

POLITECHNIKA WROCŁAWSKA

WYDZIAŁ ELEKTRONIKI

KIERUNEK: Automatyka i Robotyka (AIR)
SPECJALNOŚĆ: Brak specjalności

PRACA DYPLOMOWA INŻYNIERSKA

Odsprzęganie wejściowo-wyjściowe dla
manipulatorów

Input-output decoupling algorithm for
manipulators

AUTOR:
Wojciech Kosicki

PROWADZĄCY PRACĘ:
Dr hab. inż. Alicja Mazur, Prof. PWr K7/W4

OCENA PRACY:

Spis treści

1	Wstęp	3
1.1	Cel pracy	3
1.2	Zawartość pracy	3
2	Modelowanie manipulatora IRb-6	5
2.1	Zasada Najmniejszego Działania	5
2.2	Dynamika manipulatora IRb-6	6
2.3	Kinematyka manipulatora IRb-6 według Denavita-Hartenberga	10
3	Algorytm odsprężania wejściowo-wyjściowego	13
3.1	Afiniczny układ sterowania	13
3.2	Konfiguracje osobliwe manipulatora IRb-6	14
3.3	Sterowanie odsprężające i linearyzujące	16
4	Sterowanie układem odsprężonym	21
4.1	Generator trajektorii zadanej	21
4.2	Regulacja PD z korekcją	22
5	Badania symulacyjne	25
5.1	Warunki symulacyjne	25
5.2	Ruch prostoliniowy	26
5.3	Ruch po trajektorii w kształcie kwadratu	32
5.4	Ruch śrubowy	34
6	Podsumowanie	37
	Bibliografia	37

Rozdział 1

Wstęp

Rewolucja przemysłowa, która rozpoczęła się w Anglii, drastycznie zmieniła świat w XIX wieku. Pomimo jednak rozwinięcia gospodarki oraz opracowania wielu technologii, do drugiej połowy XX wieku główną siłą roboczą byli tani pracownicy. Powojenny świat wymagał szybkiej odbudowy przemysłu i powstało wielkie zapotrzebowanie na usprawnienie linii produkcyjnych. Zaczęły powstawać pierwsze roboty przemysłowe. Roboty te zaczęły powoli zastępować ludzi przy liniach produkcyjnych. Były szybsze od człowieka i mogły pracować znacznie dłużej. Na przestrzeni kolejnych sześćdziesięciu lat rozwoju tej dziedziny, powstało wiele rodzajów i modeli manipulatorów do różnych prac w fabrykach. Jednak większość manipulatorów jest zbudowana z przegubów obrotowych. Wynika to z faktu, że przeguby przesuwne są trudne do wykonania oraz sterowania. Takie roboty miały ograniczone bardzo możliwości. Nie mogły poruszać swoim efektem po liniach prostych, dlatego też nie mogły wykonywać wielu prostych czynności m.in. dokonać spawania po linii prostej. Dopiero zastosowanie po wielu latach algorytmu odsprzęgania wejściowo-wyjściowego umożliwiło sterowanie efektem po odcinkach prostych w manipulatorach z przegubami obrotowymi.

1.1 Cel pracy

Celem pracy jest zapoznanie się z algorytmem odsprzęgania wejściowo-wyjściowego do sterowania manipulatorem IRb-6. Dzięki temu algorytmowi będzie możliwe sterowanie efektem manipulatora po liniach prostych, pomimo tego, że robot posiada pięć obrotowych stopni swobody.

1.2 Zawartość pracy

Rozdział pierwszy zawiera wstęp oraz opis celu pracy. W rozdziale drugim omówiono modelowanie manipulatora IRb-6 w środowisku Matlab/Simulink. Opisano jego dynamikę oraz kinematykę. Rozdział trzeci dotyczy algorytmu odsprzęgania wejściowo-wyjściowego oraz jego implementacji. W tym rozdziale zawarto także opis konfiguracji osobliwych manipulatora. W rozdziale czwartym zawarto model generatora trajektorii zadanej dla efektora oraz regulację PD z korekcją. Przeprowadzono też badania symulacyjne działania całego układu. W ostatnim, piątym, rozdziale dokonano podsumowania pracy.

Rozdział 2

Modelowanie manipulatora IRb-6

2.1 Zasada Najmniejszego Działania

Aby zdefiniować równania robota, należy najpierw wyprowadzić tzw. lagranżian układu

$$L(q, \dot{q}) = K(q, \dot{q}) - V(q) = \frac{1}{2} \dot{q}^T M(q) \dot{q} - V(q). \quad (2.1)$$

Zmienne $q \in R^3$ określają wektor położeń przegubów robota, zaś $\dot{q} \in R^3$ określają prędkości IRb-6. Zmienne te składają się na stan modelowanego układu. Lagranżian jest różnicą energii kinetycznej $K(q, \dot{q})$ oraz potencjalnej $V(q)$. Zgodnie z Zasadą Najmniejszego Działania, lagranżian musi spełniać równanie Eulera-Lagrange'a, czyli

$$\frac{d}{dt} \frac{\partial L(q, \dot{q})}{\partial \dot{q}} - \frac{\partial L(q, \dot{q})}{\partial q} = F_{zew}. \quad (2.2)$$

F_{zew} to siła wypadkowa oddziaływań zewnętrznych na układ. Uwzględnia ona siły oraz momenty sił wpływające na układ, m.in. siły oporu ruchu, siły tarcia czy siły sterujące manipulatorem. Składowa ta będzie oznaczana jako sygnał sterujący u . Podstawiając (2.1) do (2.2), otrzymuje się równanie dynamiki w postaci

$$\frac{\partial^2 L(q, \dot{q})}{\partial \dot{q}^2} \ddot{q} + \frac{\partial^2 L(q, \dot{q})}{\partial q \partial \dot{q}} \dot{q} - \frac{\partial L(q, \dot{q})}{\partial q} = u. \quad (2.3)$$

Uwzględniając postać energii kinetycznej robota daną wzorem (2.1), w postaci formy kwadratowej, uzyskano następujące równania dynamiki robota

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + D(q) = u. \quad (2.4)$$

Macierz $M(q)$ jest tzw. macierzą bezwładności, która jest symetryczna i dodatnio określona. Jest to macierz formy pojawiająca się w energii kinetycznej. Z kolei $C(q, \dot{q})$ to macierz sił Coriolisa oraz sił odśrodkowych. Trzeci element modelu, $D(q)$, to wektor opisujący siły oddziaływania grawitacyjnego na układ.

2.2 Dynamika manipulatora IRb-6

Założono, że model IRb-6 jest manipulatorem sztywnym. Założenie takie uwzględnia nieruchomość bazy manipulatora oraz fakt, że ramię IRb-6 składa się ze sztywnych ogniw połączonych przegubami sztywnymi.

Robot IRb-6 posiada pięć stopni swobody, jednak czwarty i piąty stopień są odpowiedzialne jedynie za orientację manipulatora, można więc je zignorować przy opisywaniu pozycji chwytaka. Przy takim założeniu można potraktować IRb-6 jako manipulator z trzema stopniami swobody.

Pierwszym krokiem w tworzeniu modelu robota IRb-6 będzie opracowanie jego dynamiki. W tym celu wykorzystane zostaną wcześniejsze równania (2.4). Modelowanie i badania będą przeprowadzone w środowisku Matlab/Simulink, jednak należy pamiętać, że pakiet Matlab rozwiązuje równania różniczkowe w sposób numeryczny. Aby móc go zastosować, należy najpierw przekształcić równania (2.4) do takiej postaci, które umożliwią rozwiązanie ich tą metodą. W tym celu należy przedstawić dynamikę w postaci $\ddot{q} = f(\dot{q}, q)$. Przekształcając kolejno

$$M(q)\ddot{q} = u - C(q, \dot{q})\dot{q} - D(q), \quad (2.5)$$

a następnie mnożąc równanie lewostronnie przez $M^{-1}(q)$, otrzymuje się wzór

$$\ddot{q} = M^{-1}(q)(u - C(q, \dot{q})\dot{q} - D(q)), \quad (2.6)$$

lub równoważnie

$$\ddot{q} = F(q)(u - C(q, \dot{q})\dot{q} - D(q)), \quad (2.7)$$

gdzie

$$F(q) = M^{-1}(q). \quad (2.8)$$

Macierz $F(q)$ jest odwrotnością macierzy bezwładności $M(q)$, której postać dla manipulatora IRb-6 jest następująca

$$M(q) = \begin{bmatrix} M_{11} & 0 & 0 \\ 0 & M_{22} & M_{23} \\ 0 & M_{23} & M_{33} \end{bmatrix}. \quad (2.9)$$

Poszczególne elementy tej macierzy są równe

$$\begin{aligned} M_{11} &= 5.66 + 6.56125s_2^2 - 0.531s_2c_3 + 10.9c_3^2, \\ M_{22} &= 11.18525, \\ M_{23} &= 0.13275s_{23}, \\ M_{33} &= 15.524. \end{aligned} \quad (2.10)$$

gdzie c_i , s_i to oznaczenia dla $\cos(q_i)$ oraz $\sin(q_i)$, natomiast $s_{23} = \sin(q_2 + q_3)$.

Macierz $F(q)$ będąca odwrotnością macierzy $M(q)$ jest wyliczana z definicji jako

$$F(q) = M(q)^{-1} = \frac{1}{\det M(q)} \cdot (M(q)^d)^T, \quad (2.11)$$

gdzie $\det M(q)$ to wyznacznik macierzy $M(q)$, a $M(q)^d$ to macierz dopełnień algebraicznych dla macierzy bezwładności. Pojawia się tu pewne ułatwienie, albowiem macierz dopełnienia jest macierzą symetryczną

$$(M(q)^d)^T = M(q)^d. \quad (2.12)$$

Dzięki temu, można uprościć zależność do postaci

$$F(q) = M(q)^{-1} = \frac{1}{\det M(q)} \cdot M(q)^d. \quad (2.13)$$

Wyznacznik macierzy bezwładności (2.9) jest równy

$$\det M(q) = (5.66 + 6.56125s_2^2 - 0.531s_2c_3 + 10.9c_3^2) \\ (173.639821 - (0.13275s_{23})^2). \quad (2.14)$$

Wykorzystując oznaczenia (2.10) wyznacznik ten można zapisać inaczej

$$\det M(q) = M_{11} \cdot (M_{22} \cdot M_{33} - (M_{23})^2). \quad (2.15)$$

Kolejnym krokiem było wyliczenie macierzy dopełnienia $M(q)^d$

$$M(q)^d = \begin{bmatrix} M_{22} \cdot M_{33} - (M_{23})^2 & 0 & 0 \\ 0 & M_{11}M_{33} & -M_{11}M_{23} \\ 0 & -M_{11}M_{23} & M_{11}M_{22} \end{bmatrix} \quad (2.16)$$

Dokonano wyliczenia macierzy odwrotnej jako

$$M(q)^{-1} = \frac{1}{M_{11} \cdot (M_{22} \cdot M_{33} - (M_{23})^2)} \cdot \begin{bmatrix} M_{22} \cdot M_{33} - (M_{23})^2 & 0 & 0 \\ 0 & M_{11}M_{33} & -M_{11}M_{23} \\ 0 & -M_{11}M_{23} & M_{11}M_{22} \end{bmatrix}.$$

Po odpowiednim wymnożeniu uzyskano

$$F(q) = M(q)^{-1} = \begin{bmatrix} F_{11} & 0 & 0 \\ 0 & F_{22} & F_{23} \\ 0 & F_{23} & F_{33} \end{bmatrix}. \quad (2.17)$$

gdzie

$$\begin{aligned} F_{11} &= \frac{1}{M_{11}}, \\ F_{22} &= \frac{M_{33}}{M_{22} \cdot M_{33} - (M_{23})^2}, \\ F_{23} &= \frac{-M_{23}}{M_{22} \cdot M_{33} - (M_{23})^2}, \\ F_{33} &= \frac{M_{22}}{M_{22} \cdot M_{33} - (M_{23})^2}. \end{aligned} \quad (2.18)$$

Teraz, znając odpowiednie funkcje, można zaimplementować macierz $M(q)$ oraz $F(q)$ w środowisku Simulink. Należy jednak pamiętać, że Matlab liczy równanie różniczkowe metodą numeryczną. Najpierw więc stworzono układ, który generuje zmienne stanu q oraz \dot{q} .

