

Politechnika Warszawska
Wydział Elektroniki i Technik Informacyjnych
Instytut Automatyki i Informatyki Stosowanej

Praca Dyplomowa Magisterska

Anna Maria Sibilska

*Porównanie metody wykorzystującej proste i
odwrotne zadanie kinematyki oraz jacobian
do sterowania manipulatorem*

Promotor
prof. nzw. dr hab. Cezary Zieliński

29 sierpnia 2007
Warszawa

Ogromne podziękowania panu prof. nzw. dr hab. Cezaremu Zielińskiemu za niezastąpioną pomoc i wielką dozę cierpliwości, a także całemu zespołowi laboratoryjnemu, w szczególności mgr inż. Tomkowi Winiarskiemu, mgr inż. Maćkowi Staniakowi i mgr inż. Tomaszowi Kornucie za pomoc w zmaganiach laboratoryjnych.



Życiorys

Anna Maria Sibilska urodzona 22 lipca 1982 roku w Warszawie. W roku 1997 ukończyła szkołę podstawową nr. 316 w Warszawie. W czasie nauki kilkakrotnie została laureatką olimpiad matematycznych. W 2001 roku zdała z wyróżnieniem egzamin maturalny w III Liceum Ogólnokształcącym im. Gen. Józefa Sowińskiego w Warszawie. W tym samym roku rozpoczęła studia I stopnia na wydziale Elektroniki i Technik Informacyjnych Politechniki Warszawskiej. Od 2006 roku członek (student member) organizacji AIAA (American Institute of Aeronautics and Astronautics). Od tego czasu współautor poniższych prac: „Biomimic Sensors Guided Flight Stability and Control for Flapping Wings Autonomous Micro Air Vehicle (Entomopter)” 45th AIAA Aerospace Science Meeting, „Entomopter Manoeuvrability Enhancement by Deformations Control of Flexible Flapping Wings” 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference, „Experimental validation of mathematical model of autonomous gliding delivery system” 19th AIAA Aerodynamics Decelerator Systems Technology Conference and Seminar. W latach 2005 - 2006 podjęła pracę w branży Business Intelligence. Interesuje się muzyką, tańcem, jogą, podróżami, literaturą i kinematografią z gatunku science-fiction, a także elementami astronomii i fizyki teoretycznej.

EGZAMIN DYPLOMOWY

Termin egzaminu dyplomowego

Wynik egzaminu

Ogólny wynik studiów

Dodatkowe wnioski i uwagi Komisji

Streszczenie

Podstawowym zagadnieniem związanym z manipulatorami programowalnymi jest zadanie kinematyki, będące odwzorowaniem pomiędzy konfiguracją wewnętrzną robota, a pozycją końcówki roboczej. Niniejsza praca prezentuje dwa podejścia do tego problemu. Pierwsze z nich, zwane kinematyką pozycyjną, bazuje na geometrycznym opisie położenia układów współrzędnych związanych z poszczególnymi ogniwami manipulatora. Dzięki odpowiednim złożeniom przesunięć i rotacji tychże układów otrzymywane jest bezpośrednie przekształcenie pozycji ostatniego ogniwa w układzie bazowym. Drugie podejście, zwane kinematyką różniczkową, operuje na przekształceniu pochodnych tych wartości, a więc prędkości zmiennych przegubowych oraz wektorze prędkości liniowych i obrotowych końcówki roboczej. Podstawą tego odwzorowania jest jakobian, będący jedną z najszerzej stosowanych w robotyce wielkości. Opis każdej z powyższych metod poprzedzony został dogłębną analizą związanych z nimi podstaw teoretycznych, takich jak macierze przekształcenia jednorodnego, notacja Denavita-Hartenberga, czy operatory algebry przestrzennej.

Głównym celem zaprezentowanej pracy było wyprowadzenie zależności odpowiadających manipulatorowi IRp-6 o sześciu stopniach swobody, oraz wdrożenie w strukturę MRROC++ struktur sterowania opartych na kinematyce pozycyjnej i kinematyce różniczkowej.

Abstract

The fundamental task connected with programmable industrial robots is a kinematics problem, which forms a function between internal configuration and values of position and orientation of the end-effector in the Cartesian space. The following work represents two main approaches to this problem. First of them so-called positional kinematics, is based on geometrical description of coordinates connected with each of the robots links. By suitable sequence of shifts and rotations of those coordinates we get the direct transformation of the position of the end-effector in base Cartesian coordinate system. The second shown approach, named differential kinematics, operates on transformation of derivative of those values, the joint velocity vector and the end-effector linear and rotation velocity vector. The basis of this transformation is Jacobian matrix, which is one of the most commonly used in robotics quantity. Description of every from the mentioned-above methods is preceded by deep analysis of connected with them theoretical bases, such as homogeneous transformation matrix, Denavit-Hartenberg notation, or spatial algebra operators.

The main goal of presented work is to establish kinematics relations for IRp-6 industrial robot with six degrees of freedom and implement into MRROC++ structure the control structures based on positional and differential kinematics.

Spis treści

Wykaz oznaczeń	3
Wstęp	6
Zarys pracy	11
1 Opis manipulatora	13
1.1 Elementarne definicje	13
1.2 Manipulator jako łańcuch kinematyczny	15
1.3 Przekształcenia jednorodne	15
1.3.1 Przesunięcie równoległe	16
1.3.2 Macierz rotacji	16
1.3.3 Reprezentacje rotacji	18
1.4 Macierz przekształcenia jednorodnego	23
1.5 Macierz skośniesymetryczna	24
1.6 Operatory algebry przestrzennej	26
1.7 Notacja Denavita-Hartenberga	28
1.8 Typy manipulatorów	31
2 Kinematyka pozycyjna	35
2.1 Idea zadania kinematyki pozycyjnej	35
2.1.1 Proste zadanie kinematyki	35
2.1.2 Odwrotne zadanie kinematyki	36
2.2 Opis manipulatora IRp-6	37
2.2.1 Opis ogniw manipulatora IRp-6	37
2.2.2 Parametry Denavita-Hartenberga dla IRp-6	39
2.3 Równania kinematyki prostej manipulatora IRp-6	39
2.4 Równania kinematyki odwrotnej manipulatora IRp-6	41
3 Kinematyka różniczkowa	46
3.1 Sterowanie oparte na prędkości	46
3.2 Wyprowadzanie Jakobianu - MPJ	47
3.2.1 Prędkość obrotowa	47
3.2.2 Prędkość Liniowa	49

3.3	Wyprowadzanie Jakobianu - OAP	50
3.4	Postać jakobianu dla manipulatora IRp-6	51
3.5	Jakobian analityczny	60
3.6	Struktura sterowania oparta na odwrotności jakobianu	62
3.7	Odwrotność jakobianu	66
3.7.1	Postać analityczna	66
3.7.2	Metoda LU	69
3.7.3	Porównanie metody analitycznej i numerycznej	72
3.8	Transpozycja jakobianu	73
3.9	Konfiguracje osobliwe manipulatora	75
4	Wyniki symulacji w środowisku Matlab	77
4.1	Uchyb dla jednego kroku prostego zadania kinematyki różniczkowej	77
4.2	Uchyb dla jednego kroku odwrotnego zadania kinematyki różniczkowej	83
4.3	Pętla sterowania oparta na odwrotności jakobianu J^{-1}	86
4.4	Pętla sterowania oparta na transpozycji jakobianu J^T	91
5	Wdrożenie wyznaczonych metod w strukturę MRROC++	97
6	Podsumowanie	101

Wykaz ważniejszych oznaczeń matematycznych

- $\mathbf{a}_k, \mathbf{d}_k, \boldsymbol{\alpha}_k, \boldsymbol{\theta}_k$ - parametry notacji (lub zmodyfikowanej notacji) Denavita-Hartenberga dla przegubu k . Współrzędna uogólniona q_k związana z tym przegubem określona jest zależnością (1), przy czym $\sigma_k = 0$ dla przegubu obrotowego lub $\sigma_k = 1$ dla przegubu translacyjnego.

$$q_k = (1 - \sigma_k)\theta_k + \sigma_k d_k \quad (1)$$

- $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^{N \times 1}$ - wektory położenia, prędkości i przyspieszenia współrzędnych uogólnionych kolejnych złącz manipulatora.
- N - liczba stopni swobody otwartego łańcucha kinematycznego.
- O_k - początek układu współrzędnych związanego z ogniwem k .
- $\mathbf{l}_B^A \in \mathbb{R}^{3 \times 1}$ - wektor położenia punktu B względem układu A
- $\mathbf{R}_{k-1}^k \in \mathbb{R}^{3 \times 3}$ - macierz rotacji opisująca orientację układu O_{k-1} względem układu O_k .
- $\mathbf{R}_{b,\theta} \in \mathbb{R}^{3 \times 3}$ - macierz rotacji opisująca obrót o kąt θ , względem osi b .
- $\mathbf{n}, \mathbf{o}, \mathbf{a} \in \mathbb{R}^{3 \times 1}$ - wektory składowe macierzy rotacji określające układ ortonormalny w przestrzeni kartezjańskiej. Wektory te nazywane są odpowiednio wektorem normalnym, orientacji i zbliżenia.

$$R_{k-1}^k = (n \ o \ a) \quad (2)$$

- $\mathbf{T}_{k-1}^k \in \mathbb{R}^{4 \times 4}$ - macierz przekształcenia jednorodnego pomiędzy układem O_{k-1} a O_k . Na postać macierzy tej (3) składają się cztery elementy: macierz rotacji R_{k-1}^k i wektor przesunięcia \mathbf{l}_{k-1}^k oraz perspektywa $p \in \mathbb{R}^{1 \times 3}$ i skalowanie $s \in \mathbb{R}$.

$$T_{k-1}^k = \begin{pmatrix} R_{k-1}^k & \mathbf{l}_{k-1}^k \\ p & s \end{pmatrix} \quad (3)$$

- $\mathbf{S}(\mathbf{x}) \in \mathbb{R}^{3 \times 3}$ - macierz skośniesymetryczna wektora $x \in \mathbb{R}^{3 \times 1}$, postaci (4). Macierz ta umożliwia zapisanie iloczynu wektorowego w postaci: $x \times y = S(x)y$.

$$S(x) = \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix} \quad (4)$$

- $\mathbf{A}(k-1, k) \in \mathbb{R}^{6 \times 6}$ - przestrzenna macierz kosinusów kierunkowych będąca odwzorowaniem układu O_{k-1} na O_k o postaci (5), gdzie R_k^{k-1} jest macierzą rotacji układów, natomiast $O \in \mathbb{R}^{3 \times 3}$ - macierzą o elementach zerowych.

$$A(k-1, k) = \begin{pmatrix} R_k^{k-1} & O \\ O & R_k^{k-1} \end{pmatrix} \quad (5)$$

Własności macierzy $A(k-1, k)$:

$$A(k-1, k)^{-1} = \begin{pmatrix} R_k^{k-1} & O \\ O & R_k^{k-1} \end{pmatrix} \quad (6)$$

$$\forall_{j < k} A(j, k) = A(j, j+1)A(j+1, j+2) \dots A(k-1, k) \quad (7)$$

- $\Phi(k-1, k) \in \mathbb{R}^{6 \times 6}$ - przestrzenna macierz odległości pomiędzy początkiem układu O_{k-1} a O_k wyznaczona z zależności (8), gdzie $U \in \mathbb{R}^{3 \times 3}$ jest macierzą jednostkową, $O \in \mathbb{R}^{3 \times 3}$ macierzą zerową, a $S(l_k^{k-1})$ macierzą skośniesymetryczną wektora odległości między tymi układami.

$$\Phi(k-1, k) = \begin{pmatrix} U & S(l_k^{k-1}) \\ O & U \end{pmatrix} \quad (8)$$

Własności macierzy $\Phi(k-1, k)$:

$$\Phi(j, k)^{-1} = \Phi(k, j) \quad (9)$$

$$\Phi(j, j) = U_{6 \times 6} \quad (10)$$

$$\forall_{j < k} \Phi(j, k) = \Phi(j, j+1)\Phi(j+1, j+2) \dots \Phi(k-1, k) \quad (11)$$

- $\mathbf{h}_k \in \mathbb{R}^{3 \times 1}$ - wektor jednostkowy wyznaczający oś obrotu złącza k .

