym.
- Zapoznanie się z treścią instrukcji: *Obsługa i programowanie robota FANUC (R-30iA)*, Katarzyna Zadarnowska, Jacek Jagodziński.
- Nazwy programów pisane podczas zajęć powinny zaczynać się od liter STUD (np. STUDPROG1).

2 Przebieg ćwiczenia

1. Schemat ideowy robota przedstawiono na rys. 1. Na końcu każdego ogniwa robota dołączony jest układ lokalny stowarzyszony z tym ogniwem. Współrzędnymi wewnętrznymi są:

- q_1 - kąt obrotu kolumny [°],
- q_2 - kąt odchylenia ramienia względem pionu [°],
- q_3 - kąt odchylenia przedramienia względem ramienia [°],
- q_4 - kąt obrotu przedramienia względem własnej osi [°],
- q_5 - kąt obrotu chwytaka względem przedramienia [°],
- q_6 - kąt obrotu chwytaka względem własnej osi [°].

Jako współrzędne zewnętrzne wybrano następujące zmienne:

$x; y; z$ - współrzędne kartezjańskie końca efektora wyrażone w układzie bazowym $X_0Y_0Z_0$ [mm]. Model kinematyki dla robota FANUC uzyskuje się przy użyciu notacji Denavita-Hartenberga. Transformacja Denavita-Hartenberga pomiędzy lokalnymi układami współrzędnych stowarzyszonymi z poszczególnymi ogniwami robota jest następująca

$$\begin{aligned}0 - 1 : A_1 &= Rot(z, q_1) \cdot Trans(x, d_1) \cdot Rot(x, \frac{\pi}{2}), \\1 - 2 : A_2 &= Rot(z, \frac{\pi}{2} - q_2) \cdot Trans(x, d_2), \\2 - 3 : A_3 &= Rot(z, q_2 + q_3) \cdot Trans(x, d_3) \cdot Rot(x, \frac{\pi}{2}), \\3 - 4 : A_4 &= Rot(z, -q_4) \cdot Trans(z, d_4) \cdot Rot(x, -\frac{\pi}{2}), \\4 - 5 : A_5 &= Rot(z, q_5) \cdot Rot(x, \frac{\pi}{2}), \\5 - 6 : A_6 &= Rot(z, -q_6) \cdot Trans(z, d_5) \cdot Rot(x, -\frac{\pi}{2}),\end{aligned}$$

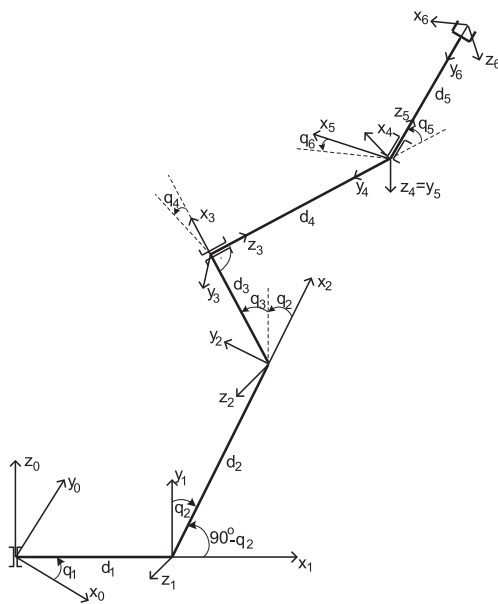
przy czym symbolami d_i oznaczono następujące parametry geometryczne robota:

- d_1 - wysunięcie ramienia względem początku układu współrzędnych $X_0Y_0Z_0$,
- d_2 - długość ramienia [mm],
- d_3 - odległość między układem $X_2Y_2Z_2$, a $X_3Y_3Z_3$,
- d_4 - długość przedramienia [mm],
- d_5 - długość chwytaka [mm].

Używając transformacji Denavita-Hartenberga obliczyć część translacyjną prostego modelu kinematyki

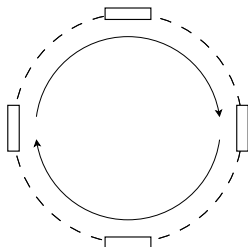
$$(q_1, q_2, q_3, q_4, q_5, q_6) \rightarrow (x, y, z).$$

2. Aby zidentyfikować parametry geometryczne d_i należy rozpocząć od ustawienia robota w pewnej konfiguracji poprzez zadanie określonych wartości współrzędnych wewnętrznych q_i . Następnie dla tak zadanej konfiguracji należy odczytać odpowiadające jej współrzędne zewnętrzne. W oparciu o prosty model kinematyki wyliczyć z układu równań stałe d_i .