Bloki odpowiedzialne za przeliczenie macierzy $M(q)$ oraz $F(q)$ zawarto w bloku dynamiki w programie Simulink. Następnym blokiem, jaki należy opracować, jest blok odpowiedzialny za macierz sił odśrodkowych i sił Coriolisa, $C(q, \dot{q})$. Wykorzystując dane z pracy [4], przyjęto następującą postać tej macierzy

$$C(q, \dot{q}) = \begin{bmatrix} C_{12}^* \dot{q}_2 + C_{13}^* \dot{q}_3 & C_{12}^* \dot{q}_1 & C_{13}^* \dot{q}_1 \\ -C_{12}^* \dot{q}_1 & 0 & 2C_{23}^* \dot{q}_2 + C_{33}^* \dot{q}_3 \\ -C_{13}^* \dot{q}_1 & -C_{23}^* \dot{q}_2 & 0 \end{bmatrix}, \quad (2.19)$$

gdzie

$$\begin{aligned}
C_{12}^* &= \frac{1}{2}(6.56125s_{22} - 0.531c_2c_3), \\
C_{13}^* &= -\frac{1}{2}(-0.531s_2s_3 + 10.9s_{33}), \\
C_{23}^* &= 0.13275c_{23}, \\
C_{33}^* &= 0.13275c_{23}.
\end{aligned} \tag{2.20}$$

Macierz ta została uproszczona do postaci

$$C(q, \dot{q}) = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ -C_{12} & 0 & C_{23} \\ -C_{13} & C_{32} & 0 \end{bmatrix}, \tag{2.21}$$

gdzie

$$\begin{aligned}
C_{11} &= \frac{1}{2}\dot{q}_2(6.56125s_{22} - 0.531c_2c_3) \\
&\quad - \frac{1}{2}\dot{q}_3(-0.531s_2s_3 + 10.9s_{33}), \\
C_{12} &= \frac{1}{2}\dot{q}_1(6.56125\sin(2q_2) - 0.531c_2c_3), \\
C_{13} &= -\frac{1}{2}\dot{q}_1(-0.531s_2s_3 + 10.9\sin(2q_3)), \\
C_{23} &= 0.2655c_{23}(\dot{q}_2 + \frac{1}{2}\dot{q}_3), \\
C_{32} &= 0.13275\dot{q}_2c_{23}.
\end{aligned} \tag{2.22}$$

Wektor oddziaływań grawitacyjnych $D(q)$ ma prostą strukturę

$$D(q) = \begin{bmatrix} D_1 \\ D_2 \\ D_3 \end{bmatrix}, \tag{2.23}$$

gdzie

$$\begin{aligned}
D_1 &= 0, \\
D_2 &= -145\sin(q_2), \\
D_3 &= -5.7879\cos(q_3).
\end{aligned} \tag{2.24}$$

Punktem wyjściowym do implementacji dynamiki w Matlabie było równanie (2.7). W pełnej swojej postaci wygląda ono następująco

$$\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} = \begin{bmatrix} F_{11} & 0 & 0 \\ 0 & F_{22} & F_{23} \\ 0 & F_{23} & F_{33} \end{bmatrix} \left(\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} - \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ -C_{12} & 0 & C_{23} \\ -C_{13} & C_{32} & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} - \begin{bmatrix} D_1 \\ D_2 \\ D_3 \end{bmatrix} \right) \tag{2.25}$$

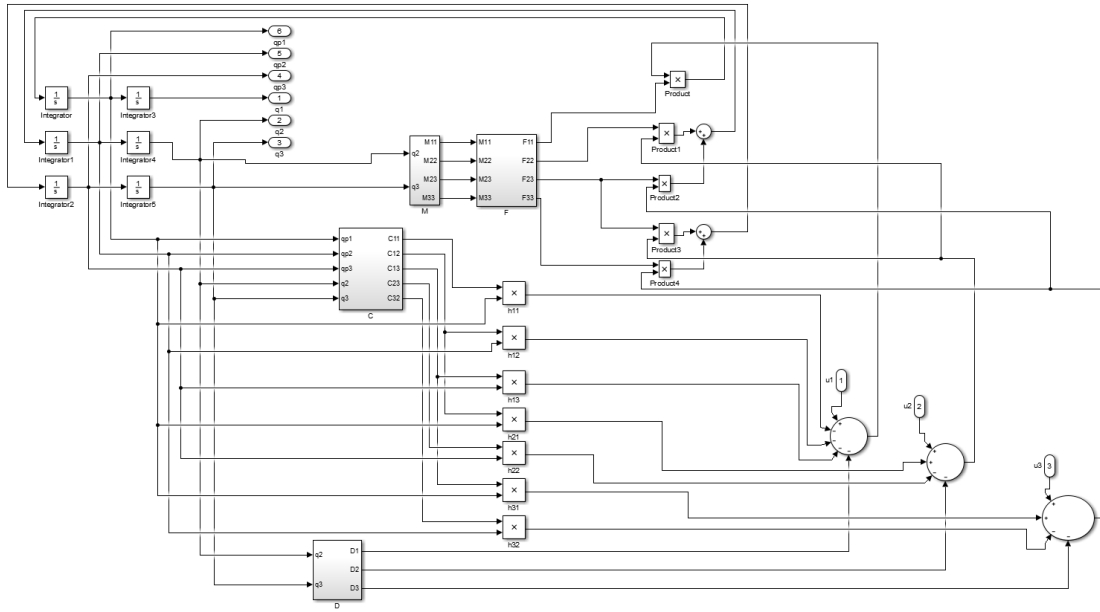
Po dokonaniu odpowiednich działań na macierzach w nawiasie uzyskano wyrażenie

$$\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} = \begin{bmatrix} F_{11} & 0 & 0 \\ 0 & F_{22} & F_{23} \\ 0 & F_{23} & F_{33} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix}, \quad (2.26)$$

gdzie

$$\begin{aligned} h_1 &= u_1 - C_{11}\dot{q}_1 - C_{12}\dot{q}_2 - C_{13}\dot{q}_3 - D_1, \\ h_2 &= u_2 + C_{12}\dot{q}_1 - C_{23}\dot{q}_3 - D_2, \\ h_3 &= u_3 + C_{13}\dot{q}_1 - C_{32}\dot{q}_2 - D_3. \end{aligned} \quad (2.27)$$

W ten sposób można zaimplementować nie tylko bloki $C(q, \dot{q})$, $D(q)$, ale także wyprowadzenie sygnału $\ddot{q} \in R^3$. Na końcu implementacji tego bloku trzeba połączyć układ z sygnałem sterującym $u \in R^3$. Ostateczny kształt układu dynamiki w programie Simulink został zamieszczony poniżej.



Rysunek 2.1 Dynamika manipulatora IRb-6 w programie Simulink.

2.3 Kinematyka manipulatora IRb-6 według Denavita-Hartenberga

W celu stworzenia kompletnego opisu matematycznego manipulatora IRb-6, oprócz jego dynamiki należy także określić jego kinematykę. Została ona opisana przy użyciu notacji Denavita-Hartenberga. Notacja ta pozwala opisać kinematykę robotów będących ramionami mechanicznymi według międzynarodowych standardów.

Parametry d_1, d_6, a_2, a_3 wynikają z własności fizycznych badanego manipulatora. Wynoszą one odpowiednio:

$$d_1 = 0.7 [m],$$

$$d_6 = 0.095 [m],$$

$$a_2 = 0.45 [m],$$

$$a_3 = 0.67 [m].$$

Manipulator IRb-6 ma pięć stopni swobody, jednak na jego położenie kąty q_4 oraz q_5 nie będą miały wpływu. Wynika to z faktu, że zmienne te są odpowiedzialne jedynie za obrót samego chwytaka. Można dzięki temu całą notację przedstawić jako odwzorowania

$$\mathbf{K}(\mathbf{q}) = A_0^1 A_1^2 A_2^3 A_3^4 A_4^5(q) : X_0 Y_0 Z_0 \rightarrow X_5 Y_5 Z_5 \quad (2.28)$$

gdzie

$$A_0^1 = \mathbf{Rot}(Z, q_1) \mathbf{Trans}(Z, d_1) \mathbf{Rot}(X, \frac{\pi}{2}),$$

$$A_1^2 = \mathbf{Rot}(Z, q_2) \mathbf{Trans}(Z, a_2),$$

$$A_2^3 = \mathbf{Rot}(Z, q_3 - q_2) \mathbf{Trans}(X, a_3),$$

$$A_3^4 = \mathbf{Rot}(Z, q_4 - q_3) \mathbf{Rot}(X, \frac{\pi}{2}),$$

$$A_4^5 = \mathbf{Rot}(Z, q_5) \mathbf{Trans}(Z, d_6) \mathbf{Rot}(X, -\frac{\pi}{2}).$$

Po przemnożeniu w odpowiedniej kolejności tych odwzorowań wyliczono macierz kinematyki jako

$$\mathbf{K}(\mathbf{q}) = \begin{bmatrix} c_1 c_4 c_5 + s_1 s_5 & -c_1 s_4 & s_1 c_5 - c_1 c_4 s_5 & c_1(a_2 c_2 + a_3 c_3 + d_6 s_4) \\ -s_4 c_5 & -c_4 & s_4 s_5 & -d_1 + d_6 c_4 - a_2 s_2 - a_3 s_3 \\ s_1 c_4 c_5 - c_1 s_5 & -s_1 s_4 & -c_1 c_5 - s_1 c_4 s_5 & s_1(a_2 c_2 + a_3 c_3 + d_6 s_4) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.29)$$

Trzy pierwsze elementy z ostatniej kolumny tej macierzy decydują o współrzędnych lokalizacji efektora w przestrzeni roboczej. Dlatego potrzebny do określenia kinematyki układ równań ma postać

$$\begin{cases} x &= c_1(a_2 c_2 + a_3 c_3 + d_6 s_4) \\ y &= -d_1 + d_6 c_4 - a_2 s_2 - a_3 s_3 \\ z &= s_1(a_2 c_2 + a_3 c_3 + d_6 s_4) \end{cases} \quad (2.30)$$

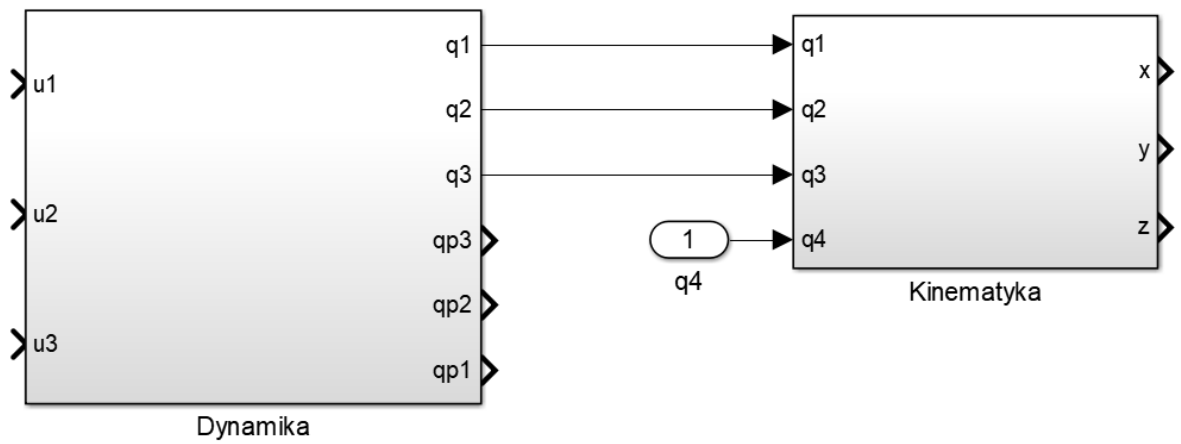
Jak widać, położenie chwytaka zależy od położenia przegubów $q_1 \div q_4$. Pozycje przegubów $q_1 \div q_3$ otrzymuje się z dynamiki robota, natomiast kąt q_4 , który jest potrzebny do wyliczenia położenia chwytaka, ma stałą wartość, ustaloną przez operatora. Z kolei q_5 wpływa tylko na orientację IRb-6, a nie na położenie chwytaka. Dlatego też przyjęto, że wartość kąta q_5 będzie ustaloną wartością stałą.

Na podstawie tych równań dokonano implementacji kinematyki robota IRb-6 w programie Simulink. Równania zarówno dynamiki i kinematyki takiego układu mają postać

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + D(q) = u, \quad (2.31)$$

$$y = K(q). \quad (2.32)$$

Powyższy opis dynamiki i kinematyki układu będzie podstawą do zamodelowania robota IRb-6 w programie Simulink. Sygnał wejściowy to $u \in R^3$, a współrzędne położenia efektora manipulatora to x, y, z . Wartości $qp1, qp2, qp3$ to odpowiednio $\dot{q}_1, \dot{q}_2, \dot{q}_3$.