- $\mathbf{H}_k \in \mathbb{R}^{6 \times 1}$ - wektor opisujący odwzorowanie względnej prędkości złącza k . W zależności od rodzaju przegubu przyjmuje on jedną z następujących postaci (12).

$$H_k = \begin{cases} \begin{pmatrix} h_k^T & O \end{pmatrix}^T & dla \quad \sigma_k = 0 \\ \begin{pmatrix} O & h_k^T \end{pmatrix}^T & dla \quad \sigma_k = 1 \end{cases} \quad (12)$$

- $\mathbf{J} \in \mathbb{R}^{6 \times N}$ - jacobian geometryczny manipulatora.
- $\mathbf{J}_A \in \mathbb{R}^{6 \times N}$ - jacobian analityczny manipulatora.
- \mathbf{t} - czas.
- $\mathbf{v}_k \in \mathbb{R}^{3 \times 1}$ - wektor prędkości liniowej układu O_k .
- $\boldsymbol{\omega}_k \in \mathbb{R}^{3 \times 1}$ - wektor prędkości kątowej układu O_k .
- $\mathbf{V}_k \in \mathbb{R}^{6 \times 1}$ - przestrzenny wektor prędkości układu O_k .

$$V_k = \begin{pmatrix} \omega_k \\ v_k \end{pmatrix} \quad (13)$$

Wstęp

Robotyka jest bardzo młodą, interdyscyplinarną dziedziną nauki, łączącą elementy mechaniki, automatyki, elektroniki, sensoryki, cybernetyki oraz informatyki. Niniejsza praca prezentuje czysto mechaniczne i automatyczne aspekty związane z robotyką, w odniesieniu do sterowania manipulatorem IRp-6. Na wstępie rozprawy zamieszczone zostało ogólne spojrzenie na dziedzinę nauki, jaką jest robotyka. Przedstawiona została pokrótce jej historia oraz przegląd najciekawszych, w jej zakresie osiągnięć.

Etymologia określenia robot wiąże się z czeskim dramatopisarzem Carelem Čapek. W napisanym przez niego w 1921 r. utopijno-fantastycznym dramacie „Roboty Uniwersalne Rossuma” po raz pierwszy użyto terminu robot w odniesieniu do maszyn-niewolników. Pierwotnie słowo to oznaczało w języku czeskim ciężką, w domyśle niewolniczą pracę. Geneza terminu robotyka również związana jest z literaturą science fiction. W 1941 r. Isaac Asimov użył tego terminu w opowiadaniu „Runaround”, które w późniejszych latach weszło w skład zekranizowanej w 2004 r. książki „Ja, Robot”¹.

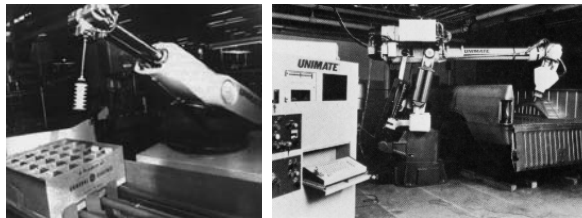
Początków idei robotyki można by doszukiwać się już w starożytności: w mitologicznych mechanicznych służących zbudowanych w kuźni Hefajstosa, micie o Pigmalionie i Galatei, czy wynalazku klepsydry i zegara wodnego. Pierwsze plany budowy robota wyszły około 1495 r. spod ręki Leonarda Da Vinci, a zbudowany na ich podstawie przez Johna D. Enderle w 2004 r. mechaniczny rycerz zdolny jest do poruszania rąk, nóg, głowy oraz szczęki. Prawdopodobnie pierwszym działającym robotem w historii ludzkości był android grający na flecie, skonstruowany przez Jacques de Vaucanson w 1738 r. Narodziny jednak samej robotyki jako nauki datuje się na rok 1961, kiedy to General Motors zainstalował, przy obsłudze wysokociśnieniowej maszyny odlewniczej pierwszy manipulator o nazwie Unimate produkcji G.C. Devol i J.S.Engelberger.

Roboty generalnie podzielić można na manipulatory i roboty mobilne. Do drugiej z tych klas zaliczamy wszelkie roboty mające zdolność swobodnego przemieszczania się w przestrzeni. Są to przede wszystkim roboty kołowe, takie jak przeznaczony do eksploracji księżycy Ratler i pozostawione na Marsie Łaziki Spirit i Opportunity oraz roboty kroczące. Najciekawszymi mechanizmami tego

¹W dziele tym przedstawiono również słynne trzy prawa robotyki, oraz ideę pozytronowego mózgu, posiadanego przez Datę - bohatera serialu „Star-Trek”.



Rysunek 1: Robot humanoidalny powstały według planów Leonarda Da Vinci [3]



Rysunek 2: Robot Unimate [14]

typu są oczywiście roboty humanoidalne, takie jak będący „medialnym logiem” robotyki robot Hondy Asimo, Qrio wyprodukowany przez Sony i niemieckiej produkcji Lara, w której konstrukcji serwomechanizmy zastąpiono sztucznymi mięśniami. Do robotów kroczących zaliczymy również robota zabawkę Aibo oraz, swą potężną posturą przypominającego plan „Gwiezdnych wojen”, robota LandWalker, który prawdopodobnie, w przyszłości znajdzie zastosowanie w operacjach militarnych. Niektóre roboty mobilne skonstruowane zostały z myślą o środowisku wodnym lub powietrznym. Bezzałogowe samoloty takie jak Raven i Predator znalazły szerokie uznanie w amerykańskich siłach wojskowych, a Madelaine inspirowana pływającymi prehistorycznymi gadami swobodnie przemieszcza się w zbiornikach wodnych badając stan podwodnej fauny i flory. Warto w tym miejscu wspomnieć również o robocie będącym dowodem tego, iż robotyka coraz szybciej wkracza w życie codzienne człowieka, a mianowicie dostępnym już w Polsce, za rozsądną cenę, autonomicznym odkurzaczu Roomba firmy iRobot.

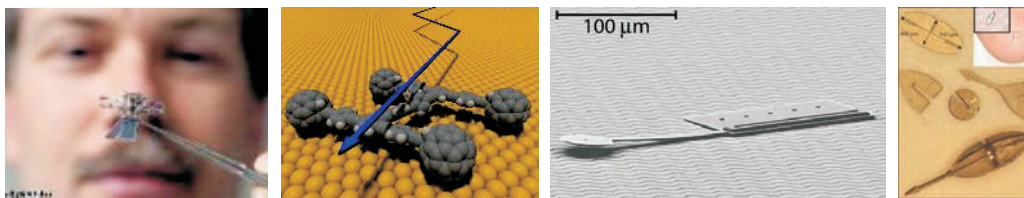
Ze względu na mały rozmiar i odmienność związanych z nimi problemów technicznych, do odrębnej klasy zaliczamy mikro i nano roboty. Mikroroboty o wielkości rzędu kilkudziesięciu *mm* wzorowane są, w większości na owadach. Roboty te projektowane są z myślą o misjach zwiadowczych i szpiegowskich, oraz diagnozie chorób układu pokarmowego. Jeszcze mniejszą, a bardzo obiecującą, grupą robotów są nanoroboty². Nanity, o rozmiarze niewiele przekraczającym wirusa, mogą zostać wpuszczone do ludzkiego krwiobiegu i przeprowadzić skomplikowane operacje medyczne na poziomie komórkowym. Jak do tej pory stworzenie zaawansowanej nanomaszyny leży poza zasięgiem współczesnej techniki,

²Nanoroboty nazywane są również nanobotami i nanitami.



Rysunek 3: Przykłady robotów mobilnych: Roomba, Asimo, Land-Walker, Lara, Spirit, Predator, Aibo [2]

jednak naukowcy są w stanie konstruować i testować proste urządzenia w tej skali. Najmniejszym, jak do tej pory, zaprojektowanym robotem jest zbudowany na Uniwersytecie Rice atomowy samochód wielkości $3\text{-}4\mu\text{m}$.



Rysunek 4: Przykłady mikro i nanorobotów[2]

Może najmniej wpływającą na wyobraźnię, jednak najszerzej obecnie stosowaną grupę robotów stanowią manipulatory, zwane również robotami przemysłowymi. Zdecydowana ich większość przeznaczona jest do pracy na fabrycznych liniach produkcyjnych, przy wykonywaniu wszelkiego rodzaju powtarzalnych, wymagających dużej siły lub precyzji i niebezpiecznych dla człowieka czynności, takich jak spawanie, montaż i obróbka. Drugim miejscem, zaraz po halach fabrycznych, w którym można napotkać manipulatory są sale operacyjne. Wysoko wyspecjalizowane mechaniczne ramiona coraz częściej pomagają lekarzom w przeprowadzeniu szeregu skomplikowanych operacji chirurgicznych, dzięki czemu zabiegi te przebiegają z coraz większą precyzją i wydajnością, przy minimalizacji inwazyjności i ryzyka pacjenta. Najbardziej zaawansowanym, jak do tej pory, robotycznym sprzętem chirurgicznym jest skonstruowany na życzenie Pentagonu, robot da Vinci. Robot ten składa się z odrębnej konsoli, będącej interfejsem chirurga, oraz platformy, wyposażonej w troje ramion³ o siedmiu

³Dostępna jest również czteroramienna wersja robota Da Vinci.

stopniach swobody i zainstalowanej nad stołem operacyjnym. Robot Da Vinci stosowany jest głównie przy zabiegach w obrębie układu pokarmowego. Innymi, powszechnie znanymi, medycznymi systemami robotycznymi są rodzimej produkcji Robin Heart, wyspecjalizowany głównie w zakresie kardiochirurgii oraz Robodoc stosowany przy operacjach wszczepiania protez biodrowych.

