ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1990
Seria: AUTOMATYKA z.101 Nr kol.1083

Artur Gan
Politechnika Śląska

OPTYMALIZACJA POŁOŻENIA ROBOTA W ŚRODOWISKU ROBOCZYM

Streszczenie. W pracy przedstawiono algorytm optymalizacji ustawienia robota przemysłowego w środowisku roboczym. W celu rozwiązania tego problemu zaproponowano kilka kryteriów optymalizacji.

1. Wprowadzenie

Jednym z ważniejszych elementów projektowania zrobotyzowanych stanowisk roboczych jest optymalizacja procesu manipulacyjnego. Optymalizacja ta obejmuje:

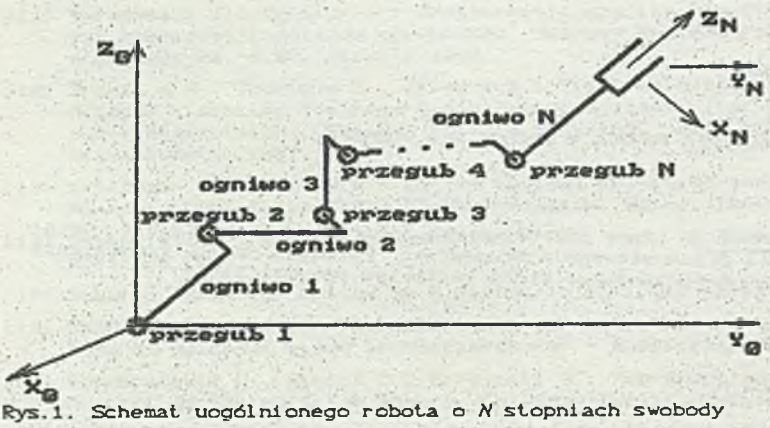
- optymalny podział zadań systemu manipulacyjnego na skończoną liczbę elementarnych powtarzających się cyklicznie kroków;
- optymalny dobór punktów podporowych zapewniający realizację trajektorii w granicach dopuszczalnej odchyłki;
- optymalne ustawienie robota przemysłowego w środowisku roboczym.

Przedmiotem tej pracy będzie optymalizacja ustawienia robota w środowisku roboczym. Właściwe ustawienie jednostki kinematycznej robota w stosunku do współpracujących z nią maszyn i urządzeń technologicznych znajdujących się w jej otoczeniu daje możliwość oszczędności energetycznych lub zapewnia minimalny czas trwania cyklu roboczego.

Zaproponowano tutaj iteracyjny algorytm optymalizacji tego problemu dla różnych kryteriów optymalizacji. Taki sposób rozwiązania problemu umożliwia stosowanie tego algorytmu niezależnie od struktury kinematycznej robota przemysłowego, który ma pracować na stanowisku roboczym.

2. Założenia i zapis matematyczny

Dla uogólnienia zadania przyjęto, że posiadamy robota o N stopniach swobody i znanej (dowolnie przyjętej) strukturze kinematycznej w postaci otwartego łańcucha kinematycznego. Struktura robota jest określana przez zwymiarowanie poszczególnych ogniw jego struktury kinematycznej i określenie rodzaju przegubu pomiędzy dwoma kolejnymi ogniwami (obrotowy lub przesuwny). Numerację ogniw i przegubów rozpoczęto od bazy robota (rys.1).



Rys.1. Schemat uogólnionego robota o N stopniach swobody
Fig.1. A schematic representation of generalized manipulator

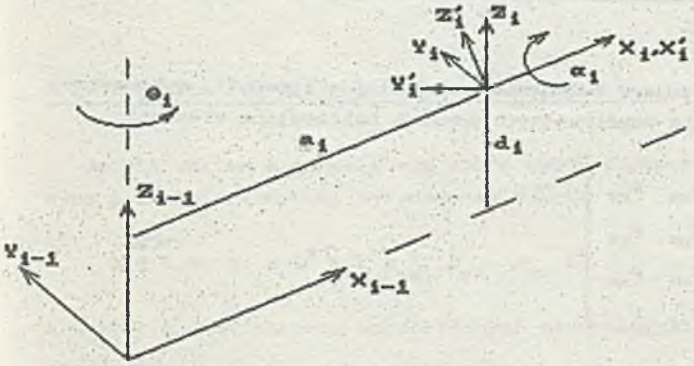
W artykule do opisu położenia i orientacji poszczególnych ogniw robota przyjęto tzw. macierze transformacji homogenicznych, które po raz pierwszy zostały wprowadzone przez Denavita i Hartenberga [1], a następnie spopularyzowane przez Piepera [2] i Paula [3].