Rysunek 2.2 Bloki dynamiki i kinematyki IRb-6 w programie Simulink.

Rozdział 3

Algorytm odsprzęgania wejściowo-wyjściowego

3.1 Afiniczny układ sterowania

Po zaimplementowaniu modelu manipulatora w programie symulacyjnym, można przystąpić do opracowania algorytmu odsprzęgania wejściowo-wyjściowego. Algorytm odsprzęgania wejściowo-wyjściowego składa się z dwóch części: odsprzężenia i linearyzacji przekształcenia wejściowo-wyjściowego oraz sterowania dla odsprzężonego układu liniowego. Taki algorytm pozwoli użytkownikowi sterować manipulatorem IRb-6 w sposób liniowy, tak jakby robot posiadał przeguby przesuwne.

Najpierw należy zacząć od uzyskania afinicznego układu sterowania. W poprzednim rozdziale zostały określone równania opisujące dynamikę i kinematykę manipulatora (2.31), (2.32). Drugie równanie opisuje kinematykę układu.

$$y_{CHi} = k_i(q), \quad i = 1, \dots, p, \quad (3.1)$$

które w postaci wektorowej wygląda następująco

$$y_{CH} = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \quad (3.2)$$

Jak widać, dla robota IRb-6 liczba wyjść jest równa $p = 3$. W celu uzyskania postaci afinicznej, należy uzyskać pochodną drugiego rzędu równania (3.1) po czasie. Dlatego zostaną wyliczone pochodne dla każdej składowej wektora $k(q)$

$$\frac{d}{dt}y_{CHi} = \frac{d}{dt}k_i(q) = \frac{\partial k_i}{\partial q} \frac{dq}{dt} = \frac{\partial k_i}{\partial q} \dot{q}, \quad i = 1, \dots, p. \quad (3.3)$$

Dla całego wektora pochodna będzie wynosiła zatem

$$\dot{y}_{CHi} = \frac{\partial k_i}{\partial q} \dot{q}, \quad i = 1, \dots, p, \quad (3.4)$$

a dla pochodnej drugiego rzędu zachodzi

$$\ddot{y}_{CHi} = \frac{d}{dt} \left(\frac{\partial k_i}{\partial q} \right) \dot{q} + \frac{\partial k_i}{\partial q} \ddot{q} = \dot{q}^T \frac{\partial^2 k_i}{\partial q^2} \dot{q} + \frac{\partial k_i}{\partial q} \ddot{q}, \quad i = 1, \dots, p. \quad (3.5)$$

Należy zwrócić uwagę na iloczyn $\dot{q}^T \frac{\partial^2 k_i}{\partial q^2} \dot{q}$. Nie jest on macierzą, tylko skalar, który oznaczony zostanie jako P_i . Równanie (3.5) można zatem uprościć do postaci

$$\ddot{y}_{CHi} = P_i + \frac{\partial k_i}{\partial q} \ddot{q}, \quad i = 1, \dots, p. \quad (3.6)$$

Równanie (3.6) dotyczy wszystkich składowych pozycji chwytaka, czyli zachodzi

$$\ddot{y}_{CH} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} + \begin{bmatrix} \frac{\partial k_1}{\partial q} \\ \frac{\partial k_2}{\partial q} \\ \frac{\partial k_3}{\partial q} \end{bmatrix} \ddot{q} = P + J\ddot{q}. \quad (3.7)$$

Symbol J oznacza macierz Jacobiego. kinematyki (2.30) wynosi

$$J = \begin{bmatrix} \frac{\partial k_1}{\partial q_1} & \frac{\partial k_1}{\partial q_2} & \frac{\partial k_1}{\partial q_3} \\ \frac{\partial k_2}{\partial q_1} & \frac{\partial k_2}{\partial q_2} & \frac{\partial k_2}{\partial q_3} \\ \frac{\partial k_3}{\partial q_1} & \frac{\partial k_3}{\partial q_2} & \frac{\partial k_3}{\partial q_3} \end{bmatrix}.$$

Kolejnym krokiem jest podstawienie równań dynamiki do układu (3.7). Wyliczoną wartość \ddot{q} podstawia się do równania (3.7), otrzymując

$$\ddot{y}_{CH} = P + JF(q)(u - C(q, \dot{q})\dot{q} - D(q)) = P + JF(q)u - JF(q)(C(q, \dot{q})\dot{q} + D(q)). \quad (3.8)$$

Z powyższego równania można wyróżnić część mnożoną przez sygnał wejściowy od reszty równania. W ten sposób równanie dochodzi do postaci afinicznej

$$\ddot{y}_{CH} = P - JF(q)(C(q, \dot{q})\dot{q} + D(q)) + JF(q)u = H + Gu, \quad (3.9)$$

gdzie:

- $H = P - JF(q)(C(q, \dot{q})\dot{q} + D(q))$,
- $G = JF(q) = JM^{-1}(q)$.

Uzyskany afiniczny układ sterowania (3.9) zostanie następnie zaimplementowany w programie Simulink. Zanim jednak zostanie to zrealizowane, należy najpierw upewnić się, czy takim układem da się sterować.

3.2 Konfiguracje osobliwe manipulatora IRb-6

W poprzednim rozdziale został przedstawiony układ sterowania dla manipulatora IRb-6 o określonej dynamice i kinematyce. Widać, że druga pochodna wyjścia y_{CH} zależy od sterowania

$$\ddot{y}_{CH} = H + Gu. \quad (3.10)$$

Aby powyższym układem dało się sterować, należy upewnić się, że macierz G jest macierzą odwracalną. Ponieważ

$$G = J(q)F(q), \quad (3.11)$$

to zachodzi

$$G^{-1} = (J(q)F(q))^{-1} = F(q)^{-1}J(q)^{-1} \quad (3.12)$$

Macierz $F(q)$ jest macierzą odwracalną, albowiem została ona wcześniej zdefiniowana jako odwrotność macierzy $M(q)$. Dlatego

$$G^{-1} = M(q)J(q)^{-1}, \quad (3.13)$$

Następnym krokiem jest określenie warunków, dla których macierz Jacobiego będzie odwracalna. Po pierwsze, macierz J musi być kwadratowa. Wynika to z warunku regularności sformułowanego w [3], czyli liczba wejść układu musi być równa liczbie wyjść układu. Warunek ten jest spełniony. Po drugie, badany manipulator nie może wykonywać ruchów przechodzących przez konfiguracje osobliwe. W tym celu policzone zostaną wszystkie osobliwości położenia manipulatora IRb-6. Cały proces sprowadza się do określenia, dla jakich kątów wyznacznik Jakobianu $\det J$ będzie równy zero. Poszczególne równania k_i to wcześniej uzyskane

$$\begin{cases} k_1 = x = c_1(a_2c_2 + a_3c_3 + d_6s_4) \\ k_2 = y = -d_1 + d_6c_4 - a_2s_2 - a_3s_3 \\ k_3 = z = s_1(a_2c_2 + a_3c_3 + d_6s_4) \end{cases}$$

Obliczając odpowiednie pochodne uzyskano macierz

$$J = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix},$$

gdzie

$$\begin{aligned} J_{11} &= \frac{\partial k_1}{\partial q_1} = -s_1(a_2c_2 + a_3c_3 + d_6s_4), \\ J_{12} &= \frac{\partial k_1}{\partial q_2} = -a_2c_1s_2, \\ J_{13} &= \frac{\partial k_1}{\partial q_3} = -a_3c_1s_3, \\ J_{21} &= \frac{\partial k_2}{\partial q_1} = 0, \\ J_{22} &= \frac{\partial k_2}{\partial q_2} = -a_2c_2, \\ J_{23} &= \frac{\partial k_2}{\partial q_3} = -a_3c_3, \\ J_{31} &= \frac{\partial k_3}{\partial q_1} = c_1(a_2c_2 + a_3c_3 + d_6s_4), \\ J_{32} &= \frac{\partial k_3}{\partial q_2} = -a_2s_1s_2, \\ J_{33} &= \frac{\partial k_3}{\partial q_3} = -a_3s_1s_3. \end{aligned} \quad (3.14)$$

Teraz określone zostaną wartości q_i , dla których konfiguracje są osobliwe. Pierwszym krokiem jest przyrównanie wyznacznika tej macierzy do zera. Założono, że

$$b = a_2c_2 + a_3c_3 + d_6s_4. \quad (3.15)$$

Podstawiając (3.15) do wyznacznika macierzy Jacobiego uzyskano zależność

$$\det J = -s_1^2 b a_2 a_3 c_2 s_3 + c_1^2 b a_2 a_3 s_2 c_3 - c_1^2 b a_2 a_3 c_2 s_3 + s_1^2 b a_2 a_3 s_2 c_3. \quad (3.16)$$

Wyciągając jedynkę trygonometryczną, równanie upraszcza się do postaci

$$\det J = -b a_2 a_3 (c_2 s_3 - s_2 c_3), \quad (3.17)$$

co równoważne jest z równaniem

$$\det J = -b a_2 a_3 \sin(q_3 - q_2). \quad (3.18)$$

Z zależności (3.18) wynikają dwa równania wyznaczające konfiguracje osobliwe:

•

$$b = a_2 c_2 + a_3 c_3 + d_6 s_4 = 0, \quad (3.19)$$

•

$$\sin(q_3 - q_2) = 0. \quad (3.20)$$

Z równania (3.19) otrzymuje się następujące wartości q_i konfiguracji osobliwych

$$(q_2 = \frac{\pi}{2} + k\pi) \cup (q_3 = \frac{\pi}{2} + k\pi) \cup (q_4 = \pi). \quad (3.21)$$

Stała k we wszystkich zależnościach oznacza liczby całkowite. Ostatecznie cały zbiór konfiguracji osobliwych to

$$((q_2 = \frac{\pi}{2} + k\pi) \cup (q_3 = \frac{\pi}{2} + k\pi) \cup (q_4 = k\pi)) \cap (q_3 - q_2 = k\pi). \quad (3.22)$$

Podczas sterowania układem należy upewnić się, że robot nie będzie przechodził przez konfiguracje osobliwe (3.22).

3.3 Sterowanie odsprężające i linearyzujące

W poprzednim rozdziale dokonano przekształcenia równania układu do postaci afinicznej

$$\ddot{y}_{CH} = P - JF(q)(C(q, \dot{q})\dot{q} + D(q)) + JF(q)\mathbf{u} = H + Gu, \quad (3.23)$$

gdzie:

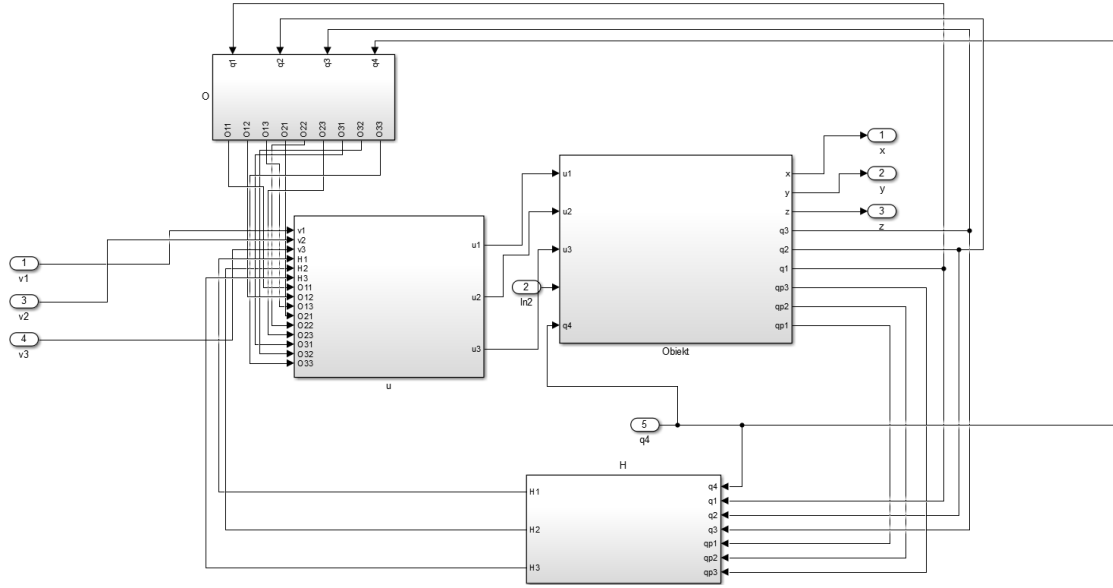
- $H = P - JF(q)(C(q, \dot{q})\dot{q} + D(q)),$
- $G = JF(q).$

Kolejnym krokiem będzie zaimplementowanie tego układu sterowania w programie Simulink. Polegać będzie to na utworzeniu odpowiednich sprzężeń zwrotnych dla tego obiektu i zamknięciu całości w nowy podsystem nazwany Układem.