Kolejnym polem zastosowań manipulatorów jest wojsko i oddziały antyterrorystyczne. Roboty tego typu, zamocowane na bazie jezdnej i charakteryzujące się dużą manewrowalnością, pomagają człowiekowi w neutralizacji ładunków wybuchowych i eksploracji nieznanych, trudno dostępnych terenów. Przykładami takich robotów są polskiej produkcji antyterrorystyczny robot Expert, czy stosowana w Iraku wersja EOD robota PacBot firmy iRobot.

Coraz więcej manipulatorów powstaje z myślą o zastosowaniu czysto domowym. Jednym z najbardziej innowatorskich robotów usługowych jest niewątpliwie MySpoon firmy Secom, będący sztucznym ramieniem zaprojektowanym do pomagania w jedzeniu ludziom, niezdolnym do używania własnych rąk. Innym przykładem „domowego manipulatora” jest skonstruowane przez szwajcarską firmę Neuronics ramię Katana. Robot ten dzięki powolnym ruchom i delikatnym uchwycie stał się doskonałym pomocnikiem człowieka w wszelkiego rodzaju mozolnych i czasochłonnych domowych zadaniach.



Rysunek 5: Przykłady manipulatorów: robot Da Vinci, Expert, Katana, My Spoon[2]

Elementarnym zagadnieniem związanym z manipulatorami programowalnymi jest zadanie kinematyki. Z matematycznego punktu widzenia zadanie to stanowi odwzorowanie między konfiguracją wewnętrzną⁴ robota a pozycją końcówki manipulatora. W zależności od kierunku tego odwzorowania mówimy o zadaniu prostym lub odwrotnym kinematyki. Zadanie proste, w którym zadane są wartości współrzędnych wewnętrznych, wylicza pozycję układu współrzędnych związanego z końcówką manipulatora. Zadanie odwrotne, z kolei, polega na wyznaczeniu wartości poszczególnych zmiennych przegubowych w funkcji zadanej pozycji końcówki roboczej. Prezentowana praca dotyczy jedynie szeregowej struktury manipulatorów, dla której zadanie proste jest z reguły trywialne i posiada jednoznaczne rozwiązanie. Natomiast zadanie odwrotne, dla tej struk-

⁴Poprzez konfigurację wewnętrzną rozumiemy tu wartości poszczególnych zmiennych przegubowych (zmiennych wewnętrznych), które jednoznacznie wynikają z ułożenia wałów silnika, a więc wielkości sterowalnych robota.

tury, jest znacznie trudniejsze, a jego rozwiązanie, o ile możliwe do wyznaczenia, w większości wypadków jest niejednoznaczne.

Niniejsza praca prezentuje dwa podejścia do problemu kinematyki. Pierwsze z nich, zwane kinematyką pozycyjną, bazuje na geometrycznym opisie położenia układów współrzędnych związanych z poszczególnymi ogniwami manipulatora. Dzięki odpowiednim złożeniom przesunięć i rotacji tychże układów otrzymywane jest bezpośrednie przekształcenie pozycji ostatniego ogniw w układzie bazowym. Drugą prezentowaną metodą jest kinematyka różniczkowa, która operuje na przekształceniu pochodnych tych wartości, a więc prędkościach zmieniających przegubowych, oraz wektorze prędkości liniowych i obrotowych końcówki roboczej. Podstawą tego odwzorowania jest jakobian, będący jedną z najważniejszych i najszerzej stosowanych w robotyce wielkości. W stosowanej przez robotyków nomenklaturze jakobianem nazywana jest macierz jakobiego, a nie jej wyznacznik. Struktura jakobianu poza problemem kinematyki prędkości stosowana jest na przykład w sterowaniu siłowym, lub wyznaczaniu trajektorii gładkich.

Celem prezentowanej pracy jest wyznaczenie i wdrożenie opisanych powyżej metod rozwiązania zagadnienia kinematyki, w strukturę sterowania manipulatorem IRp-6 o sześciu stopniach swobody. W ramach testów tych zagadnień zostały również przeprowadzone serie symulacji w środowisku Matlab.

Zarys Pracy

Rozdział 1

W rozdziale tym przedstawiono podstawową nomenklaturę stosowaną w robotyce, a następnie, w jej świetle, opisano strukturę manipulatora. Wprowadzenie tego opisu jest nieodzownym elementem dalszej analizy kinematycznej robota. Zostały tu szczegółowo omówione pojęcia takie jak para kinematyczna, współrzędne wewnętrzne czy łańcuch kinematyczny. W kolejnej części rozdziału przytoczono metody opisu położenia i orientacji bryły sztywnej w przestrzeni kartezjańskiej, a więc wektor przesunięcia i macierz rotacji oraz ich złożenia. Wspomniane zostały również alternatywne metody opisu orientacji tj. obrót wokół dowolnej osi oraz dwa wybrane zestawy kątów Eulera.

W kolejnych punktach wyprowadzono macierz przekształcenia jednorodnego umożliwiającą spójny opis pozycji układu współrzędnych oraz szeroko stosowane w robotyce operatory algebry przestrzennej. Kolejnym ważnym zagadnieniem opisanym w rozdziale są metody powiązania układów współrzędnych z poszczególnymi ogniwami manipulatora. W robotyce, do tego celu, stosowana jest głównie notacja Denavita-Hartenberga w jej prostej i zmodyfikowanej postaci. Rozdział zamyka elementarna klasyfikacja manipulatorów, oparta na ich geometrycznej konfiguracji.

Rozdział 2

W kolejnym rozdziale wyjaśniono ideę prostego i odwrotnego zadania kinematyki pozycyjnej, a następnie rozwiązano je w odniesieniu do manipulatora IRp-6 o sześciu stopniach swobody. Wyprowadzenie tychże równań poprzedzone zostało dokładnym opisem struktury manipulatora, oraz wyznaczeniem dla niego parametrów zmodyfikowanej notacji Denavita-Hartenberga.

Rozdział 3

Następna część pracy opisuje alternatywną metodę rozwiązywania problemu kinematyki. Opiera się ona na przekształceniu prędkości zmiennych przegubowych na ruch końcówki manipulatora. Do jej zastosowania konieczne jest wyznaczenie jacobianu geometrycznego⁵.

W pracy przedstawione zostały dwie równoważne metody wyprowadzenia macierzy jacobianu dla manipulatora IRp-6. Pierwsza z nich bazuje na postaci macierzy przekształcenia jednorodnego, druga na przekształceniu operatorów algebry przestrzennej. W pracy przedstawiono również ideę jacobianu analitycznego, metodę analityczną i numeryczną wyznaczenia odwrotności jacobia-

⁵W dalszej części pracy terminem jacobian określany będzie jacobian geometryczny manipulatora.

nu oraz krótkie omówienie problemu osobliwości manipulatora. Rozdział kończy opis algorytmów sterowania opartych na macierzy jacobianu, jego odwrotności i transpozycji.

Rozdział 4

W celu zbadania stabilności układu sterowania opartego na jacobianie przeprowadzone zostały adekwatne symulacje w środowisku Matlab. Testy te przeprowadzono w trzech seriach, a ich wyniki, w postaci wykresów, zamieszczone zostały w tym rozdziale.

Rozdział 5

W rozdziale tym zamieszczono opis wdrożonych w MRROC++ struktur sterowania opartych na kinematyce pozycyjnej oraz jacobianie.

Rozdział 6

W końcowej części pracy znajduje się podsumowanie przeprowadzonych badań, a więc porównanie obu przedstawionych metod.

Rozdział 1

Opis manipulatora

1.1 Elementarne definicje

Do opisu manipulatora konieczne jest wprowadzenie adekwatnej nomenklatury. Poniżej znajduje się krótki opis najczęściej stosowanych w robotyce pojęć.

- **Przestrzeń robocza** manipulatora jest całkowitym obszarem, do którego sięga końcówka robocza, przy pełnych zakresach wszystkich możliwych ruchów robota.
- **Przestrzeń kolizyjna** jest obszarem w obrębie którego zawierają się wszystkie elementy konstrukcyjne i przemieszczają się wszystkie zespoły ruchu manipulatora.
- **Strefa zagrożenia** jest to przestrzeń zabroniona przepisami lub normami BHP, do osiągnięcia w czasie pracy jednostki kinematycznej.
- **Współrzędne globalne** są układem odniesienia¹ przypisanym do stanowiska, w którym pracuje manipulator (podstawy) i względem którego wykonywane są wszystkie jego czynności.
- **Współrzędne lokalne** są układami współrzędnych przypisanych do każdego z członów manipulatora.
- **Efektor** jest częścią robota, umożliwiającą interakcję z otoczeniem. Mogą to być nogi maszyny kroczącej, koła, lub jak w przypadku manipulatora chwytak.
- **Receptor** jest to czujnik, enkoder lub inne urządzenie dzięki któremu robot zbiera informacje dotyczące środowiska, w którym operuje.

¹Układ ten nazywany jest również układem bazowym.

- **Stopień swobody** jest to liczba niezależnych ruchów, jakie ciało jest w stanie zrealizować w przestrzeni. Ciało swobodne w przestrzeni trójwymiarowej może posiadać maksymalnie sześć stopni swobody - trzy związane z ruchem postępowym i trzy obrotowym.
- **Ciało sztywne** jest idealizacją ciał fizycznych i oznacza ciało, którego elementy nie mogą się wzajemnie przemieszczać.
- **Przegub** to połączenie dwóch elementów umożliwiające ich wzajemny ruch obrotowy dookoła osi lub punktu.
- **Więzy** są ograniczeniami ruchu nałożonymi na poruszające się ciało. Rozróżniamy więzy holonomiczne (dające się opisać prostymi równaniami ruchu) i nieholonomiczne (nie dające się opisać pojedynczymi równaniami różniczkowymi), a także więzy reonomiczne (zależne jawnie od czasu) i przeciwne do nich skleronomiczne.
- **Para kinematyczna** to połączenie ruchowe dwóch członów mechanizmu, odbierające tym członom pewną część stopni swobody. W zależności od ilości więzów nałożonych na człony wyróżniamy pięć klas par kinematycznych. Para kinematyczna pierwszego stopnia pozostawia członom pięć stopni swobody (np. moneta tocząca się bez tarcia po stole), natomiast piątego stopnia, tylko jeden (np. przegub translacyjny, lub obrotowy).
- **Łańcuch kinematyczny** jest szeregiem połączonych ze sobą członów tworzących jedną lub wiele par kinematycznych, realizującym zdefiniowane przeniesienie ruchu. Większość manipulatorów posiada strukturę szeregową, tworząc tym samym otwarty łańcuch kinematyczny.
- **Ruchliwość łańcucha kinematycznego** określa ilość stopni swobody posiadanych przez zadany łańcuch kinematyczny, co odpowiada ilości typów ruchu jakie jest on w stanie przenieść. Ruchliwość wyznaczamy z równania (1.1), gdzie n stanowi liczbę ogniw manipulatora, a p_i liczbę par kinematycznych stopnia i .