Komercjalnie dostępne roboty zwykle mają prostą strukturę. Przeguby są obrotowe lub przesuwne, natomiast ogniwa łańcucha kinematycznego są proste i sztywne, a skręty pomiędzy nimi odbywają się w płaszczyźnie prostopadłej lub równoległej. Istnieje więc potrzeba wprowadzenia tylko prostych relacji, aby opisać transformacje układów współrzędnych opisujących położenie ogniw kinematycznych.

Przyjęto tutaj następujące konwencje dla wyboru układów współrzędnych opisujących położenie ogniw robota:
- ustawiać osie X wszystkich układów współrzędnych w przegubach w tym samym kierunku co oś X bazowego układu współrzędnych;
- przyjmować przeguby obrotowe obracane wokół ich indywidualnych osi Z;
- przyjmować przeguby przesuwne poruszające się wzdłuż ich indywidualnych osi Z.

Przyjęty układ współrzędnych stosowny do wyżej opisanych reguł, jako kolejna relacja pomiędzy ogniwem i a ogniwem $i-1$ (rys.2), jest opisany przez następujący zbiór parametrów:
θ_i – kąt rotacji θ_i wokół osi Z_{i-1};
d_i – translacja na odległość d_i wzdłuż osi Z_{i-1};
a_i – translacja na odległość a_i wzdłuż osi X_{i-1};
α_i – rotacja wokół osi X_i zgodnie z ruchem wskazówek zegara o kąt obrotu α_i pomiędzy osiami Z_i i Z_{i-1}.

Dla tak określonych reguł opisu macierz transformacji A_i, opisująca przejście z układu współrzędnych opisującego ogniwo $i-1$ do układu współ-



Rys.2. Relacja pomiędzy ogniwem i-1 a ogniwem i łańcucha kinematycznego robota

Fig.2. Relationship between two adjacent links: i-1 and i

rzędnych opisującego ogniwo i, ma postać:

$$A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (1)$$

Jeżeli znamy wszystkie parametry opisujące wzajemne położenie dwóch kolejnych ogniw, to macierz transformacji A_i pomiędzy kolejnymi układami współrzędnych jest określona. Wtedy transformacja układu współrzędnych opisującego położenie i orientację końcówki wykonawczej robota do układu współrzędnych bazowych jest dana jako:

$$T_N = A_1 A_2 A_3 \ldots A_{N-1} A_N \qquad (2)$$

Jeżeli i-ty przegub robota jest obrotowy, to współrzędną naturalną opisującą ruch tego przegubu jest kąt θ_i, natomiast gdy przegub jest przesuwny, to współrzędną naturalną jest zmienna d_i. Dla każdego przegubu jest określony zakres ruchów od $q_{i\min}$ do $q_{i\max}$. Wielkość $q_{i\min}$ określa minimalną wartość współrzędnej naturalnej q_i danego przegubu ($q_i = \theta_i$ dla przegubu obrotowego i $q_i = d_i$ dla przegubu przesuwnego), natomiast $q_{i\max}$ jej wartość maksymalną.

2.1. Algorytm iteracyjnego obliczania współrzędnych położenia końcówki wykonawczej robota

Transformacja $\mathbb{T}_N$ opisująca położenie i orientację końcówki wykonawczej robota w bazowym układzie współrzędnych posiada następujące elementy:

$$\mathbb{T}_N = \begin{bmatrix} n_{xN} & o_{xN} & a_{xN} & p_{xN} \\ n_{yN} & o_{yN} & a_{yN} & p_{yN} \\ n_{zN} & o_{zN} & a_{zN} & p_{zN} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Współrzędne położenia końcówki wykonawczej robota p_{xN}, p_{yN} i p_{zN} mogą być obliczone iteracyjnie na podstawie wzoru (2) mnożąc kolejno macierze transformacji od końca, czyli od $\mathbb{A}_N$ do $\mathbb{A}_1$:

$$\mathbb{T}_N^{N-i} = \mathbb{A}_{N-i+1}\mathbb{T}_N^{N-i+1}, \qquad \mathbb{T}_N^N = \mathbb{I}, \qquad i = 1,\ldots,N \tag{4}$$

przy czym $\mathbb{T}_N = \mathbb{T}_N^0$.