Najpierw przekształcono równanie, definiując nowy sygnał wejściowy do obiektu, czyli v . Niech

$$u = G^{-1}(-H + v) \quad (3.24)$$

gdzie v jest oznaczeniem nowego wejścia do układu. Na rysunku 3.1 zamieszczono schemat podsystemu Układ.



Rysunek 3.1 Podsystem Układ.

Pierwszym utworzonym blokiem był blok H wynikający z równania

$$H = P - JF(q)(C(q, \dot{q})\dot{q} + D(q)) \quad (3.25)$$

Jego zawartość została podzielona na bloki M, N, F, J, G, E, P . Bloki M oraz F są kopią bloków z podsystemu Dynamiki i mają za zadanie utworzyć $F(q)$, czyli odwrotność macierzy bezwładności $M(q)$. Blok J odpowiedzialny jest za utworzenie macierzy Jacobiego $J(q)$.

$$J(q) = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix}$$

Pozostałe bloki E, G, P, N wynikają bezpośrednio z obliczeń macierzy równania H . W pierwszej kolejności przeanalizowano kwestię bloku P . Niech

$$P_i = \dot{q}^T \frac{\partial^2 k_i}{\partial q^2} \dot{q}, \quad i = 1, \dots, p. \quad (3.26)$$

Z mnożenia macierzy można dojść do wniosku, że jest to skalar zależny od wartości q_i oraz \dot{q}_i , według wzoru

$$P_i = \begin{bmatrix} \dot{q}_1 & \dot{q}_2 & \dot{q}_3 \end{bmatrix} \begin{bmatrix} P_{i11} & P_{i12} & P_{i13} \\ P_{i21} & P_{i22} & P_{i23} \\ P_{i31} & P_{i32} & P_{i33} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}, \quad i = 1, \dots, p. \quad (3.27)$$

gdzie:

$$\begin{aligned}
P_{111} &= -c_1(a_2c_2 + a_3c_3 + d_6s_4), \\
P_{112} &= -\sin(q_1)(-a_2\sin q_2 + a_3\cos q_3 + d_6\sin q_4), \\
P_{113} &= -s_1(a_2c_2 - a_3s_3 + d_6s_4), \\
P_{121} &= a_2s_1s_2, \\
P_{122} &= -a_2c_1c_2, \\
P_{132} &= 0, \\
P_{131} &= a_3s_1s_3, \\
P_{132} &= 0, \\
P_{133} &= -a_3c_1c_3, \\
P_{211} &= 0, \\
P_{212} &= 0, \\
P_{213} &= 0, \\
P_{221} &= 0, \\
P_{222} &= a_2s_2, \\
P_{223} &= 0, \\
P_{231} &= 0, \\
P_{232} &= 0, \\
P_{233} &= a_3s_3, \\
P_{311} &= -s_1(a_2c_2 + a_3c_3 + d_6s_4), \\
P_{312} &= -a_2c_1s_2, \\
P_{313} &= -a_3c_1s_3, \\
P_{321} &= -a_2c_1s_2, \\
P_{322} &= -a_2s_1c_2, \\
P_{323} &= 0, \\
P_{331} &= -a_3c_1s_3, \\
P_{332} &= 0, \\
P_{333} &= -a_3s_1c_3.
\end{aligned} \tag{3.28}$$

Po wymnożeniu uzyskano

$$\begin{aligned}
P_1 &= \dot{q}_1(-\dot{q}_1c_1(a_2c_2 + a_3c_3 + d_6s_4) + \dot{q}_2a_2s_1s_2 - \dot{q}_3a_3s_1s_3) \\
&\quad + \dot{q}_2(-\dot{q}_1s_1(-a_2s_2 + a_3c_3 + d_6s_4) - \dot{q}_2a_2c_1c_2) \\
&\quad + \dot{q}_3(-\dot{q}_1s_1(a_2c_2 - a_3s_3 + d_6s_4) - \dot{q}_3c_1c_3), \\
P_2 &= \dot{q}_2(\dot{q}_2a_2s_2) + \dot{q}_3\dot{q}_3a_3s_3, \\
P_3 &= \dot{q}_1(-\dot{q}_1s_1(a_2c_2 + a_3c_3 + d_6s_4) - \dot{q}_2a_2c_1s_2 - \dot{q}_3a_3c_1s_3) \\
&\quad + \dot{q}_2(-\dot{q}_1a_2c_1s_2 - \dot{q}_2a_2s_1c_2) + \dot{q}_3(-\dot{q}_1a_3c_1s_3 - \dot{q}_3a_3s_1c_3).
\end{aligned} \tag{3.29}$$

Blok G wynika z mnożenia $J(q)F(q) = G(q)$. W celu utworzenia macierzy $G(q)$ wykorzystano składowe macierzy Jacobiego oraz odwrotności macierzy bezwładno-

ści. Macierz ta ma postać

$$G(q) = \begin{bmatrix} G_{11} & G_{12} & G_{13} \\ G_{21} & G_{22} & G_{23} \\ G_{31} & G_{32} & G_{33} \end{bmatrix}$$

gdzie

$$\begin{aligned} G_{11} &= J_{11}F_{11}, \\ G_{12} &= J_{12}F_{22} + J_{13}F_{23}, \\ G_{13} &= J_{12}F_{22} + J_{13}F_{33}, \\ G_{21} &= 0, \\ G_{22} &= J_{22}F_{22} + J_{23}F_{23}, \\ G_{23} &= J_{22}F_{23} + J_{23}F_{33}, \\ G_{31} &= J_{32}F_{11}, \\ G_{32} &= J_{32}F_{22} + J_{33}F_{23}, \\ G_{33} &= J_{32}F_{23} + J_{33}F_{33}. \end{aligned} \tag{3.30}$$

W równaniu dokonywane jest działanie na macierzach $C(q, \dot{q})\dot{q} + D(q)$ dające wektor, który oznaczono jako $E(q, \dot{q})$. Za niego odpowiedzialny jest właśnie blok E . Jego składowe to

$$E(q, \dot{q}) = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ -C_{12} & 0 & C_{23} \\ -C_{13} & C_{32} & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} + \begin{bmatrix} D_1 \\ D_2 \\ D_3 \end{bmatrix} = \begin{bmatrix} C_{11}\dot{q}_1 + C_{12}\dot{q}_2 + C_{13}\dot{q}_3 + D_1 \\ -C_{12}\dot{q}_1 + C_{23}\dot{q}_3 + D_2 \\ -C_{13}\dot{q}_1 + C_{32}\dot{q}_2 + D_3 \end{bmatrix}.$$

Do zaimplementowania bloku E wykorzystano wcześniej utworzone bloki C oraz D . Ostatni Blok N zaś jest wynikiem mnożenia $G(q)E(q, \dot{q}) = N(q, \dot{q})$.

$$N(q, \dot{q}) = \begin{bmatrix} G_{11}E_1 + G_{12}E_2 + G_{13}E_3 \\ G_{21}E_1 + G_{22}E_2 + G_{23}E_3 \\ G_{31}E_1 + G_{32}E_2 + G_{33}E_3 \end{bmatrix}. \tag{3.31}$$

Ostatecznie, by uzyskać składowe wektora $H(q)$ należy na końcu odjąć wektor $N(q, \dot{q})$ od P .

Drugim blokiem w podsystemie Układ jest blok O , wyliczający macierz $O(q)$, która jest odwrotnością wcześniej wyliczonej macierzy $G(q)$. Sposób wyliczenia jest standardowy, wykorzystujący odwrotność wyznacznika tej macierzy. Wyznacznik macierzy $G(q)$ równa się

$$\det(G(q)) = G_{11}G_{22}G_{33} + G_{31}G_{12}G_{23} - G_{11}G_{32}G_{23} - G_{31}G_{22}G_{13} \tag{3.32}$$

Uzyskano macierz

$$O(q) = G^{-1}(q) = \begin{bmatrix} O_{11} & O_{12} & O_{13} \\ O_{21} & O_{22} & O_{23} \\ O_{31} & O_{32} & O_{33} \end{bmatrix},$$

z elementami równymi odpowiednio

$$\begin{aligned}
O_{11} &= \frac{1}{\det(G(q))} (G_{22}G_{33} - G_{23}G_{32}), \\
O_{12} &= \frac{1}{\det(G(q))} (G_{13}G_{32} - G_{11}G_{33}), \\
O_{13} &= \frac{1}{\det(G(q))} (G_{12}G_{23} - G_{13}G_{22}), \\
O_{21} &= \frac{1}{\det(G(q))} (G_{23}G_{31}), \\
O_{22} &= \frac{1}{\det(G(q))} (G_{11}G_{33} - G_{13}G_{31}), \\
O_{23} &= \frac{1}{\det(G(q))} (-G_{11}G_{23}), \\
O_{31} &= \frac{1}{\det(G(q))} (-G_{22}G_{31}), \\
O_{32} &= \frac{1}{\det(G(q))} (G_{12}G_{31} - G_{11}G_{32}), \\
O_{33} &= \frac{1}{\det(G(q))} (G_{11}G_{22}).
\end{aligned} \tag{3.33}$$

Ostatnim blokiem w Układzie jest blok U odpowiedzialny za połączenie wyników z bloków H oraz O ze sprzężenia zwrotnego. Jego działanie jest zależne od równania

$$u = \begin{bmatrix} O_{11} & O_{12} & O_{13} \\ O_{21} & O_{22} & O_{23} \\ O_{31} & O_{32} & O_{33} \end{bmatrix} \begin{bmatrix} v_1 - H_1 \\ v_2 - H_2 \\ v_3 - H_3 \end{bmatrix} = \begin{bmatrix} O_{11}(v_1 - H_1) + O_{12}(v_2 - H_2) + O_{13}(v_3 - H_3) \\ O_{21}(v_1 - H_1) + O_{22}(v_2 - H_2) + O_{23}(v_3 - H_3) \\ O_{31}(v_1 - H_1) + O_{32}(v_2 - H_2) + O_{33}(v_3 - H_3) \end{bmatrix}.$$

Rozdział 4

Sterowanie układem odsprzężonym

4.1 Generator trajektorii zadanej

W celu przeprowadzenia analizy jakości opracowanego algorytmu, trzeba określić jaką trajektorię ruchu manipulatora będzie realizowana. Z tego powodu utworzono dodatkowy blok nazwany Generatorem, który będzie generował żądane współrzędne chwytaka. Na początek wybrano ruch śrubowy o następujących wzorach:

$$\begin{cases} x_d(t) &= 0.05 \cos(t) \\ y_d(t) &= 0.05 \sin(t) \\ z_d(t) &= \frac{t}{100} \end{cases} \quad (4.1)$$

Tak jak w poprzednich rozdziałach, uwzględniono, że idealne różniczkowanie numeryczne nie istnieje. Zamiast tego zamodelowano najwyższe pochodne, a następnie scałkowano je z odpowiednimi warunkami początkowymi

$$\begin{aligned} x_d(t) &= 0.05 \cos(t), \\ \dot{x}_d(t) &= -0.05 \sin(t), \\ \ddot{x}_d(t) &= -0.05 \cos(t), \\ x_d(0) &= 0.05, \\ \dot{x}_d(0) &= 0, \\ y_d(t) &= 0.05 \sin(t), \\ \dot{y}_d(t) &= 0.05 \cos(t), \\ \ddot{y}_d(t) &= -0.05 \sin(t), \\ y_d(0) &= 0, \\ \dot{y}_d(0) &= 0.05, \end{aligned} \quad (4.2)$$

$$\begin{aligned} z_d(t) &= \frac{t}{100}, \\ \dot{z}_d(t) &= \frac{1}{100}, \\ \ddot{z}_d(t) &= 0, \\ z_d(0) &= 0, \\ \dot{z}_d(0) &= \frac{1}{100}. \end{aligned} \quad (4.3)$$

Z tak zaimplementowanym Generatorem będzie określana dla sygnału wejściowego odpowiednia trajektoria ruchu chwytaka. Ta z kolei będzie na bieżąco sprawdzana z rzeczywistymi współrzędnymi efektora IRb-6, co pozwoli na wyliczenie uchybu układu.