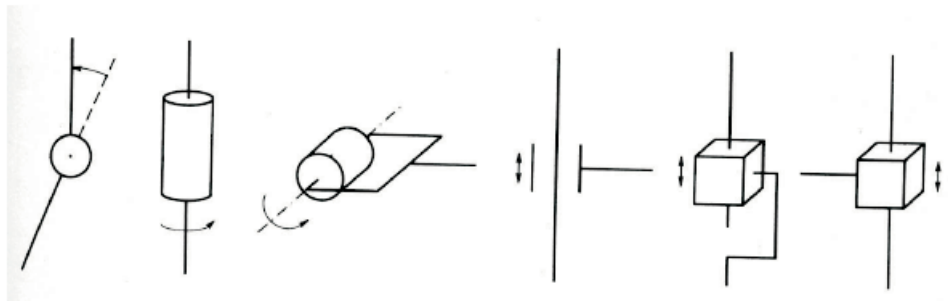
$$w = 6(n - 1) - 5p_5 - 4p_4 - 3p_3 - 2p_2 - p_1 \quad (1.1)$$

- **Powtarzalność** manipulatora określa jak blisko końcówka robocza może dojść do pozycji uprzednio osiągniętej.
- **Dokładność** manipulatora określa jak blisko końcówka robocza może dojść do zadanego punktu w przestrzeni roboczej.
- **Osobliwość** manipulatora wyznacza konfiguracje, z których pewne kierunki ruchu końcówki mogą być nieosiągalne.

1.2 Manipulator jako łańcuch kinematyczny

Manipulator, z punktu widzenia kinematyki, rozpatrujemy jako układ brył sztywnych, zwanych ogniwami, połączonych w łańcuch kinematyczny za pomocą przegubów. Stosowane w typowych manipulatorach przeguby posiadają zazwyczaj jeden stopień swobody i tworzą wraz z towarzyszącymi im ogniwami parę kinematyczną piątego stopnia. Najczęściej w konstrukcji robotów przemysłowych wykorzystywane są przeguby obrotowe (inaczej rotacyjne) mające charakter osi obrotu, oraz pryzmatyczne (inaczej przesuwne) o charakterze osi przesuwu. Istnieje oczywiście możliwość stosowania złącz o wyższej ruchliwości, takich jak przeguby sferyczne i kielichowe, jednak stosowanie ich wiąże się ze znaczną komplikacją sterowalności manipulatora. Do kinematycznego opisu złącz o k stopniach swobody używa się uproszczenia polegającego na ich dekompozycji na k złączy o jednym stopniu swobody i zerowej długości lub stosuje się alternatywną metodę ich opisu - zewnętrzną notację Denavita-Hartenberga [42], korzystającą z pięciu, a nie czterech jak notacja klasyczna, parametrów.

W dalszej części pracy przyjęto konwencję rosnącej, od nieruchomej podstawy o indeksie 0, numeracji ogniw. Trzymając się tej konwencji manipulator o N stopniach swobody zbudowany jest z $N + 1$ ogniw połączonych N złączami.



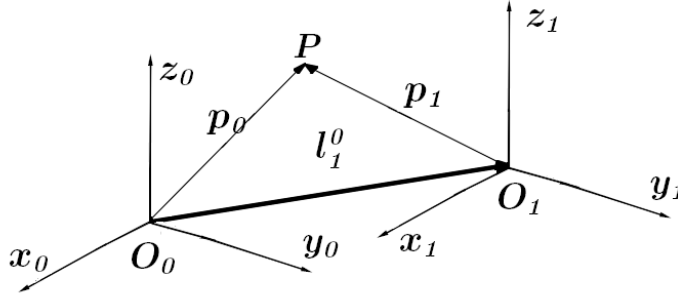
Rysunek 1.1: Graficzna reprezentacja przegubu obrotowego i translacyjnego [25]

1.3 Przekształcenia jednorodne

Poszczególne ogniwa manipulatora, oraz przedmioty podlegające manipulacji mogą być przybliżone bryłami sztywnymi. Dzięki temu założeniu usytuowanie każdego elementu robota jest w pełni opisane za pomocą jego przesunięcia i orientacji. Do matematycznego opisu tychże przekształceń stosowany jest wektor translacji, oraz macierz rotacji. Przy pomocy tak przyjętego aparatu matematycznego możliwy jest jednoznaczny opis dowolnej pozycji bryły sztywnej, oraz operacji złożenia tychże odwzorowań.

1.3.1 Przesunięcie równoległe

Rysunek (1.2) obrazuje przesunięcie równoległe układu współrzędnych w przestrzeni trójwymiarowej:



Rysunek 1.2: Przesunięcie równoległe układu współrzędnych

Wersory układów wejściowego i przesuniętego są wzajemnie równoległe, tak więc współrzędne dowolnego wektora p w odpowiednich układach wyznaczone są z zależności:

$$p_0 = p_1 + l_1^0 \quad (1.2)$$

Gdzie l_1^0 jest trójelementowym wektorem odległości układów wzdłuż osi Ox , Oy i Oz skierowanym od początku układu O_0 do O_1 .

$$l_1^0 = \begin{pmatrix} l_x \\ l_y \\ l_z \end{pmatrix} \quad (1.3)$$

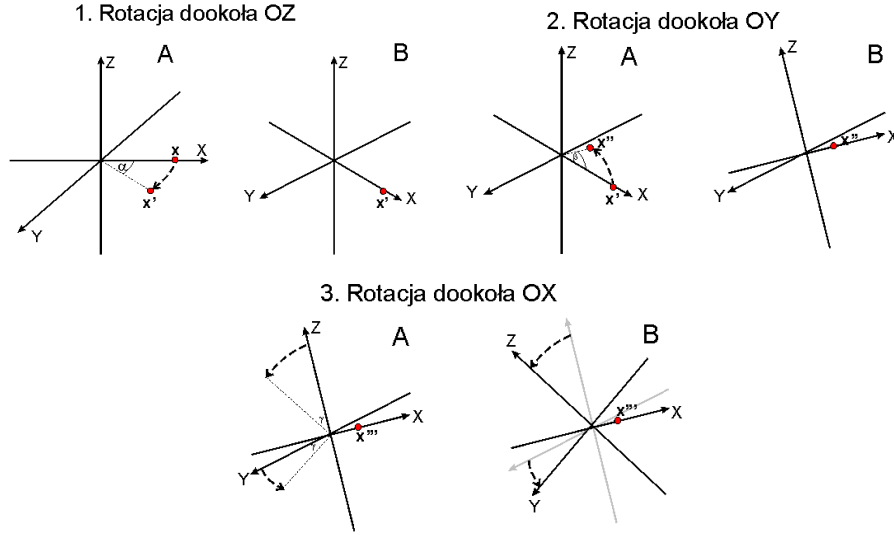
1.3.2 Macierz rotacji

Rysunek (1.3) przedstawia z kolei rotacje układów współrzędnych odpowiednio wokół osi Oz , Oy i Ox .

Współrzędne dowolnego wektora p w bazowym p_0 i obróconym p_1 układzie współrzędnych wyrażamy za pomocą równań (1.4). Po odpowiednich przekształceniach związek pomiędzy powyższymi wektorami wyrażamy równaniem postaci (1.6), gdzie macierz R_0^1 przyjmuje postać (1.7).

$$\begin{cases} p_0 = x_0 i_0 + y_0 j_0 + z_0 k_0 & \text{układ bazowy} \\ p_1 = x_1 i_1 + y_1 j_1 + z_1 k_1 & \text{układ obrócony} \end{cases} \quad (1.4)$$

$$\begin{cases} x_1 = p_1 i_1 = p_0 i_1 = x_0 i_0 i_1 + y_0 j_0 i_1 + z_0 k_0 i_1 \\ y_1 = p_1 j_1 = p_0 j_1 = x_0 i_0 j_1 + y_0 j_0 j_1 + z_0 k_0 j_1 \\ z_1 = p_1 k_1 = p_0 k_1 = x_0 i_0 k_1 + y_0 j_0 k_1 + z_0 k_0 k_1 \end{cases} \quad (1.5)$$



Rysunek 1.3: Obrót układu współrzędnych względem osi Oz , Oy i Ox

$$p_1 = R_0^1 p_0 \quad (1.6)$$

$$R_0^1 = \begin{pmatrix} i_0 i_1 & j_0 i_1 & k_0 i_1 \\ i_0 j_1 & j_0 j_1 & k_0 j_1 \\ i_0 k_1 & j_0 k_1 & k_0 k_1 \end{pmatrix} \quad (1.7)$$

Przekształcenie odwrotne do danego otrzymamy przez transpozycję macierzy R_0^1 według równania (1.9). Macierz, której odwrotnością jest macierz transponowana nazywana jest macierzą ortogonalną. Wyznacznik zdefiniowanej zależności (1.7) macierzy wynosi $\det R_i^j = \pm 1$, a ograniczając rozważania wyłącznie do układów prawoskrętnych $\det R_i^j = +1$. Reasumując macierze ortogonalne o wyznaczniku $+1$ są nazywane macierzami obrotu lub rotacji, a ich kolumny stanowią kosinusy kierunkowe osi układu obróconego względem układu wyjściowego.

$$p_0 = (R_0^1)^{-1} p_1 = R_1^0 p_1 \quad (1.8)$$

$$R_1^0 = (R_0^1)^{-1} = (R_0^1)^T = \begin{pmatrix} i_1 i_0 & j_1 i_0 & k_1 i_0 \\ i_1 j_0 & j_1 j_0 & k_1 j_0 \\ i_1 k_0 & j_1 k_0 & k_1 k_0 \end{pmatrix} \quad (1.9)$$

Elementarne macierze obrotu wokół osi Ox , Oy i Oz przyjmują postać:

$$R_{x,\theta} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{pmatrix} \quad (1.10)$$

$$R_{y,\theta} = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \quad (1.11)$$

$$R_{z,\theta} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (1.12)$$

Macierz rotacji stanowi według wzoru (1.6) odwzorowanie między dwoma układami współrzędnych. Jeśli dołączony zostanie trzeci układ powiązany z poprzednimi przez rotację, to dowolny wektor p reprezentowany może być następującymi zależnościami.

$$\begin{cases} p_0 &= R_1^0 p_1 \\ p_0 &= R_2^0 p_2 \\ p_1 &= R_2^1 p_2 \end{cases} \quad (1.13)$$

Po podstawieniu i porównaniu równań (1.13) otrzymamy bardzo ważną zależność (1.15) będącą prawem składania obrotów. Przy pomocy tego prawa i wzorów (1.10, 1.11, 1.12) możliwe jest opisanie dowolnej rotacji układu współrzędnych, względem dowolnej osi.

$$p_0 = R_2^0 p_2 = R_1^0 R_2^1 p_2 \quad (1.14)$$

$$R_2^0 = R_1^0 R_2^1 \quad (1.15)$$

Warto w tym miejscu zaznaczyć, iż operacja składania obrotów nie jest działaniem przemienne. Prawostronne wymnożenie macierzy odpowiada rotacji określonej przez drugą macierz względem lokalnego układu współrzędnych (tzn. opisanego przez pierwszą z macierzy). Lewostronne, z kolei, odpowiada obrotowi względem układu podstawowego.