Taki sposób mnożenia macierzy $\mathbb{A}_i$ pozwala na obliczenie współrzędnych położenia końcówki wykonawczej poprzez iteracyjne obliczanie tylko elementów p_{xi}, p_{yi} i p_{zi} macierzy transformacji $\mathbb{T}_N^{N-i}$, określającej położenie i orientację końcówki wykonawczej względem końcówki ogniwa i-1:

$$\mathbb{T}_N^{N-i} = \begin{bmatrix} n_{xi} & o_{xi} & a_{xi} & p_{xi} \\ n_{yi} & o_{yi} & a_{yi} & p_{yi} \\ n_{zi} & o_{zi} & a_{zi} & p_{zi} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

Wzory na obliczanie podanych elementów mają postać:

$$p_{xi} = p_{xi-1}\cos\theta_{N-i+1} - p_{yi-1}\sin\theta_{N-i+1}\cos\alpha_{N-i+1} + p_{zi-1}\sin\theta_{N-i+1}\sin\alpha_{N-i+1} + a_{N-i+1}\cos\theta_{N-i+1} \tag{6}$$

$$p_{yi} = p_{xi-1}\sin\theta_{N-i+1} - p_{yi-1}\cos\theta_{N-i+1}\cos\alpha_{N-i+1} - p_{zi-1}\cos\theta_{N-i+1}\sin\alpha_{N-i+1} + a_{N-i+1}\sin\theta_{N-i+1} \tag{7}$$

$$p_{zi} = p_{yi-1}\sin\alpha_{N-i+1} + p_{zi-1}\cos\alpha_{N-i+1} + d_{N-i+1} \tag{8}$$

Ta metoda, ze względu na swoją prostotę, jest bardzo efektywna.

Dla i-tego kroku współrzędna naturalna q_i dla danego przegubu (θ_i dla przegubu obrotowego i d_i dla przesuwnego) powinna mieścić się w zakresie:

$$q_{i\min} \le q_i \le q_{i\max} \tag{9}$$

2.2. Algorytm iteracyjnego obliczania Jakobianu

Jeżeli wektor X określa nam zbiór współrzędnych kartezjańskich opisujących położenie końcówki wykonawczej robota:

$$X = [\, x,\ y,\ z \,]^T = [\, \rho_{xN},\ \rho_{yN},\ \rho_{zN} \,]^T \tag{10}$$

a wektor q jest zbiorem współrzędnych naturalnych:

$$q = [\, q_1,\ q_2,\ \ldots,\ q_N \,]^T \tag{11}$$

to zależność współrzędnych wektora X od współrzędnych wektora q (opisana powyżej algorytmem iteracyjnego obliczania) może być przedstawiona w ogólnej postaci:

$$X = F(q) \tag{12}$$

lub wyrażając w postaci macierzowej:

$$\begin{bmatrix} \rho_{xN} \\ \rho_{yN} \\ \rho_{zN} \end{bmatrix} = \begin{bmatrix} F_x(q_1, q_2, \ldots, q_N) \\ F_y(q_1, q_2, \ldots, q_N) \\ F_z(q_1, q_2, \ldots, q_N) \end{bmatrix} \tag{13}$$

Różniczkując równanie (12) po czasie otrzymujemy:

$$X' = (\partial F(q)/\partial q)(dq/dt) = J(q)q' \tag{14}$$

lub w postaci macierzowej:

$$\begin{bmatrix} \rho'_{xN} \\ \rho'_{yN} \\ \rho'_{zN} \end{bmatrix} = \begin{bmatrix} \partial F_x/\partial q_1 & \partial F_x/\partial q_2 & \ldots & \partial F_x/\partial q_N \\ \partial F_y/\partial q_1 & \partial F_y/\partial q_2 & \ldots & \partial F_y/\partial q_N \\ \partial F_z/\partial q_1 & \partial F_z/\partial q_2 & \ldots & \partial F_z/\partial q_N \end{bmatrix} =$$

$$= \begin{bmatrix} J_{x1} & J_{x2} & \ldots & J_{xN} \\ J_{y1} & J_{y2} & \ldots & J_{yN} \\ J_{z1} & J_{z2} & \ldots & J_{zN} \end{bmatrix} \tag{15}$$

gdzie macierz J(q) jest macierzą Jakobianu.