4.2 Regulacja PD z korekcją

Posiadając generator trajektorii, można przystąpić do tworzenia algorytmu gwarantującego zbieżność błędu śledzenia. Zaimplementowany zostanie regulator PD określony zależnością

$$v = \ddot{y}_{CHd} - K_d \dot{e}_y - K_p e_y, \quad (4.4)$$

gdzie

$$e_y = y_{CH} - y_{CHd}. \quad (4.5)$$

Regulator ten został opisany w [1]. Stałe K_d oraz K_p są nastawami regulatora. Wartość e_y oraz jej pochodna są sygnałami uchybu wynikającego z różnicy pomiędzy rzeczywistym położeniem efektora w przestrzeni roboczej, a żadaną trajektorią określoną przez generator.

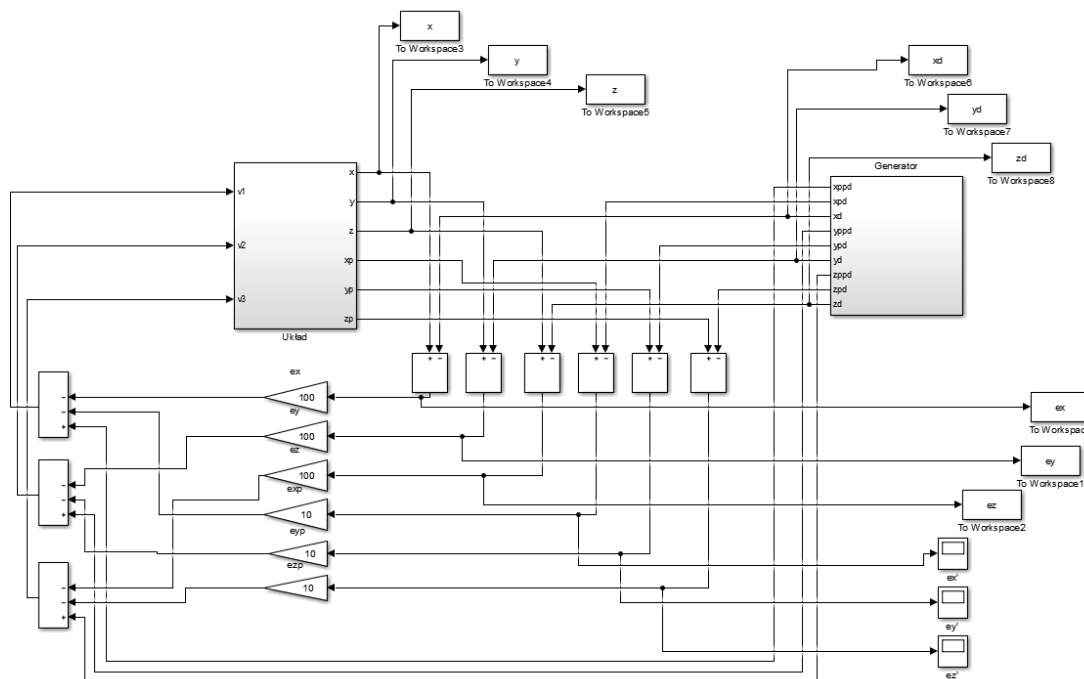
Na podstawie wzoru (4.4) opisano zależność sygnału sterującego od uchybu

$$\begin{aligned} v_1 &= \ddot{x}_d - K_d(\dot{x} - \dot{x}_d) - K_p(x - x_d), \\ v_2 &= \ddot{y}_d - K_d(\dot{y} - \dot{y}_d) - K_p(y - y_d), \\ v_3 &= \ddot{z}_d - K_d(\dot{z} - \dot{z}_d) - K_p(z - z_d). \end{aligned} \quad (4.6)$$

gdzie współrzędne x_d, y_d, z_d oraz ich pochodne pochodzą z generatora trajektorii, a współrzędne x, y, z można wyliczyć z kinematyki obiektu (2.30). Stałe współczynniki K_p oraz K_d dobrano uwzględniając zależność $K_p = 10K_d$. Wypróbowano kilka rzędów wielkości wartości stałych, jednak symulacje przeprowadzono dla wartości

- $K_p = 100$,
- $K_d = 10$.

W [1] wykorzystywano wartości stałych dziesięć razy mniejsze. Zwiększenie wartości stałych zmniejszy uchyb sterowania układu. Jednak będzie to też wymagało większej mocy obliczeniowej. Cały schemat regulacji został przedstawiony na rysunku 4.1.



Rysunek 4.1 Regulacja PD z korekcją.

Rozdział 5

Badania symulacyjne

5.1 Warunki symulacyjne

W celu przetestowania skuteczności algorytmu odsprzęgania wejściowo-wyjściowego, dokonano symulacji sterowania manipulatorem IRb-6 dla konkretnych trajektorii efektora. Symulacje zostały przeprowadzone w programie Matlab i Simulink w wersji R2016b. Komputer, który przeprowadzał symulację miał następujące parametry:

- Procesor Intel(R) Core(TM) i7-8550U CPU 1.80 GHz,
- Pamięć RAM 8.00 GB,
- System 64-bitowy Windows 10.

Dla wszystkich symulacji wykorzystywano metodę całkowania ode23 o dokładności 10^{-10} .

Dla każdej trajektorii dobrano jednakowe warunki początkowe ustawienia manipulatora. Wynoszą one kolejno

$$\begin{aligned}q_1(0) &= 1 \approx 57.3^\circ, \\q_2(0) &= 0.4 \approx 22.9^\circ, \\q_3(0) &= 0.1 \approx 5.7^\circ, \\q_4(0) &= 1.6 \approx 91.7^\circ.\end{aligned}\tag{5.1}$$

Konfiguracja początkowa została tak dobrana, by zminimalizować szansę osiągnięcia przez manipulator konfiguracji osobliwych, we wszystkich trajektoriach. Współrzędne efektora dla tej konfiguracji początkowej wynoszą

$$\begin{aligned}x &= 0.6354439932 \approx 0.64, \\y &= -0.9449005978 \approx -0.94, \\z &= 0.9104502922 \approx 0.91.\end{aligned}\tag{5.2}$$

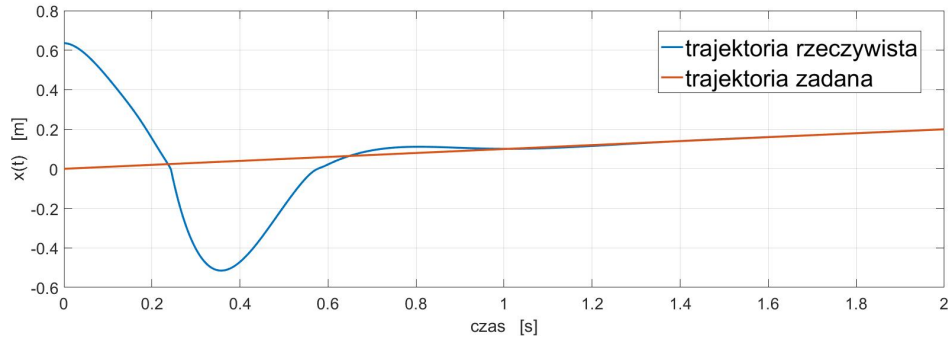
5.2 Ruch prostoliniowy

Pierwszym testem dla sterowania będzie przemieszczanie efektora w przestrzeni roboczej po liniach prostych. Przeprowadzono pomiar

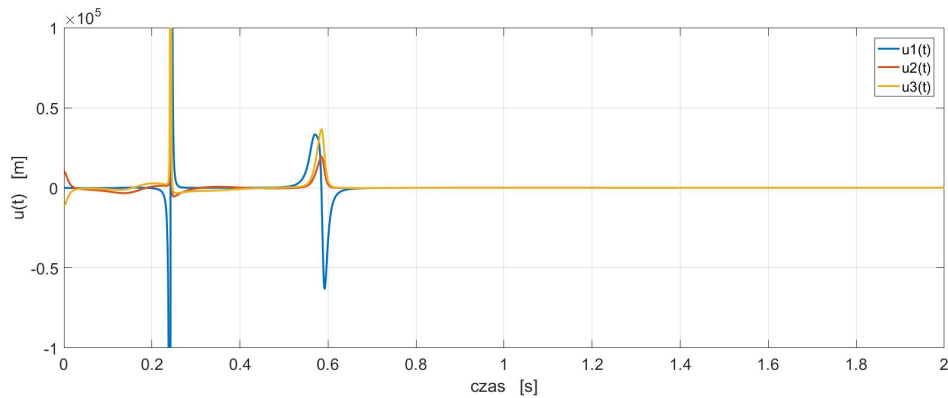
- zależności położenia efektora w przestrzeni od czasu,
- zależności uchybu położenia od czasu.
- sygnały sterujące $u(t)$

Najpierw poprowadzono efektor wzdłuż osi x , po linii prostej. Ruch ten opisany jest zależnościami $x_d(t) = \frac{t}{10}$, $y_d = 0$ oraz $z_d = 0$.

Na rysunku 5.1 przedstawiono charakterystyki trajektorii zadanej oraz trajektorii rzeczywistej dla współrzędnej x . Sygnały sterujące dla ruchu po linii prostej w kierunku osi OX zostały przedstawione na rysunku 5.2.

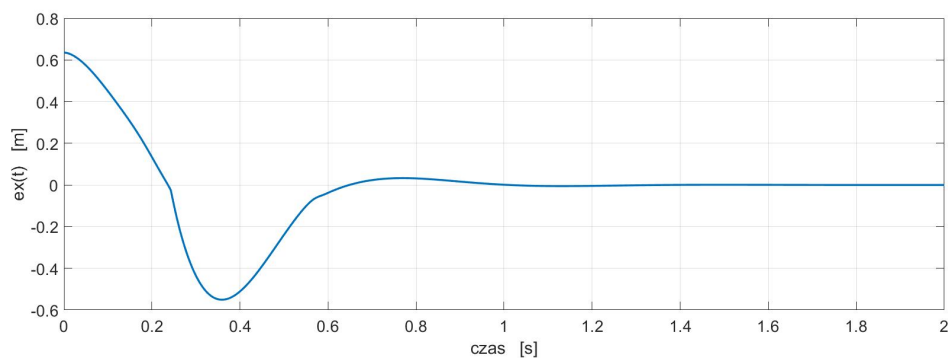


Rysunek 5.1 Przebieg zmiennej $x(t)$ dla trajektorii zadanej oraz rzeczywistej.

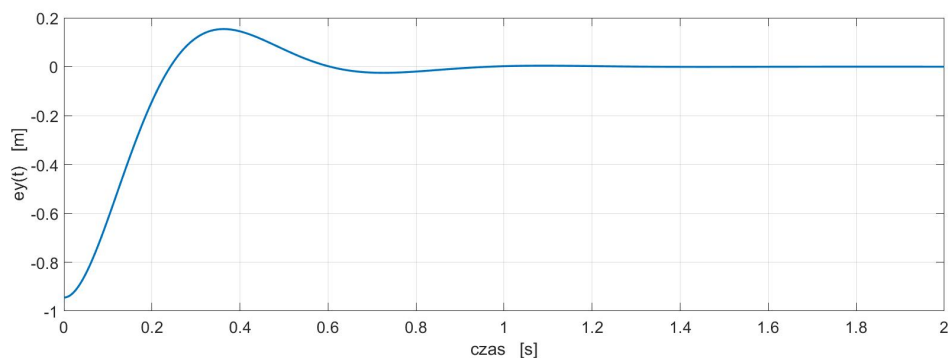


Rysunek 5.2 Sygnały sterujące $u(t)$ dla ruchu po linii prostej w kierunku osi OX.

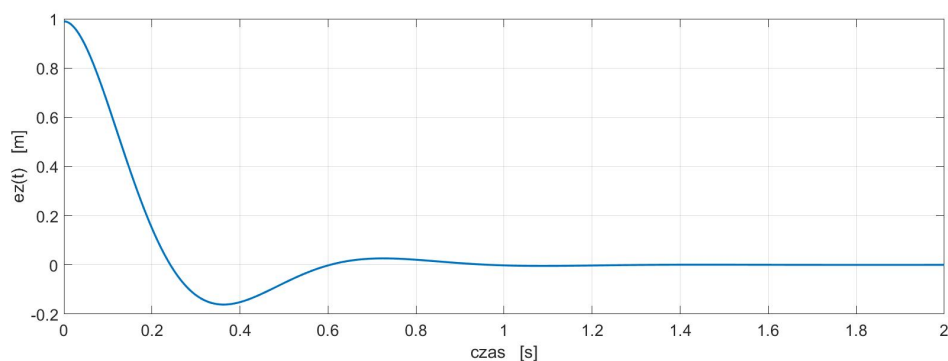
Uchyb śledzenia linii prostej wzdłuż osi OX dla wszystkich współrzędnych maleje do zera, co można zaobserwować na wykresach 5.3, 5.4 oraz 5.5.