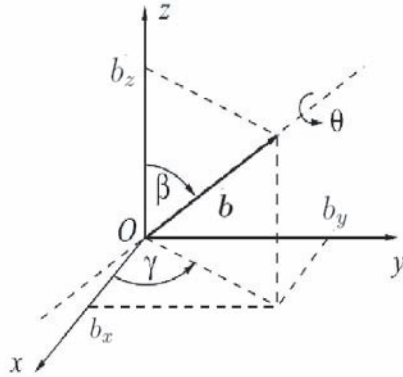
1.3.3 Reprezentacje rotacji

Macierz rotacji złożona jest z dziewięciu elementów powiązanych sześcioma ograniczeniami, wynikającymi z ortonormalności wektorów. Tymczasem minimalny opis orientacji bryły sztywnej w przestrzeni zawiera trzy elementy. Tak więc macierz rotacji jest operatorem redundantnym (nadmiarowym). Poniżej przedstawiono alternatywne metody opisu rotacji: reprezentacja oś - kąt i dwa zestawy kątów Eulera. W robotyce do opisu obrotów stosowana jest jeszcze jedna, pominięta w pracy, dosyć nowatorska metoda. Są nią kwaterniony będące trójwymiarowym uogólnieniem liczb zespolonych.

Reprezentacja oś - kąt

W opisie tym obrót reprezentowany jest przez cztery parametry: kąt obrotu θ , oraz trójelementowy wersor osi rotacji b . Jednym ze sposobów otrzymania wynikowej macierzy $R_{b,\theta}$ jest obrócenie wektora b na jedną z osi układu bazowego, w tym przypadku z , następnie obrót osi z o kąt θ i w końcu powrót wektora b do wejściowego położenia. Powyższe przekształcenia opisać można za pomocą poniższej sekwencji elementarnych rotacji:

- $R_{z,-\gamma}$ - obrót o kąt $-\gamma$ wokół osi z układu bazowego, gdzie γ jest kątem pomiędzy rzutem wektora b na płaszczyznę Oxy , a osią x układu bazowego.
- $R_{y,-\beta}$ - obrót o kąt $-\beta$ wokół osi y układu bazowego, gdzie β jest kątem między wektorem b , a osią z układu bazowego.
- $R_{z,\theta}$ - obrót o kąt θ wokół osi z układu bazowego, z którą pokrywa się, dzięki poprzednim rotacjom, wektor osi obrotu b .
- $R_{y,\beta}$ - obrót o kąt β wokół osi y układu bazowego
- $R_{z,\gamma}$ - obrót o kąt γ wokół osi z układu bazowego, dzięki czemu wektor b uzyskuje swoje pierwotne położenie.



Rysunek 1.4: Opis rotacji przez reprezentację oś - kąt[25]

Macierz rotacji odpowiadająca tym przekształceniom otrzymana jest przez prawostronne wymnożenie macierzy elementarnych (1.16).

$$\begin{aligned}
 R_{b,\theta} &= R_{z,\gamma} R_{y,\beta} R_{z,\theta} R_{y,-\beta} R_{z,-\gamma} = \\
 &= \begin{pmatrix} b_x^2(1 - \cos \theta) + \cos \theta & b_x b_y(1 - \cos \theta) - b_z \sin \theta & b_x b_z(1 - \cos \theta) + b_y \sin \theta \\ b_x b_y(1 - \cos \theta) + b_z \sin \theta & b_y^2(1 - \cos \theta) + \cos \theta & b_y b_z(1 - \cos \theta) - b_x \sin \theta \\ b_x b_z(1 - \cos \theta) - b_y \sin \theta & b_y b_z(1 - \cos \theta) + b_x \sin \theta & b_z^2(1 - \cos \theta) + \cos \theta \end{pmatrix}
 \end{aligned} \tag{1.16}$$

Mając zadaną macierz $R_{b,\theta}$,

$$R_{b,\theta} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \quad (1.17)$$

odwzorowanie odwrotne otrzymuje się stosując zależności (1.18, 1.19).

$$\theta = \arccos \left(\frac{r_{11} + r_{22} + r_{33} - 1}{2} \right) \quad (1.18)$$

$$b = \frac{1}{2 \sin \theta} \begin{pmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{pmatrix} \quad \sin \theta \neq 0 \quad (1.19)$$

Reprezentacja ta jest niejednoznaczna ponieważ spełniona jest zależność (1.20).

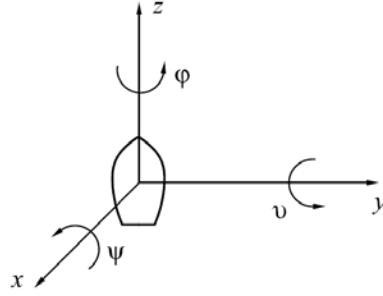
$$R_{b,\theta} = R_{-b,-\theta} \quad (1.20)$$

Kąty Eulera (RPY)

Przedstawiona powyżej reprezentacja ós-kąąt jest nadal operatorem nadmiarowym (do jej opisu konieczne są cztery elementy). Minimalna reprezentacja, z kolei złożona jest z trzech niezależnych parametrów, a tworzą ją tzw. kąty Eulera. Stanowią one złożenie trzech elementarnych obrotów, tworząc łącznie 12 możliwych ich kombinacji.

W rozdziale tym zostały zaprezentowane dwa najczęściej stosowane zestawy kątów Eulera. Po pierwsze jest to rozpowszechniony w nawigacji morskiej i powietrznej zestaw kątów RPY (ang. Roll - Pitch - Yaw) - tzn. kołysanie boczne, kołysanie wzdłużne, zbaczanie. Zestaw tych kątów $(\varphi, \vartheta, \psi)$ stanowi minimalną reprezentację orientacji i odpowiada złożeniu elementarnych rotacji względem bazowego układu współrzędnych:

- $\mathbf{R}_{x,\psi}$ - obrót o kąt ψ wokół osi x układu odniesienia (zbaczanie)
- $\mathbf{R}_{y,\vartheta}$ - obrót o kąt ϑ wokół osi y układu odniesienia (kołysanie wzdłużne)
- $\mathbf{R}_{z,\varphi}$ - obrót o kąt φ wokół osi z układu odniesienia (kołysanie boczne)



Rysunek 1.5: Opis rotacji za pomocą kątów Eulera RPY [25]

Macierz rotacji odpowiadająca tym przekształceniom otrzymana jest poprzez lewostronne wymnożenie macierzy (1.21), natomiast odwzorowanie odwrotne przyjmuje postać ² (1.22), gdy $\vartheta \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$ oraz (1.23) dla $\vartheta \in \left(\frac{\pi}{2}, \frac{3\pi}{2}\right)$.

$$R_{rpy} = R_{z,\varphi} R_{y,\vartheta} R_{x,\psi} = \quad (1.21)$$

$$= \begin{pmatrix} \cos \varphi \cos \vartheta & \cos \varphi \sin \vartheta \sin \psi - \sin \varphi \cos \psi & \cos \varphi \sin \vartheta \cos \psi + \sin \varphi \sin \psi \\ \sin \varphi \cos \vartheta & \sin \varphi \sin \vartheta \sin \psi + \cos \varphi \cos \psi & \sin \varphi \sin \vartheta \cos \psi - \cos \varphi \sin \psi \\ -\sin \vartheta & \cos \vartheta \sin \psi & \cos \vartheta \cos \psi \end{pmatrix}$$

$$\begin{cases} \varphi = \arctan\left(\frac{r_{21}}{r_{11}}\right) \\ \vartheta = \arctan\left(\frac{-r_{31}}{\sqrt{r_{32}^2 + r_{33}^2}}\right) \\ \psi = \arctan\left(\frac{r_{32}}{r_{33}}\right) \end{cases} \quad (1.22)$$

$$\begin{cases} \varphi = \arctan\left(\frac{-r_{21}}{-r_{11}}\right) \\ \vartheta = \arctan\left(\frac{-r_{31}}{-\sqrt{r_{32}^2 + r_{33}^2}}\right) \\ \psi = \arctan\left(\frac{-r_{32}}{-r_{33}}\right) \end{cases} \quad (1.23)$$

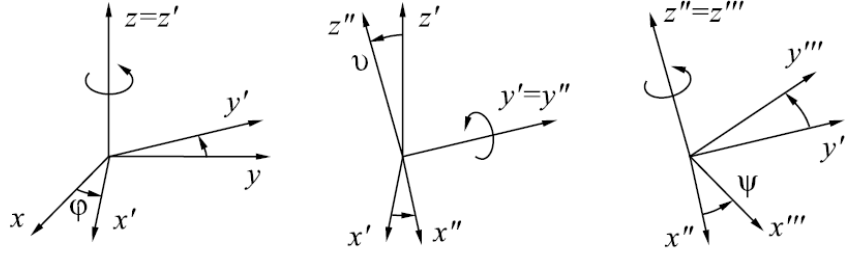
Jak widać dla $\vartheta = \pm\frac{\pi}{2}$ powyższe równania są niejednoznaczne i istnieje nieskończenie wielka liczba rozwiązań dla φ i ψ .

²Do zaimplementowania funkcji arcus tangens wykorzystywana jest dostępna w języku C++ oraz pakiecie Matlab funkcja $Atan2(x,y)$ wyliczająca wartość funkcji arcus tangens ilorazu $\frac{x}{y}$ oraz wyznaczająca na podstawie znaków argumentów położenie wynikowego kąta w odpowiedniej ćwiartce układu współrzędnych.