Wartości elementów macierzy Jakobianu można obliczyć (podobnie jak współrzędne położenia końcówki wykonawczej) w sposób iteracyjny. Z wzorów iteracyjnych (6)-(8) widać, że tylko w kroku $i=N-j+1$ istnieje zależność od zmiennej q_{N-j+1}. Stąd (po zróżniczkowaniu elementów p_{xi}, p_{yi} i p_{zi} względem zmiennych q_n) wzory na obliczanie elementów macierzy Jakobianu J_{xj}, J_{yj} oraz J_{zj} będą miały postać:

$$J^i_{xk}=\begin{cases} J^{i-1}_{xj}\cos\theta_{N-i+1}-J^{i-1}_{yj}\sin\theta_{N-i+1}\cos\alpha_{N-i+1}+ & \\ \quad +J^{i-1}_{zj}\sin\theta_{N-i+1}\sin\alpha_{N-i+1}+a_{N-i+1}\cos\theta_{N-i+1} & \text{dla } i\neq N-k+1 \\ -J^{i-1}_{xj}\sin\theta_{N-i+1}-J^{i-1}_{yj}\cos\theta_{N-i+1}\cos\alpha_{N-i+1}+ & \\ \quad +J^{i-1}_{zj}\cos\theta_{N-i+1}\sin\alpha_{N-i+1}-a_{N-i+1}\sin\theta_{N-i+1} & \text{dla } i=N-k+1 \text{ i } q_i=\theta_i \\ 0 & \text{dla } i=N-k+1 \text{ i } q_i=d_i \end{cases} \tag{16}$$

$$J^i_{yk}=\begin{cases} J^{i-1}_{xj}\sin\theta_{N-i+1}+J^{i-1}_{yj}\cos\theta_{N-i+1}\cos\alpha_{N-i+1}+ & \\ \quad -J^{i-1}_{zj}\cos\theta_{N-i+1}\sin\alpha_{N-i+1}+a_{N-i+1}\sin\theta_{N-i+1} & \text{dla } i\neq N-k+1 \\ J^{i-1}_{xj}\cos\theta_{N-i+1}-J^{i-1}_{yj}\sin\theta_{N-i+1}\cos\alpha_{N-i+1}+ & \\ \quad +J^{i-1}_{zj}\sin\theta_{N-i+1}\sin\alpha_{N-i+1}+a_{N-i+1}\cos\theta_{N-i+1} & \text{dla } i=N-k+1 \text{ i } q_i=\theta_i \\ 0 & \text{dla } i=N-k+1 \text{ i } q_i=d_i \end{cases} \tag{17}$$

$$J^i_{zk}=\begin{cases} J^{i-1}_{yj}\sin\alpha_{N-i+1}+J^{i-1}_{zj}\cos\alpha_{N-i+1}+d_{N-i+1} & \text{dla } i\neq N-k+1 \\ 0 & \text{dla } i=N-k+1 \text{ i } q_i=\theta_i \\ 1 & \text{dla } i=N-k+1 \text{ i } q_i=d_i \end{cases} \tag{18}$$

2.3. Algorytm iteracyjnego rozwiązywania równań kinematyki

Jeżeli równania kinematyki dla danej struktury robota są określone za pomocą równania (12) $X = F(q)$, to rozwiązywanie równań kinematyki możemy interpretować jako znajdowanie takich wartości q_r, dla których $X = F(q_r) = X_r$. Jedną z najbardziej rozpowszechnionych numerycznych metod poszukiwania rozwiązań nieliniowych układów równań jest metoda Newtona - -Raphsona [4]. Metoda ta bazuje na problemie optymalizacji kryterium funkcyjnego:

$$d(q) = | X_r - X |^2 = | X_r - F(q) |^2 \tag{19}$$

W przypadku gdy minimum funkcji $d(q)$ osiąga zero, to współrzędne wektora q stanowią rozwiązanie równań kinematyki, natomiast gdy minimum tej funkcji osiąga wartość większą od zera, to rozwiązanie równań kinematyki

nie istnieje (zadany punkt X_r znajduje się poza przestrzenią roboczą robota).

Jeżeli q_k jest estymatą poszukiwanego rozwiązania q_r funkcji $X_r - F(q)$ w k-tej iteracji metody Newtona-Raphsona to nowa estymata q_{k+1} może być obliczona przez linearyzację funkcji $X_r - F(q)$ w najbliższym sąsiedztwie aktualnej estymaty q_k:

$$X_r - F(q_{k+1}) = X_r - F(q_k) - J(q_k)\Delta q_k \qquad (20)$$

ponieważ linearyzacja funkcji $F(q)$ ma postać:

$$F(q_{k+1}) = F(q_k) - J(q_k)\Delta q_k \qquad (21)$$

gdzie $J(q_k)$ jest macierzą Jakobianu opisaną w poprzednim podrozdziale.