Rysunek 5.3 Przebieg błędów $e_x(t)$ dla linii prostej wzdłuż osi OX.

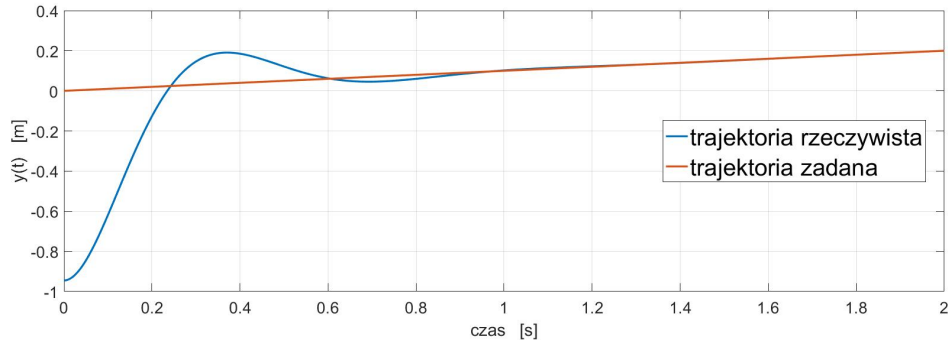


Rysunek 5.4 Przebieg błędów $e_y(t)$ dla linii prostej wzdłuż osi OX.

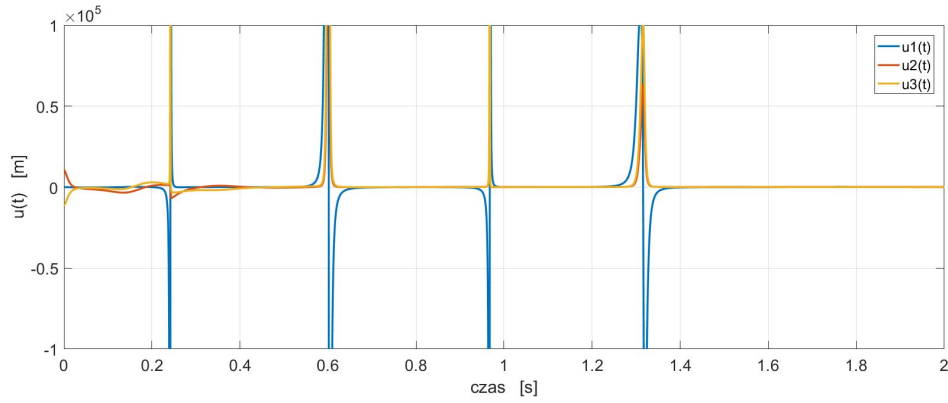


Rysunek 5.5 Przebieg błędów $e_z(t)$ dla linii prostej wzdłuż osi OX.

Na rysunku 5.6 przedstawiono charakterystykę $y(t)$ dla ruchu po linii prostej w kierunku osi OY. Ruch ten opisany jest zależnościami $x_d(t) = 0$, $y_d = \frac{t}{10}$ oraz $z_d = 0$. Tak samo jak w przypadku ruchu po linii prostej w kierunku osi OX wykreślono także zależności sygnałów sterujących od czasu na rysunku 5.7.

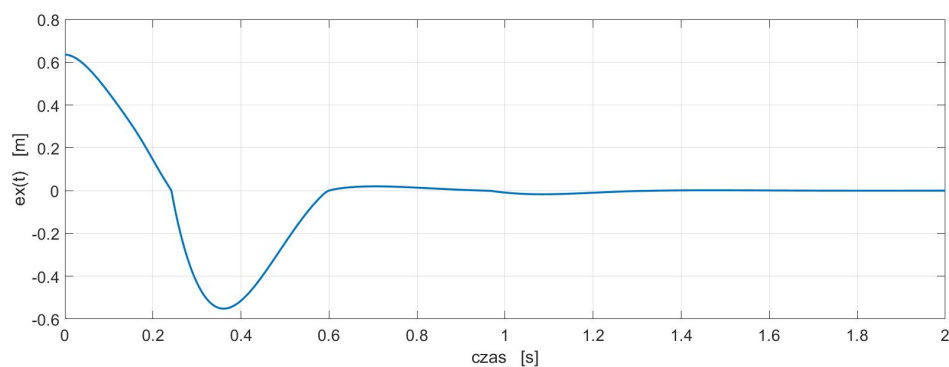


Rysunek 5.6 Przebieg zmiennej $y(t)$ dla trajektorii zadanej oraz rzeczywistej.

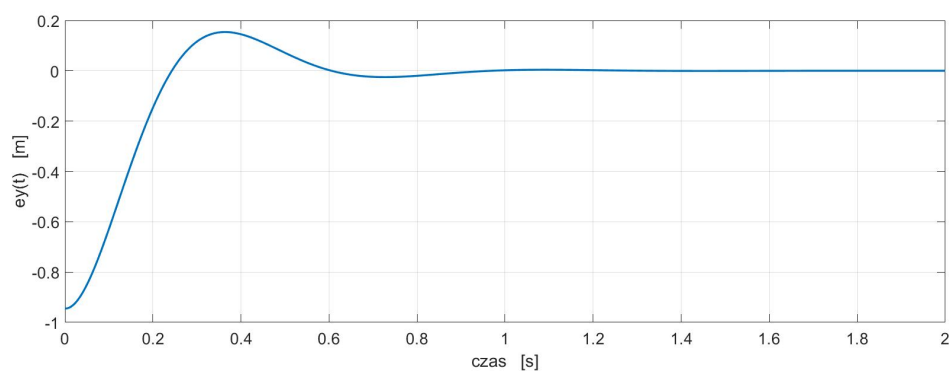


Rysunek 5.7 Sygnały sterujące $u(t)$ dla ruchu po linii prostej w kierunku osi OY.

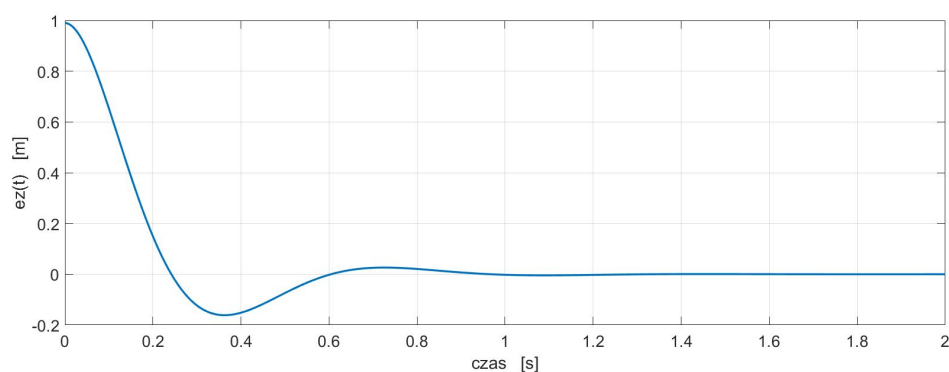
Uchyb śledzenia linii prostej wzdłuż osi OY dla wszystkich współrzędnych maleje do zera, co można zaobserwować na wykresach 5.8, 5.9 oraz 5.10.



Rysunek 5.8 Przebieg błędów $e_x(t)$ dla linii prostej wzdłuż osi OY.

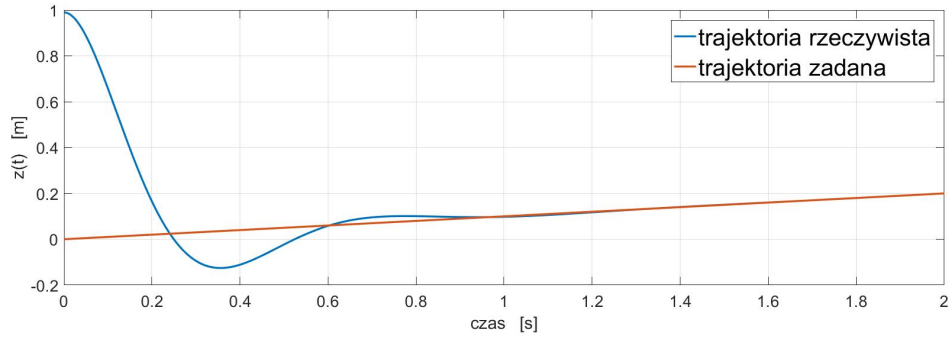


Rysunek 5.9 Przebieg błędów $e_y(t)$ dla linii prostej wzdłuż osi OY.

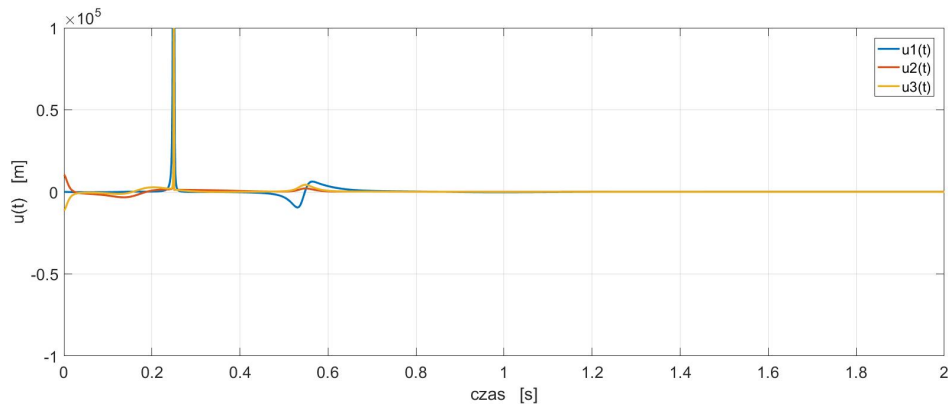


Rysunek 5.10 Przebieg błędów $e_z(t)$ dla linii prostej wzdłuż osi OY.

Na końcu poprowadzono efektor po linii prostej w kierunku osi OZ i sprawdzono charakterystyki trajektorii $z(t)$. Ruch ten opisany jest zależnościami $x_d(t) = 0$, $y_d = 0$ oraz $z_d = \frac{t}{10}$. Trajektorię zadaną i rzeczywistą przedstawiono na rysunku 5.11. Na rysunku 5.12 zaprezentowano sygnały sterujące dla tego ruchu.

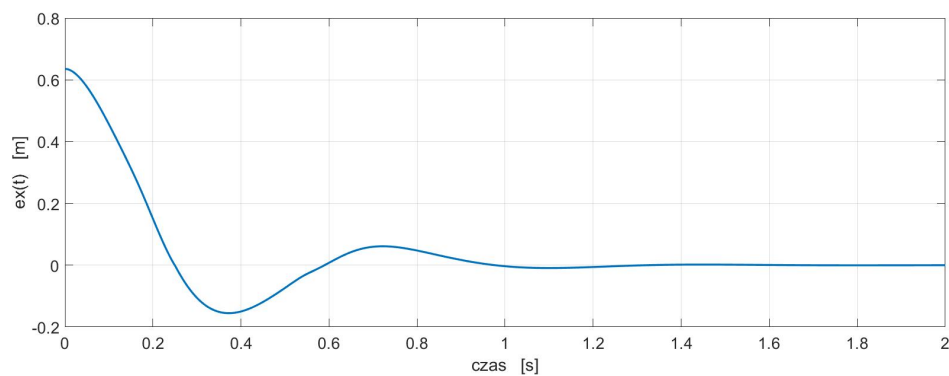


Rysunek 5.11 Przebieg zmiennej $z(t)$ dla trajektorii zadanej oraz rzeczywistej.