Kąty Eulera (ZYZ)

Ten zestaw kątów Eulera zwany potocznie kątami precesji, nutacji i obrotu własnego opisany jest przez następujący szereg rotacji względem bieżącego układu współrzędnych:

- $R_{z,\varphi}$ - obrót o kąt φ wokół osi z układu odniesienia
- $R_{y',\vartheta}$ - obrót o kąt ϑ wokół osi y' bieżącego układu
- $R_{z'',\psi}$ - obrót o kąt ψ wokół osi z'' bieżącego układu



Rysunek 1.6: Opis rotacji za pomocą kątów Eulera ZYZ [25]

Macierz rotacji odpowiadająca tym przekształceniom otrzymana jest przez prawostronne wymnożenie macierzy (1.24), natomiast odwzorowanie odwrotne przyjmuje postać (1.25), gdy $\vartheta \in (0, \pi)$ oraz (1.26) dla $\vartheta \in (-\pi, 0)$.

$$R_{zyz} = R_{z,\varphi} R_{y',\vartheta} R_{z'',\psi} = \quad (1.24)$$

$$= \begin{pmatrix} \cos \varphi \cos \vartheta \cos \psi - \sin \varphi \sin \psi & -\cos \varphi \cos \vartheta \sin \psi - \sin \varphi \cos \psi & \cos \varphi \sin \vartheta \\ \sin \varphi \cos \vartheta \cos \psi + \cos \varphi \sin \psi & -\sin \varphi \cos \vartheta \sin \psi + \cos \varphi \cos \psi & \sin \varphi \sin \vartheta \\ -\sin \vartheta \cos \psi & \sin \vartheta \sin \psi & \cos \vartheta \end{pmatrix}$$

$$\begin{cases} \varphi = \arctan\left(\frac{r_{23}}{r_{13}}\right) \\ \vartheta = \arctan\left(\frac{\sqrt{r_{13}^2 + r_{23}^2}}{r_{33}}\right) \\ \psi = \arctan\left(\frac{r_{32}}{-r_{31}}\right) \end{cases} \quad (1.25)$$

$$\begin{cases} \varphi = \arctan\left(\frac{-r_{23}}{-r_{13}}\right) \\ \vartheta = \arctan\left(\frac{-\sqrt{r_{13}^2 + r_{23}^2}}{r_{33}}\right) \\ \psi = \arctan\left(\frac{-r_{32}}{r_{31}}\right) \end{cases} \quad (1.26)$$

W przypadku, gdy $\sin \vartheta = 0$ równania te są niejednoznaczne i istnieje dla nich nieskończona liczba rozwiązań, gdyż dla $\vartheta = 0$ lub $\vartheta = \pi$ obroty o kąty φ i ψ następują wokół równoległych osi.

1.4 Macierz przekształcenia jednorodnego

Za pomocą macierzy przekształcenia jednorodnego możliwa jest prezentacja obrotu i przesunięcia przy pomocy pojedynczego ruchu. Macierz ta składa się z czterech podmacierzy: omówionego wcześniej wektora przesunięcia l i macierzy rotacji R , oraz perspektywy i skalowania. W kinematyce robotów ostatni wiersz nie ma znaczenia, a jako wartość skalowania zawsze wstawiana jest jedność, a perspektywy wektor zerowy.

$$T = \begin{pmatrix} \text{rotacja} & \text{przesuniecie} \\ \text{perspektywa} & \text{skalowanie} \end{pmatrix} = \begin{pmatrix} R & l \\ 0 & 1 \end{pmatrix} \quad (1.27)$$

Kartezjański układ współrzędnych posiada sześć stopni swobody, a każdemu z nich odpowiada jedna z opisanych poniżej macierzy przekształcenia jednorodnego: kolejno przesunięcia wzdłuż osi Ox Oy i Oz o a , oraz obroty wokół tychże osi o kąt θ . Za pomocą zbioru przedstawionych macierzy możliwe jest określenie położenia i orientacji dowolnego układu współrzędnych w trójwymiarowej przestrzeni.

$$Trans_{x,a} = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.28)$$

$$Trans_{y,a} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & a \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.29)$$

$$Trans_{z,a} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.30)$$

$$Rot_{x,\theta} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.31)$$

$$Rot_{y,\theta} = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.32)$$

$$Rot_{z,\theta} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.33)$$

Jeżeli dany układ współrzędnych wykonuje serię ruchów, to współrzędne dowolnego wektora p w poszczególnych układach mogą zostać opisane za pomocą równań (1.34). Następnie porównując odpowiednie współczynniki dostaniemy równania (1.35), będące prawem składania ruchów i dające zapisać się w zwartej postaci (1.37). Odwrotność macierzy przekształcenia jednorodnego, po uwzględnieniu własności (1.9), wyliczana jest z zależności (1.38).

$$\begin{aligned} p_1 &= R_2^1 p_2 + l_2^1 \\ p_0 &= R_2^0 p_2 + l_2^0 \\ p_0 &= R_1^0 p_1 + l_1^0 = R_1^0 R_2^1 p_2 + R_1^0 l_2^1 + l_1^0 \end{aligned} \quad (1.34)$$

$$\begin{aligned} R_2^0 &= R_1^0 R_2^1 \\ l_2^0 &= R_1^0 l_2^1 + l_1^0 \end{aligned} \quad (1.35)$$

$$\begin{pmatrix} R_1^0 & l_1^0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R_2^1 & l_2^1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} R_1^0 R_2^1 & R_1^0 l_2^1 + l_1^0 \\ 0 & 1 \end{pmatrix} \quad (1.36)$$

$$T_b^a = T_c^a T_b^c \quad (1.37)$$

$$(T_b^a)^{-1} = T_a^b = \begin{pmatrix} (R_b^a)^T & -(R_b^a)^T l_b^a \\ 0 & 1 \end{pmatrix} \quad (1.38)$$

1.5 Macierz skośniesymetryczna

Iloczyn wektorowy $x \times y$ dwóch wektorów $x, y \in \mathbb{R}^3$ wyliczany jest z zależności (1.39). W przypadku, gdy oba wektory należą do przestrzeni \mathbb{R}^3 długość wektora stanowiącego wynik iloczynu wektorowego wyznaczana jest z zależności (1.40), gdzie α jest kątem między jego argumentami. Kierunek i zwrot wyniku iloczynu wektorowego jest wyznaczony według reguły śruby prawoskrętnej i jest prostopadły do obu wektorów.

$$x \times y = \det \begin{pmatrix} i & j & k \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{pmatrix} = \begin{pmatrix} x_2 y_3 - x_3 y_2 \\ x_3 y_1 - x_1 y_3 \\ x_1 y_2 - x_2 y_1 \end{pmatrix} \in \mathbb{R}^3 \quad (1.39)$$

$$x \times y = \|x\| \|y\| \sin \alpha \quad (1.40)$$

Podstawowe własności iloczynu wektorowego:

$$x \times x = 0 \quad (1.41)$$

$$x \times y = -y \times x \quad (1.42)$$

$$x \times (y + z) = x \times y + y \times z \quad (1.43)$$

$$\alpha (x \times y) = (\alpha x) \times y = x \times (\alpha y) \quad (1.44)$$

Kolejną własnością iloczynu wektorowego jest zależność (1.45). Oznacza ona że, o ile R jest macierzą ortogonalną, to operacja wyznaczenia iloczynu wektorowego i obrotu są przemienne.

$$R(a \times b) = Ra \times Rb \quad (1.45)$$

Alternatywną metodą reprezentacji iloczynu wektorowego jest macierz skośniesymetryczna (1.47). Macierz $S(\cdot)$ jest operatorem liniowym tzn. (1.48).

$$S(x) = \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix} \quad (1.46)$$

$$x \times y = S(x)y = -S(y)x \quad (1.47)$$

$$S(\alpha a + \beta b) = \alpha S(a) + \beta S(b) \quad (1.48)$$

Korzystając z powyższych własności można wyprowadzić bardzo ważną zależność dotyczącą składania obrotów (1.49). Zależność ta stwierdza, iż reprezentacja macierzy $S(a)$ w układzie współrzędnych obróconym przez R jest taka sama jak macierz symetryczna $S(Ra)$ odpowiadająca wektorowi a obróconemu przez R .

$$RS(a)R^T b = R(a \times R^T b) = (Ra) \times (RR^T b) = (Ra) \times b = S(Ra)b$$

$$RS(a)R^T = S(Ra) \quad (1.49)$$

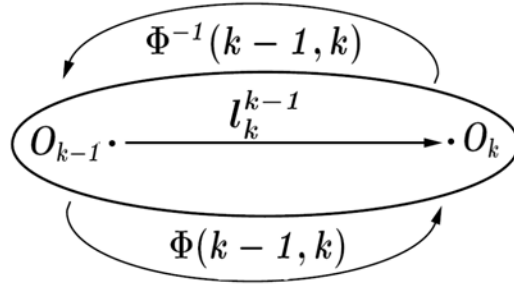
Macierz skośniesymetryczna odgrywa kluczowe znaczenie przy wyznaczeniu przestrzennej macierzy odległości oraz wyliczeniu różniczki macierzy rotacji. Problem ten zostanie szczegółowo omówiony w kolejnych rozdziałach pracy.

1.6 Operatory algebry przestrzennej

Alternatywnym, do macierzy przekształcenia jednorodnego, opisem ogniw manipulatora są, wykorzystane po raz pierwszy przez Rodrigueza [4, 17], operatory przestrzenne. Przy ich pomocy możliwe jest zwarte opisanie przekształceń pozycji, prędkości i sił między członami robota, a co za tym idzie wyznaczenie jakobianu. Poniżej opisane zostały stosowane w dalszej części pracy operatory.