Rysunek 1: Struktura kinematyczna robota

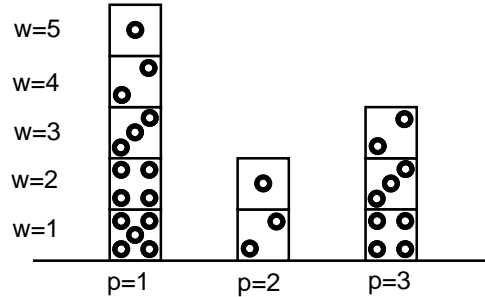
3. Napisz program przemieszczający manipulator między dwoma punktami w przestrzeni konfiguracyjnej i zadaniowej. Zaobserwuj różnice. Wskazówka: najlepiej zrobić to gdy chwytak jest opuszczony pionowo w dół, dlaczego?
4. Stwórz program kreślący dwukrotnie okrąg końcówką roboczą manipulatora, z równoczesnym obrotem wokół własnej osi (rys. 2).



Rysunek 2: Rzut ruchu po okręgu na płaszczyznę XY

5. Napisz prosty program do przestawiania 5 klocków z jednej ustalonej pozycji do innej. Wykorzystaj makra Open i Close do otwierania i zamykania chwytaka.
6. Napisz program poruszający efekтором pomiędzy 6 punktami (cztery niech zostaną zdefiniowane bardzo blisko siebie, a dwa względnie daleko). Ustaw dokładność na Fine, a następnie na CNT100. Zaobserwuj różnice.
7. Otwórz i zamykaj chwytak z programu zmieniając zmienne robota. Napisz program do pulsacyjnego zamykania i otwierania chwytaka co 1 sek. Wskazówka: sprawdź jak zbudowane są makra Open i Close.
8. Napisz program kreślący kwadrat końcówką chwytaka, następnie zmodyfikuj go wykorzystując Offset.
9. Zmodyfikuj poprzedni program na podprogram STUDKWADRAT kreślący kwadrat końcówką manipulatora w płaszczyźnie XY, wywoływany z dwoma argumentami: wielkością kwadratu oraz punktem określającym jeden z wierzchołków kwadratu. Następnie wykorzystaj STUDKWADRAT do wykreślenia szachownicy końcówką manipulatora.

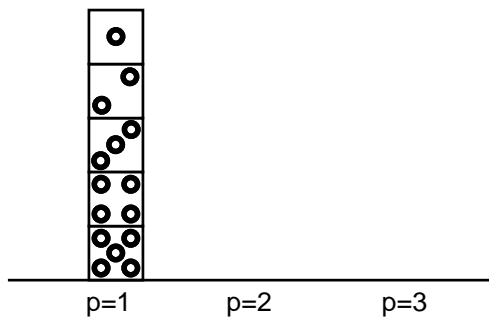
10. Zmodyfikuj program z poprzedniego zadania, tak by do komunikacji pomiędzy programami były wykorzystane rejestry.
11. Niech będą zdefiniowane trzy położenia w płaszczyźnie XY oznaczone przez $p \in \{1, 2, 3\}$ (wzdłuż osi np. X). W każdej pozycji można ustawić wieżę z klocków, wysokość na której stoi klocek opisana jest za pomocą zmiennej w , która przyjmuje wartość 1 gdy klocek stoi na podłożu, 2 gdy stoi na jednym klocku, 3 gdy na dwóch itd. (por. rys. 3). Napisz podprogram `STUDPRZESTAW`



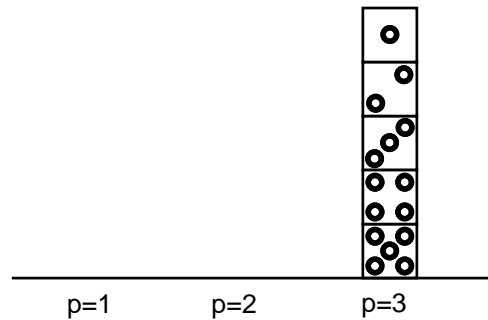
Rysunek 3: Pozycjonowanie klocków

`KLOCEK(w1, p1, w2, p2)` służący do przekładania kostki pomiędzy dwoma pozycjami, gdzie w_1, p_1 oznacza pozycję początkową klocka, natomiast w_2, p_2 jego docelową pozycję.

12. Wieża Hanoi: Korzystając z podprogramu `STUDPRZESTAWKLOCEK` przestaw klocki z pozycji początkowej rys. 4, do końcowej rys. 5 przestrzegając następujących reguł: przestawiaj tylko po jednym klocku, nie wolno kłaść klocka o większym numerze na klocku o mniejszym.



Rysunek 4: Wieża Hanoi pozycja startowa



Rysunek 5: Wieża Hanoi pozycja końcowa