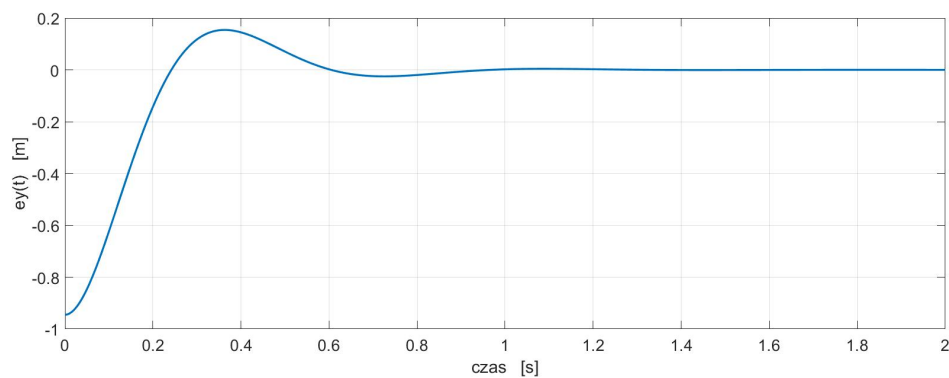


Rysunek 5.12 Sygnały sterujące $u(t)$ dla ruchu po linii prostej w kierunku osi OZ.

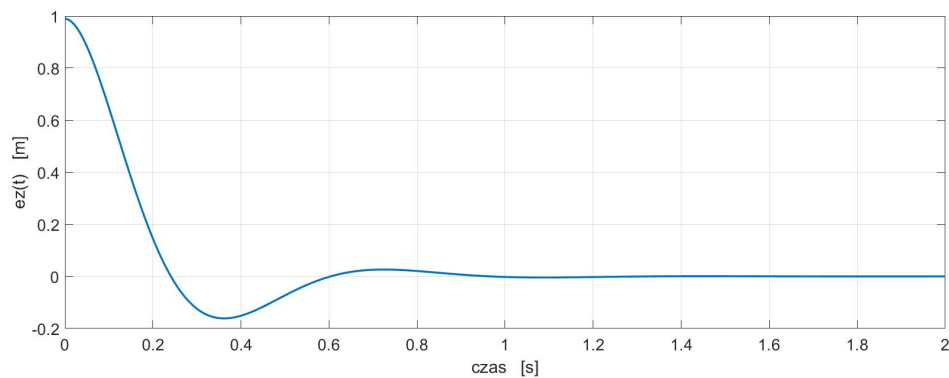
Uchyb śledzenia linii prostej wzdłuż osi OZ dla wszystkich współrzędnych maleje do zera, co można zaobserwować na wykresach 5.13, 5.14 oraz 5.15.



Rysunek 5.13 Przebieg błędów $e_x(t)$ dla linii prostej wzdłuż osi OZ.



Rysunek 5.14 Przebieg błędów $e_y(t)$ dla linii prostej wzdłuż osi OZ.



Rysunek 5.15 Przebieg błędów $e_z(t)$ dla linii prostej wzdłuż osi OZ.

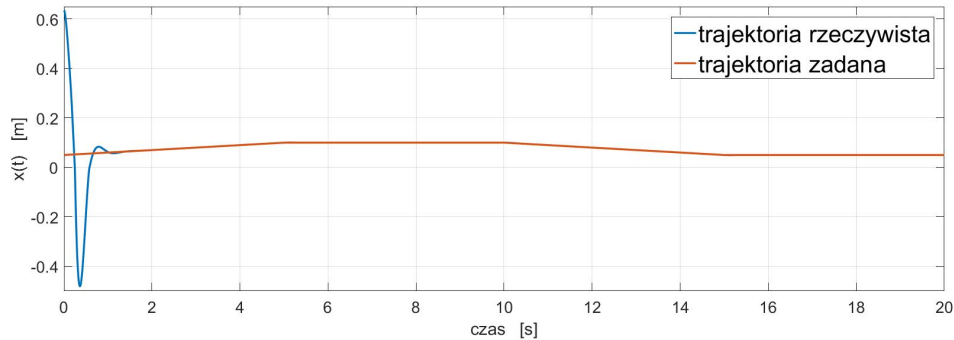
5.3 Ruch po trajektorii w kształcie kwadratu

Wraz z opanowaniem sterowania po liniach prostych, można przystąpić do trajektorii efektora składających się z kilku odcinków. Prostim przykładem takiego ruchu jest ruch efektora po trajektorii o kształcie kwadratu. Ruch ten będzie wykonywany na płaszczyźnie XY dla wartości $z = 0$. Efektor ma rozpocząć swój ruch z punktu $L_0 = (0.64; -0.94; 0)$. Następnie nakreśli kształt kwadratu o boku 50 cm. Zaprogramowanie takiego ruchu w Rapid dla np. manipulatora IRb-1400 wymagałoby użycia polecenia *MoveL*. Wymusza ta funkcja ruch efektora manipulatora po linii prostej do danego punktu. Przykłady fragment kodu

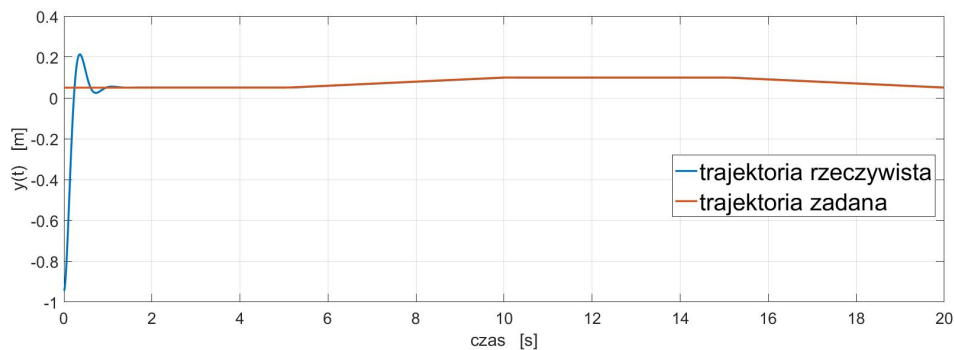
```
1 MoveL p1, v1000\T:=5, fine, tool0;
2 MoveL p2, v1000\T:=5, fine, tool0;
3 MoveL p3, v1000\T:=5, fine, tool0;
4 MoveL p4, v1000\T:=5, fine, tool0;
```

gdzie każemy wykonywać manipulatorowi ruchy efektem po punktach $p1$, $p2$, $p3$, $p4$ w ciągu 20 sekund.

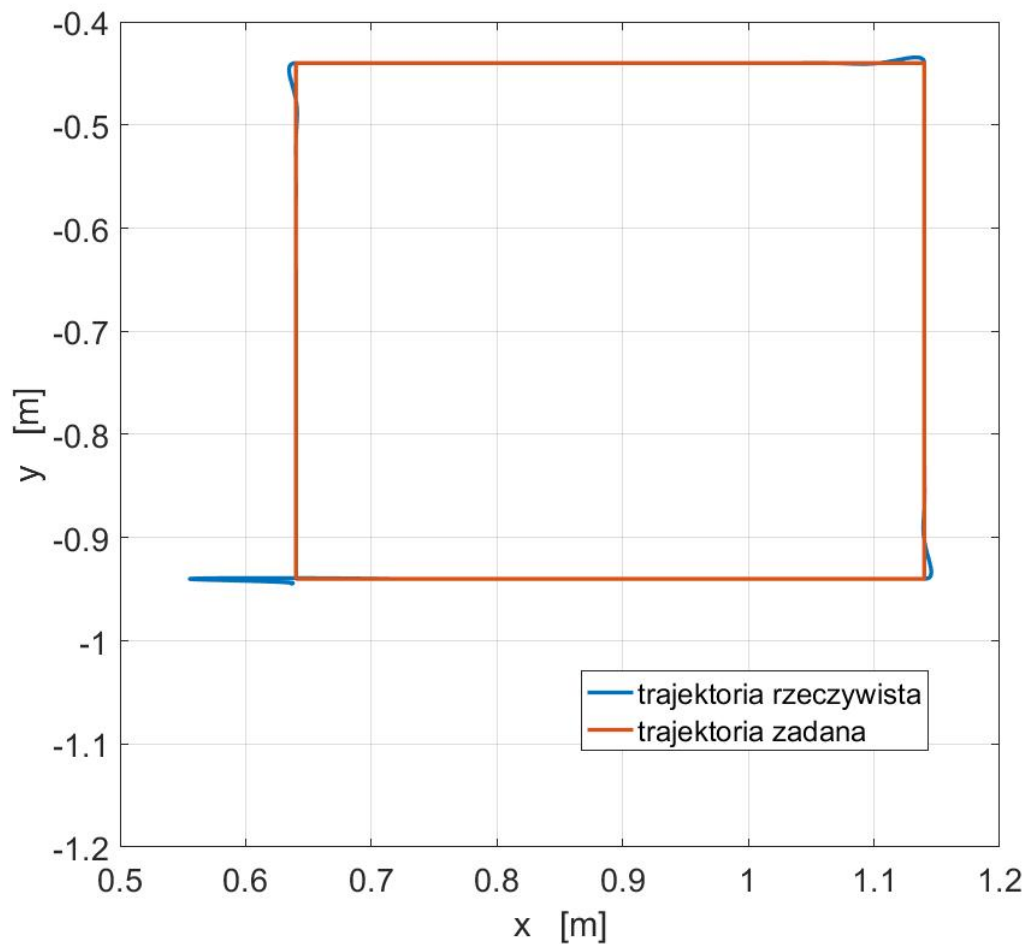
Wyniki pomiaru zależności współrzędnych położenia efektora w przestrzeni roboczej od czasu przedstawiono na wykresach 5.16 oraz 5.17. Z kolei zależność współrzędnej y od współrzędnej x , zarówno dla trajektorii zadanej jak i trajektorii rzeczywistej pokazano na rysunku 5.18.



Rysunek 5.16 Przebieg zmiennej $z(t)$ dla trajektorii zadanej oraz rzeczywistej.

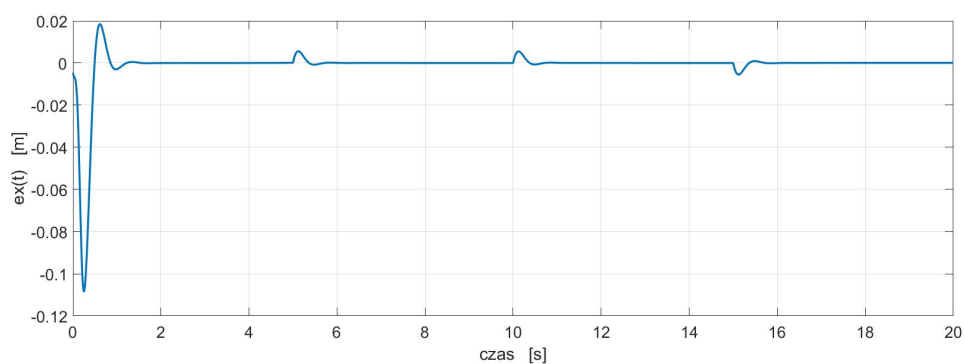


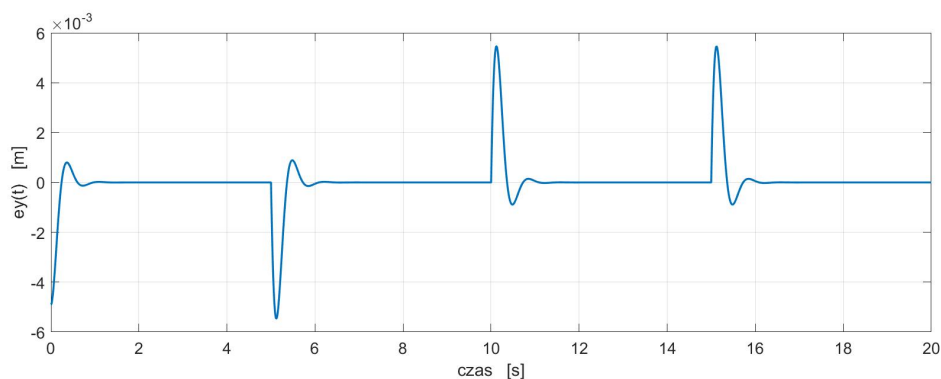
Rysunek 5.17 Sygnały sterujące $u(t)$ dla ruchu po linii prostej w kierunku osi OZ .