- $\Phi(k-1, k)$ będąca przestrzennym operatorem odległości przekształcającym układ O_{k-1} na O_k postaci (1.50). W zapisie tym macierze O i U są odpowiednio macierzą zerową i jednostkową określoną w przestrzeni $\mathbb{R}^{3 \times 3}$, a $S(l_k^{k-1})$ macierzą skośnosymetryczną wektora odległości l_k^{k-1} .

$$\Phi(k-1, k) = \begin{pmatrix} U & S(l_k^{k-1}) \\ O & U \end{pmatrix} \quad (1.50)$$



Rysunek 1.7: Przestrzenny operator odległości

Ruch każdego punktu bryły sztywnej opisany jest przez sześcioelementowy wektor postaci (1.51), zawierający prędkości kątowe i liniowe, wzdłuż każdej z osi układu odniesienia. Chcąc wyznaczyć wartości elementów tego wektora względem układu O_k , przy zadanych jego wartościach względem układu O_{k-1} , oraz zadanej odległości tych układów l_k^{k-1} , stosujemy zależności postaci (1.52), dające zapisać się, przy użyciu operatora odległości, w zwartej postaci (1.53).

$$V_k = \begin{pmatrix} \omega_k \\ v_k \end{pmatrix} \quad (1.51)$$

$$\begin{cases} \omega_k = \omega_{k-1} \\ v_k = v_{k-1} + \omega_{k-1} \times l_k^{k-1} \end{cases} \quad (1.52)$$

$$V_k = (\Phi(k-1, k))^T V_{k-1} \quad (1.53)$$

Analogiczne przekształcenia wyznaczamy dla wektora sił i momentów sił przestrzennych (1.56).

$$f_k = \begin{pmatrix} N_k \\ F_k \end{pmatrix} \quad (1.54)$$

$$\begin{cases} N_k = N_{k-1} + l_k^{k-1} \times F_{k-1} \\ F_k = F_{k-1} \end{cases} \quad (1.55)$$

$$f_k = \Phi(k-1, k)f_{k-1} \quad (1.56)$$

Przestrzenną macierz odległości cechują następujące własności :

$$\Phi(j, k)^{-1} = \Phi(k, j) \quad (1.57)$$

$$\Phi(j, j) = U_{6 \times 6} \quad (1.58)$$

$$\forall_{j < k} \Phi(j, k) = \Phi(j, j+1)\Phi(j+1, j+2) \dots \Phi(k-1, k) \quad (1.59)$$

- $A(k-1, k)$ jest przestrzenną macierzą kosinusów kierunkowych, wyznaczoną bezpośrednio z macierzy rotacji. Macierz ta jest dopełnieniem operatora odległości na przypadek, gdy pomiędzy układami współrzędnych występuje rotacja.

$$A(k-1, k) = \begin{pmatrix} R_k^{k-1} & O \\ O & R_k^{k-1} \end{pmatrix} \quad (1.60)$$

Poniżej przedstawiono cechy tego operatora:

$$A(k-1, k)^{-1} = \begin{pmatrix} R_{k-1}^k & O \\ O & R_{k-1}^k \end{pmatrix} \quad (1.61)$$

$$\forall_{j < k} A(j, k) = A(j, j+1)A(j+1, j+2) \dots A(k-1, k) \quad (1.62)$$

- $L(k-1, k)$ zwana przestrzenną macierzą pozycji jest złożeniem obu poprzednich operatorów, dzięki czemu opisuje ona zarówno położenie, jak i orientację układu.

$$L(k-1, k) = \Phi(k-1, k)A(k-1, k) \quad (1.63)$$

$$L(k-1, k) = \begin{pmatrix} R_k^{k-1} & S(l_k^{k-1})R_k^{k-1} \\ O & R_k^{k-1} \end{pmatrix} \quad (1.64)$$

- \mathbf{H}_k wektor rzutowania złącza jest operatorem odwzorowującym wektor prędkości zmiennych wewnętrznych złącza, na względną przestrzenną prędkość między tym, a kolejnym ogniwem.

$$\mathbf{V}_k = \mathbf{H}_k \dot{\mathbf{q}} \quad (1.65)$$

Poniżej przedstawiono wartości operatora \mathbf{H}_k dla różnego rodzaju, powszechnie stosowanych złącz:

$$\begin{array}{ccc} \mathbf{H}_k = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}^T & \mathbf{H}_k = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}^T & \mathbf{H}_k = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ p \end{pmatrix}^T \\ \text{przegub obrotowy} & \text{przegub przesuwny} & \text{przegub śrubowy} \end{array}$$

$$\begin{array}{ccc} \mathbf{H}_k = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}^T & \mathbf{H}_k = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}^T & (1.66) \\ \text{przegub cylindryczny} & \text{przegub sferyczny} & \end{array}$$

1.7 Notacja Denavita-Hartenberga

Najczęściej stosowanym w robotyce opisem ogniwa manipulatora są parametry Denavita-Hartenberga³. Przyporządkowują one każdemu ogniwu cztery wielkości:

- \mathbf{a}_i - długość ogniwa i , mierzona jako odległość między osiami przegubów i oraz $(i + 1)$.
- α_i - kąt skręcenia ogniwa i prawoskrętnie wokół a_i , mierzony jako kąt między osiami przegubów i oraz $(i + 1)$.
- \mathbf{d}_i - odległość między a_{i-1} i a_i mierzona wzdłuż osi przegubu i .

³Przykładami innej, stosowanej w robotyce, notacji jest S-model [40] korzystający z sześciu parametrów lub CPC (completeness and parametric continuity) [47].

- θ_i - kąt między a_{i-1} i a_i określony prawoskrętnie wokół osi przegubu i .

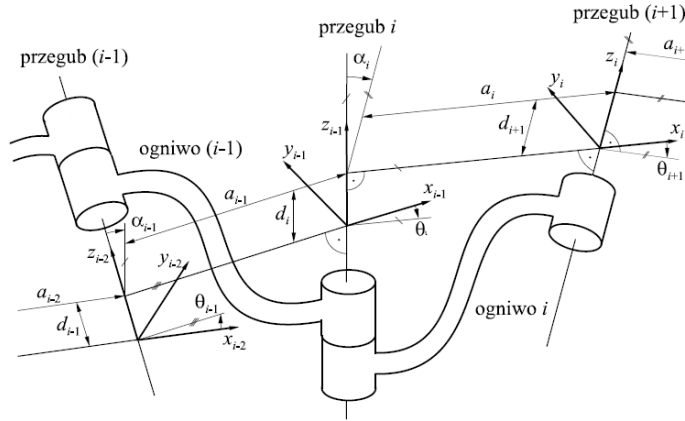
Pierwsze dwie wielkości opisują ogniwo, pozostałe dwie przegub. Zmienną przegubową związaną z ogniwem i oznaczamy przez q_i i wyznaczamy z zależności (1.67), przy czym $\sigma_i = 0$ dla przegubu obrotowego ($q_i = \theta_i$; $a_i, \alpha_i, d_i = \text{const}$), lub $\sigma_i = 1$ gdy przegub jest przesuwny ($q_i = d_i$; $a_i, \alpha_i, \theta_i = \text{const}$)

$$q_i = (1 - \sigma_i) \theta_i + \sigma_i d_i \quad (1.67)$$

W literaturze stosowane są równoległe dwie notacje przyporządkowujące poszczególnym ogniwom układy współrzędnych: Notacja Denavita-Hartenberga DH [13, 40], oraz Zmodyfikowana notacja Denavita-Hartenberga ZDH [39, 21], zwana również notacją Khalila-Kleinfingera [22].

1. Przyporządkowanie osi układów według notacji Denavita-Hartenberga (DH)

- Oś z_i pokrywa się z osią przegubu $(i + 1)$.
- Oś x_i jest prostopadła do osi z_{i-1} oraz z_i i jest skierowana od przegubu $(i - 1)$ do i
- Oś y_i uzupełnia prostokątny układ współrzędnych



Rysunek 1.8: Opis parametrów według notacji DH[25]

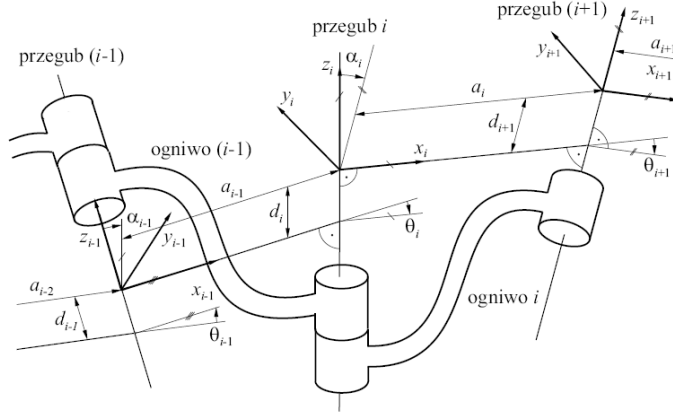
Macierz przekształcenia jednorodnego dla tego układu uzyskamy przez złożenie następujących przekształceń.

$$T = Rot_{z, \theta_i} Trans_{z, d_i} Trans_{x, a_i} Rot_{x, \alpha_i} = \quad (1.68)$$

$$= \begin{pmatrix} \cos(\theta_i) & -\sin(\theta_i) \cos(\alpha_i) & \sin(\theta_i) \sin(\alpha_i) & a_i \cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \cos(\alpha_i) & -\cos(\theta_i) \sin(\alpha_i) & a_i \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.69)$$

2. Przyporządkowanie osi układów według zmodyfikowanej notacji Denavita-Hartenberga (ZDH)

- Oś z_i pokrywa się z osią przegubu i
- Oś x_i jest prostopadła do osi z_i oraz z_{i+1} i jest skierowana od przegubu i do $(i+1)$
- Oś y_i uzupełnia prostokątny układ współrzędnych



Rysunek 1.9: Opis parametrów według notacji ZDH[25]

W tym przypadku macierz przekształcenia jednorodnego przyjmie postać:

$$T = Trans_{x, a_{i-1}} Rot_{x, \alpha_{i-1}} Rot_{z, \theta_i} Trans_{z, d_i} = \quad (1.70)$$

$$= \begin{pmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_{i-1} \\ \sin(\theta_i) \cos(\alpha_{i-1}) & \cos(\theta_i) \cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & -d_i \sin(\alpha_{i-1}) \\ \sin(\theta_i) \sin(\alpha_{i-1}) & \cos(\theta_i) \sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & d_i \cos(\alpha_{i-1}) \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.71)$$

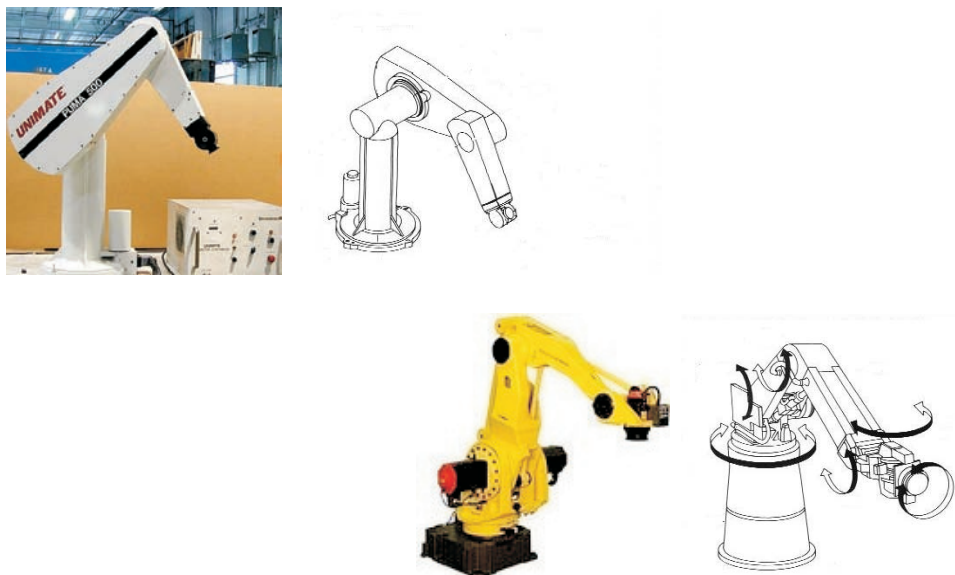
Zasadnicza różnica między przedstawionymi wyżej notacjami polega na tym, iż w notacji ZDH oś z_i , jest położona wzdłuż osi przegubu i , a w notacji DH wzdłuż przegubu $(i+1)$. Konsekwencją tego jest to, że do opisu ogniwa i notacja DH korzysta z parametrów θ_i , d_i , α_i , a_i (związanych z przegubem i), natomiast notacja ZDH parametrów θ_i , d_i , α_{i-1} , a_{i-1} (związanych z przegubem $(i-1)$). Notacja DH była wprowadzona jako pierwsza. Obecnie częściej stosowana jest notacja zmodyfikowana, będąca bardziej intuicyjną. W ZDH układ współrzędnych związany z danym ogniwem usytuowany jest na jego początku tzn. bliżej podstawy manipulatora, co jest bardziej naturalne, bo ruch bryły sztywnej opisywany jest zazwyczaj względem układu bazowego. W dalszej części pracy, podczas wyznaczania parametrów dla manipulatora IRp-6, wykorzystana została notacja zmodyfikowana.