Jeżeli założymy, że q_{k+1} jest rozwiązaniem, to $X_r - F(q_{k+1}) = 0$, co w konsekwencji daje:

$$X_r - F(q_k) - J(q_k)\Delta q_k = 0 \qquad (22)$$

stąd, po przekształceniu, otrzymujemy wzór iteracyjny na obliczanie estymaty rozwiązania q_r:

$$q_{k+1} = q_k + J^{-1}(q_k)(X_r - X_k) \qquad (23)$$

przy czym wartości elementów wektora $X_k = F(q_k)$ są obliczane iteracyjnie według wzorów (8)-(8), a elementy macierzy Jakobianu $J(q_k)$ na podstawie wzorów (16)-(18).

3. Algorytm optymalizacji położenia robota w środowisku roboczym

Dla rozwiązania zagadnienia optymalizacji położenia robota w środowisku roboczym załóżmy, że mamy określone zadanie manipulacyjne przez zadanie $M+1$ punktów podporowych X_l $(l=0,\ldots,M)$, przez które ma przejść końcówka wykonawcza robota. Jeżeli przez X_B oznaczymy współrzędne bazowe położenia robota w środowisku roboczym, to współrzędne punktów podporowych X_l^* $(l=0,\ldots,M)$ odniesione do układu współrzędnych opisujących położenie robota w bazowym układzie współrzędnych, są określone jako:

$$X_l^* = X_l - X_B \qquad (24)$$

Celem optymalizacji położenia robota w środowisku roboczym może być, między innymi, minimalizacja czasu trwania przemieszczeń, minimalizacja zużytej energii lub też minimalizacja aktywnej przestrzeni roboczej robota.

Dla rozwiązania tego problemu optymalizacji zaproponowano tutaj trzy

różne kryteria funkcyjne:

$$I_1 = \sum_{l=1}^{M} \sum_{i=1}^{N} w_i^2 (q_{i,l} - q_{i,l-1})^2 \tag{25}$$

$$I_2 = \prod_{i=1}^{N} (q_{imax}^a - q_{imin}^a) \tag{26}$$

$$I_3 = \sum_{i=1}^{N} v_i (q_{imax}^a - q_{imin}^a) \tag{27}$$

gdzie $q_{i,l}$ - i-ta współrzędna naturalna robota dla l-tego punktu podporowego; q_{imax}^a, q_{imin}^a - zakresy aktywnych ruchów dla i-tej współrzędnej naturalnej robota:

$$q_{imin}^a = \min_l q_{i,l} \tag{28}$$

$$q_{imax}^a = \max_l q_{i,l} \tag{29}$$

w_i, v_i - współczynniki dopasowujące wartości współrzędnych naturalnych.

Kryterium I_1 minimalizuje czas trwania przemieszczeń (ze względu na minimalizację uogólnionej drogi we współrzędnych naturalnych robota). Pozostałe dwa kryteria minimalizują przede wszystkim wielkość aktywnej przestrzeni roboczej robota, co nie zawsze zapewnia zminimalizowanie cyklu roboczego lub zużycia energii.

Jeżeli przyjmiemy ogólną postać kryterium optymalizacji w postaci funkcji

$$I = I(q_1, \ldots, q_N) \tag{30}$$

i weźmiemy pod uwagę (na podstawie wzoru (12) i (24)), że

$$q_l = F^{-1}(X_l^*) = F^{-1}(X_l - X_B) \tag{31}$$

to możemy stwierdzić, że kryterium optymalizacji jest bezpośrednio funkcją współrzędnych bazowych położenia robota w środowisku roboczym.

Przedstawiony poniżej algorytm optymalizacji jest oparty na metodzie całościowego przeszukiwania. W pierwszej fazie można przyjąć cały obszar do przeszukiwania z dokładnością np. co 100 mm, w drugiej fazie przyjmuje się obszar 10-krotnie mniejszy w każdej współrzędnej z aktualnie najlepszym rozwiązaniem w centrum tego obszaru i dokonuje się przeszukania z dokładnością 10 mm itd.

krok 1 Przyjęcie pierwszego punktu poszukiwania $X_B^{(m)}$; $I_p \to \infty$ oraz $m = 0$;
krok 2 Obliczenie współrzędnych X_l^* dla $l = 1, \ldots, M$ na podst. wzoru (14);
krok 3 Rozwiązanie równań kinematyki q_l ($l = 1, \ldots, M$) według wzoru (23);
krok 4 Obliczenie wartości kryterium funkcyjnego I na podstawie wzoru