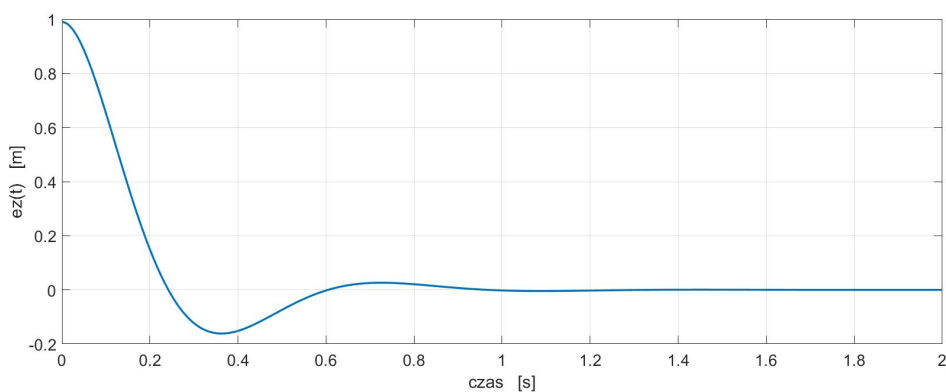
Rysunek 5.18 Trajektorie efektora w przestrzeni XY.

Następnie zmierzono zależność uchybu sterowania dla ruchu po kwadracie. Wynikowe charakterystyki przedstawiono na wykresach 5.19, 5.20 oraz 5.21.

Rysunek 5.19 Przebieg błędu $e_x(t)$ dla ruchu po kwadracie.



Rysunek 5.20 Przebieg błędów $e_y(t)$ dla ruchu po kwadracie.

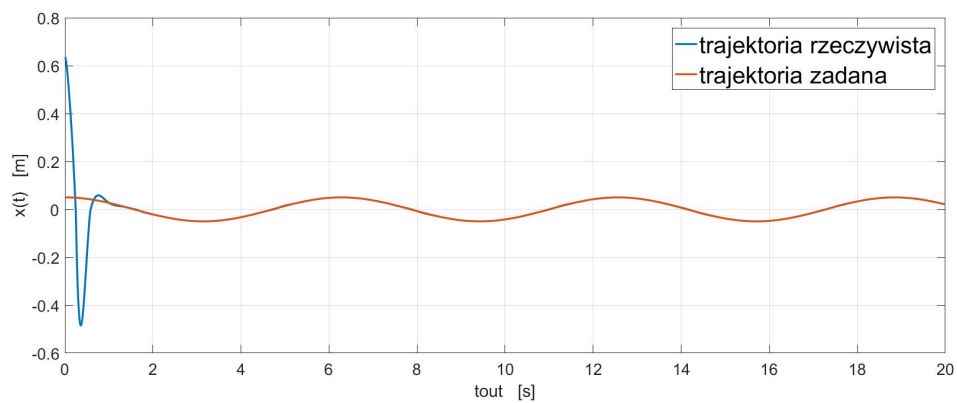


Rysunek 5.21 Przebieg błędów $e_z(t)$ dla ruchu po kwadracie.

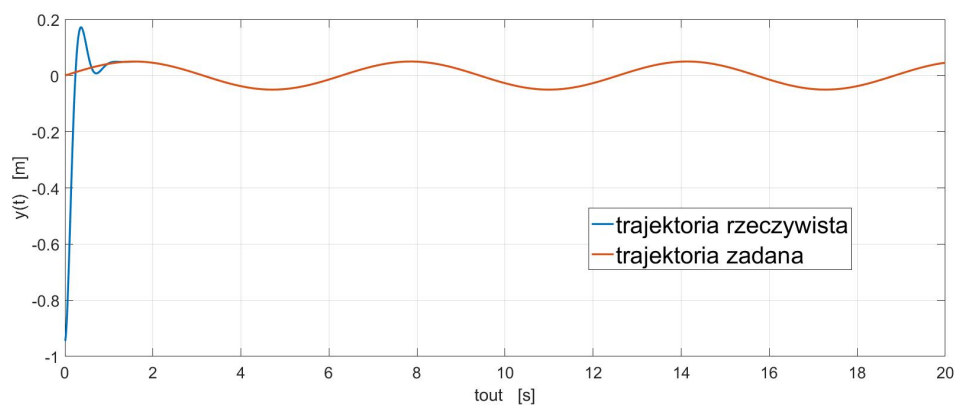
5.4 Ruch śrubowy

Ostatecznym celem zbadania skuteczności algorytmu odsprężania wejściowo-wyjściowego miało być poprowadzenie efektora ruchem śrubowym określonej układem równań (4.1). Zależność współrzędnych trajektorii zarówno zadanej jak i rzeczywistej przedstawiono na wykresach 5.22, 5.23 oraz 5.24.

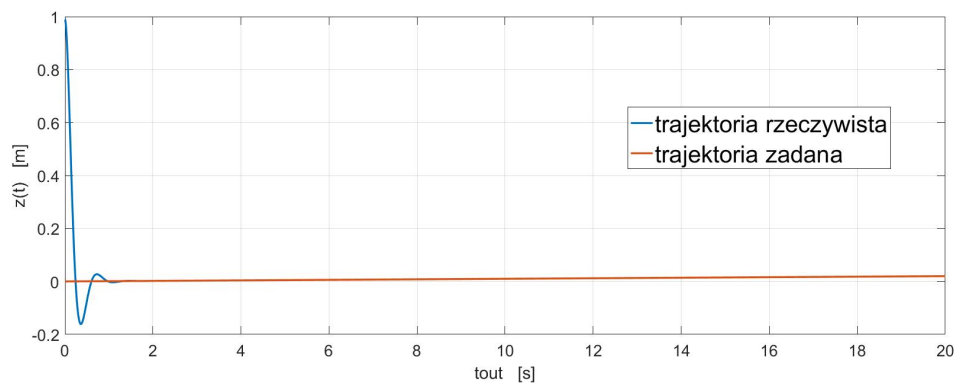
W celu dobrej wizualizacji sterowania efekтором dla tej trajektorii, zastosowano trójwymiarowy wykres 5.25. Odzwierciedla on dobrze trajektorię ruchu chwytaka w przestrzeni roboczej.



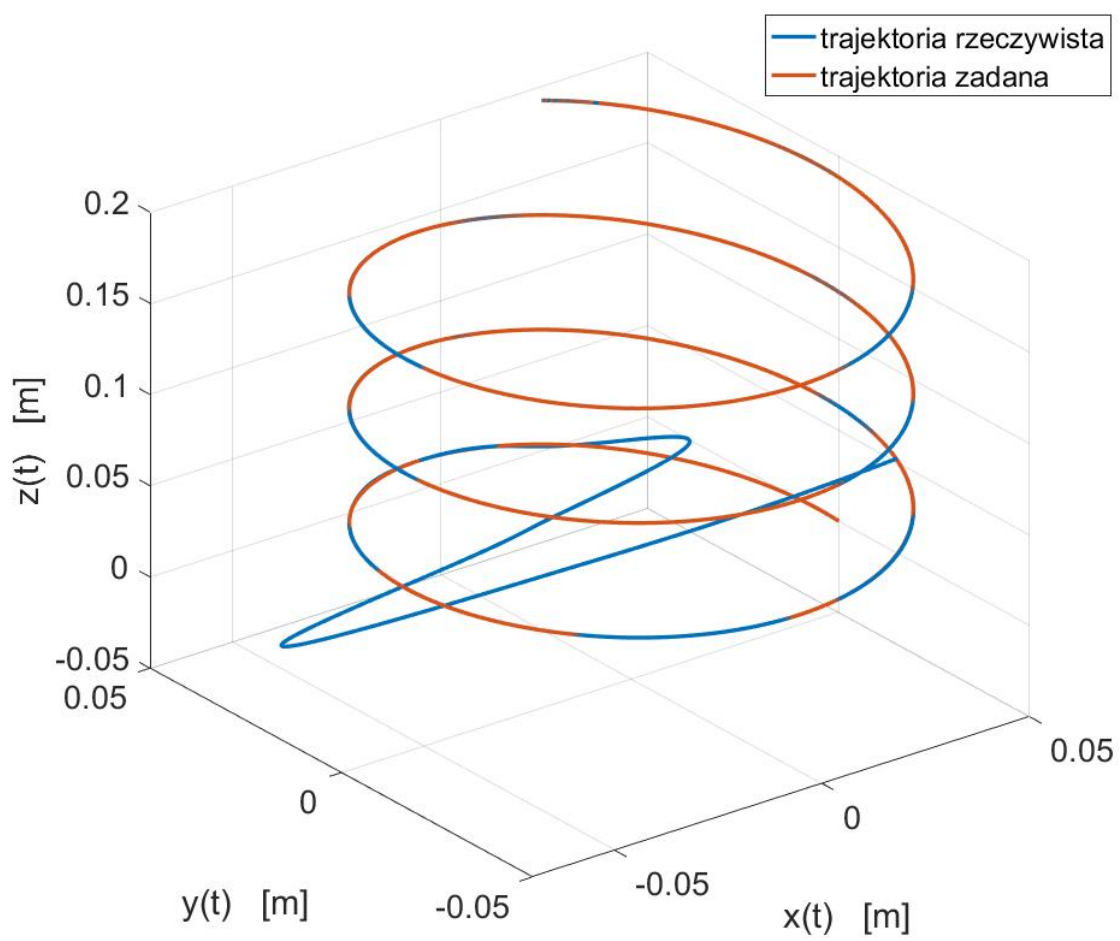
Rysunek 5.22 Przebieg zmiennej $x(t)$ ruchu śrubowego dla trajektorii zadanej oraz rzeczywistej.



Rysunek 5.23 Przebieg zmiennej $y(t)$ ruchu śrubowego dla trajektorii zadanej oraz rzeczywistej.



Rysunek 5.24 Przebieg zmiennej $z(t)$ ruchu śrubowego dla trajektorii zadanej oraz rzeczywistej.



Rysunek 5.25 Trajektoria chwytaka w przestrzeni XYZ dla ruchu śrubowego.

Rozdział 6

Podsumowanie

W pracy stworzono model manipulatora IRb-6, w środowisku Matlab/Simulink. Bazując na bibliografii opisano dokładnie jego dynamikę oraz kinematykę, które miały być podstawą do przeprowadzenia symulacji.

Następnie zrealizowano implementację algorytmu odsprzęgania wejściowo-wyjściowego. Pomyślnie stworzono afiniczny układ sterowania. Jednocześnie dla tego układu wyliczono konfiguracje osobliwe, których unikano w sterowaniu na etapie badań symulacyjnych. Gdy udało się zaimplementować sterowanie odsprzęgające i linearyzujące, utworzono w środowisku symulacyjnym generator trajektorii zadanej oraz regulację PD.

Na podstawie tak prawidłowo utworzonego sterowania z regulacją przeprowadzono badania symulacyjne. Przebadano ruch prostoliniowy w kierunku poszczególnych osi OX, OY oraz OZ. Potem sprawdzono skuteczność sterowania dla ruchu po kwadracie na płaszczyźnie XY. Na końcu dokonano symulacji ruchu efektora po trajektorii śruby. Za każdym razem sprawdzano czy manipulator nie znajduje się w konfiguracji osobliwej.

Manipulator wykonywał odpowiednio wszystkie zadanie trajektorie. Wszystkie badane przebiegi uchybu dążyły do zera. Większe początkowe wartości uchybów często wynikały z oddalenia początkowego położenia efektora od trajektorii zadanej. Jednak różnica ta była szybko niwelowana. W ruchu efektora po kwadracie można zaobserwować lekkie wzrosty wartości uchybów przy każdej nagłej zmianie kierunku poruszania się chwytaka. Wynikają one jednak ze skokowych zmian wartości sygnałów sterowania. W trakcie pozostałego czasu wykonywania tej trajektorii przez manipulator uchyb utrzymywał wartość zero. Pomyślnie zaimplementowano na modelu robota IRb-6 algorytm odsprzęgania wejściowo-wyjściowego. **Cel pracy został zrealizowany.**

Bibliografia

- [1] K. Tchoń, A. Mazur, I. Dulęba, R. Hossa, R. Muszyński *Manipulatory i roboty mobilne. Modele, planowanie ruchu, sterowanie*. Akademicka Oficyna Wydawnicza PLJ, Warszawa, 2000.
- [2] M. Spong, S. Hutchinson, M. Vidyasagar, *Dynamika i sterowanie robotów*. WNT, Warszawa, 1997.
- [3] A. Isidori, *Nonlinear Control Systems*. Springer, Londyn, 1995.
- [4] A. Gosiewski i in., *Badanie własności dynamicznych oraz projekt regulacji nadążnej dla robotów IRb-6 oraz IRb-60*. Raport IA Politechniki Warszawskiej, Warszawa, 1984.
- [5] T. Szkodny, *Modelowanie i symulacja ruchu manipulatorów robotów przemysłowych*. Wydawnictwo Politechniki Śląskiej, Gliwice, 2004.