(25), (26) lub (27);

krok 5 Jeżeli $I<I_p$, to przyjmujemy $X_B^*=X_B^{(m)}$ oraz $I_p=I$;

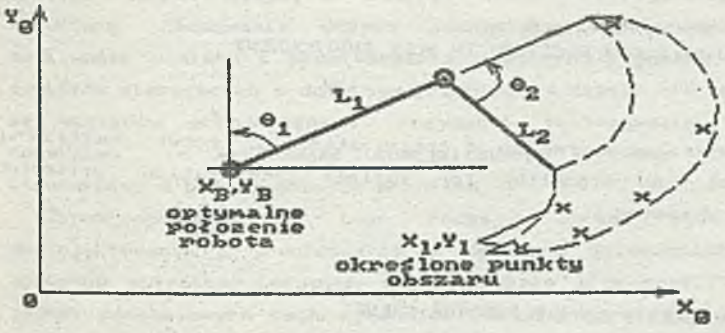
krok 6 Czy przeszukano wszystkie punkty $X_B^{(m)}$ w ramach zadanego obszaru dla zadanej dokładności? Jeżeli nie, to $m=m+1$, znalezienie kolejnego punktu poszukiwań $X_B^{(m)}$ i skok do kroku 2;

krok 7 Czy przeszukiwano obszar dla najmniejszej zadanej dokładności? Jeżeli nie, to zmiana dokładności, $m=m+1$, znalezienie kolejnego punktu poszukiwań $X_B^{(m)}$ i skok do kroku 2;

krok 8 Wyprowadzenie wartości współrzędnych optymalnego położenia X_B^*.

3. Przykład

Dla zilustrowania przedstawionego algorytmu weźmy pod uwagę przykład robota o dwóch obrotowych stopniach swobody (rys.3). Przyjmiemy, że długości ramion L_1 oraz L_2 są równe i wynoszą 245 mm. Dla zadanych pięciu kolejnych punktów podporowych (X_i,Y_i): (800,500), (900,500), (1000,500), (1100,500), (1000,750) rozwiązano problem optymalizacji przy wykorzystaniu kryterium funkcyjnego I_2 (wzór (26)), w wyniku czego uzyskano optymalne położenie robota w punkcie o współrzędnych (810,532).



Rys.3. Ilustracja obliczeń optymalnego położenia robota

Fig.3. Illustration to the calculation of the optimum base position of the robot

5. Podsumowanie

W pracy został przedstawiony algorytm optymalizacji położenia robota w środowisku roboczym. Algorytm ten odznacza się dużą prostotą i daje możliwość stosowania różnych kryteriów optymalizacji. Praktyczne zastosowanie tego algorytmu zostało zrealizowane na komputerze klasy IBM PC/XT/AT.

w wyniku czego stwierdzono dużą efektywność obliczeniową tego algorytmu. Dodatkową zaletą tego algorytmu jest możliwość jego stosowania w uniwersalnych systemach projektowania i symulacji pracy robota.

LITERATURA

[1] Denavit J., Hartenberg R.S.: A Kinematic Notation for Lower Pair Mechanisms Based on Matrices. Transactions of ASME, J. Applied Mechanics, June 1955.

[2] Pieper D.L.: The Kinematics of Manipulators Under Computer Control. Stanford Artifical Intelligence Laboratory, Stanford University, CA, USA, 1968.

[3] Paul R.P.: Robot Manipulators - Mathematics, Programming and Control, MIT Press, Cambridge, MA, USA, 1983.

[4] Whitney D.E.: Optimum Stepsize Control for Newton-Raphson Solution of Non-linear Vector Equations. Transactions of ASME, J. Dynamic Systems, Measurement and Control, December 1972.

Recenzent: Prof.dr h.inż.J.Cyklis

Wpłynęło do Redakcji do 1990-04-30.

THE OPTIMIZATION OF ROBOT POSITION IN WORK ENVIRONMENT

S u m m a r y

This paper describes a problem of optimization of robot position in work environment. An algorithm for various optimization criterions was developed and tested.

ОПТИМИЗАЦИЯ ПОЛОЖЕНИЯ РОБОТА В РАБОЧЕЙ СРЕДЕ

Р е з ю м е

В статье представлен алгоритм оптимизации положения робота в рабочей среде. Для решения этой проблемы предложены несколькие критерии оптимизации.