1.8 Typy manipulatorów

Manipulatory są klasyfikowane na podstawie ich przeznaczenia, sposobu sterowania, czy rodzaju zastosowanego napędu, najbardziej jednak klasyczny jest podział uwzględniający ich geometrię. Stosowane obecnie roboty przemysłowe posiadają najczęściej sześć stopni swobody, z czego trzy związane są z ramieniem i podstawą, a trzy kolejne z kciścią. Klasyfikacja geometryczna bazuje na rodzaju i ułożeniu trzech pierwszych przegubów, determinujących kształt przestrzeni roboczej manipulatora. Rozróżniamy pięć podstawowych geometrii manipulatorów, różniących się kolejnością i rodzajem zastosowanych przegubów. Powyższa klasyfikacja odnosi się jedynie do manipulatorów o strukturze szeregowej. Oddrębną grupę stanowią roboty o strukturze równoległej, omówionej w końcowej części podrozdziału.

Konfiguracja Stawowa (RRR)

Manipulatory o tej konfiguracji zwane są inaczej manipulatorami obrotowymi, lub antropomorficznymi i składają się z trzech przegubów obrotowych. Są one najszerszym stosowanym typem manipulatorów, zapewniającym stosunkowo dużą swobodę ruchu, w dość zwartej przestrzeni. Roboty o takiej konfiguracji mogą być wykorzystywane przy operacjach montażu, obróbki, polerowania i lakierowania. W grupie tej wyróżnić można dwa rodzaje struktur - manipulatory z łokciem których przykładem jest manipulator PUMA, lub manipulatory o strukturze z równoległobokiem np. Nachi SC06F.



Rysunek 1.10: Konfiguracja Stawowa: z łokciem i równoległoboczna

Konfiguracja Sferyczna (RRP)

Składa się ona z dwóch przegubów obrotowych i przegubu przesuwnego. Osie tych przegubów są wzajemnie prostopadłe. Wartości zmiennych przegubowych manipulatora stanowią równocześnie współrzędne położenia końcówki roboczej, w sferycznym układzie współrzędnych, o początku w punkcie przecięcia osi drugiego i trzeciego przegubu. Konfiguracja ta kosztem dużego zasięgu poziomego, ma mały zasięg pionowy i jest niewygodna w omijaniu przeszkód. Ten typ robota stosowany jest najczęściej w różnego typu operacjach spawania. Przykładem konfiguracji sferycznej jest manipulator Fanuc L-1000.



Rysunek 1.11: Konfiguracja Sferyczna[1]

Konfiguracja Scara (RRP)

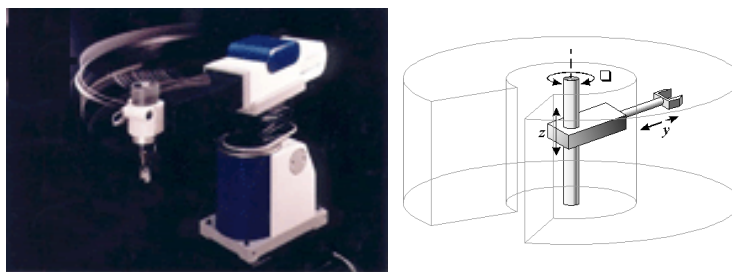
Manipulatory tego typu również posiadają dwa przeguby obrotowe i jeden przesuwny, jednak w odróżnieniu od poprzedniej klasy ich osie ułożone są równoległe. Manipulatory tego typu przeznaczone są głównie do montażu elementów i podzespołów, oraz przenoszenia i sortowania detali. Znacznym defektem konfiguracji jest skomplikowana budowa ramienia i liczne położenia osobliwe, przez co algorytm sterowania staje się bardziej złożony. Przykładami tego typu manipulatorów są Adept One firmy Adept i Yamaha YK1000X.



Rysunek 1.12: Manipulator typu SCARA[1]

Konfiguracja Cylindryczna (RPP)

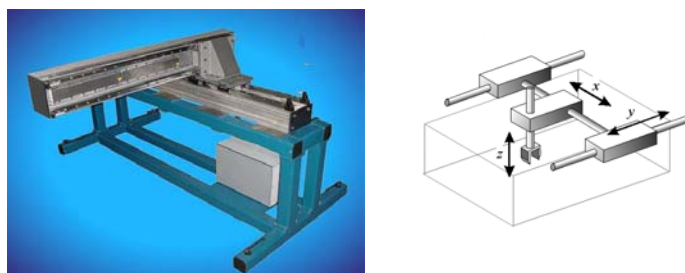
Pierwszy przegub jest obrotowy i wykonuje obrót wokół podstawy, podczas gdy pozostałe dwa są przesuwne. Zmienne przegubowe manipulatora stanowią współrzędne cylindryczne końcówki roboczej manipulatora względem podstawy, a przestrzenią roboczą jest niepełny cylinder. Konfiguracja upraszcza programowanie ruchu obrotowego, jednak uniemożliwia ruch końcówki powyżej manipulatora i znacznie utrudnia omijanie przeszkód. Roboty o tej konfiguracji stosowane są na przykład przy operacjach montażu i zgrzewaniu punktowym. Przykładem tej konfiguracji jest Seiko RT3300 i GMF M-100.



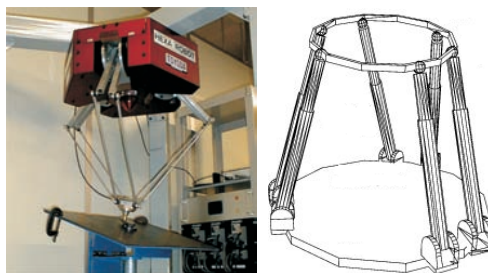
Rysunek 1.13: Konfiguracja Cylindryczna[1]

Konfiguracja Kartezjańska (PPP)

Trzy pierwsze przeguby manipulatora o tej konfiguracji są translacyjne. Dla tego rodzaju robotów zmienne przegubowe stanowią równocześnie wartości współrzędnych kartezjańskich końcówki roboczej, względem podstawy. Z punktu widzenia kinematyki, wizualizacji pracy, czy sterowania jest to najprostsza konfiguracja, jednak ich możliwość manipulacji jest znacznie uboższa i wymaga dużej przestrzeni. Tego rodzaju struktura stosowana jest głównie do montażu na blacie stołu, oraz transportu ładunku. Przykładami tego typu manipulatorów są Bosch Rexroth CKK i Epson EZ modules.



Rysunek 1.14: Konfiguracja Kartezjańska[1]



Rysunek 1.15: Manipulator Równoległy

Konfiguracja równoległa

Przykładem robotów posiadających zamknięty łańcuch kinematyczny są manipulatory równoległe. Roboty te posiadają na ogół troje ramion, wprowadzających trzy stopnie swobody. Poprzez sterowanie tych ramion, zmieniana jest pozycja ruchomej platformy, wyposażonej w efektor o dodatkowym stopniu swobody. Tego typu roboty znalazły zastosowanie między innymi w przemyśle spożywczym, farmaceutycznym oraz elektronicznym. Przykładem robota równoległego jest robot IRB 340 firmy ABB.

Rozdział 2

Kinematyka pozycyjna

2.1 Idea zadania kinematyki pozycyjnej

2.1.1 Proste zadanie kinematyki

Istotą zadania kinematyki prostej, zwanej inaczej kinematyką konfiguracji, jest określenie pozycji końcówki roboczej lub narzędzia, w bazowym układzie współrzędnych, przy zadanych wartościach zmiennych przegubowych. Biorąc pod uwagę, iż każda para kinematyczna ma jeden stopień swobody, działanie każdego przegubu można opisać jedną liczbą rzeczywistą: kątem obrotu w przypadku członu obrotowego lub przemieszczeniem w przypadku członu translacyjnego.

Do opisu kinematyki robotów konieczne jest wprowadzenie jednoznacznej konwencji obliczeń, takiej jak omówiona wcześniej notacja Denavita-Hartenberga w jej prostej i zmodyfikowanej postaci. Umożliwia ona wyznaczenie, według wzoru (1.71), wartości macierzy przekształcenia jednorodnego pomiędzy poszczególnymi ogniwami manipulatora, a następnie, korzystając z własności (1.37), powiązanie pozycji końcówki manipulatora z układem bazowym związanym z kolumną manipulatora. Wyznaczona w ten sposób macierz przekształcenia jednorodnego jest rozwiązaniem prostego zadania kinematyki. Należy jeszcze podkreślić, iż rozwiązanie to, w przypadku otwartych łańcuchów kinematycznych, jest zawsze jednoznaczne.

Zaprezentowana w niniejszej pracy metoda nie jest jedyną możliwością podejścia do rozwiązywania problemu kinematyki prostej w oparciu o geometryczną konfigurację manipulatora. Do alternatywnych rozwiązań należy np. omówiona w pracy [38] metoda, w której rozwiązywanie tego zagadnienia odbywa się bez konieczności wyznaczenia parametrów Denavita-Hartenberga, a macierze przekształcenia jednorodnego poszczególnych członów manipulatora wyznaczane są bezpośrednio względem bazowego układu współrzędnych. Rzecz jasna w metodzie tej końcowa postać rozwiązania otrzymana zostanie poprzez lewostronne wymnożenie pośrednich wyników.

2.1.2 Odwrotne zadanie kinematyki

Zadanie kinematyki odwrotnej polega na znalezieniu odwrotnego do powyższego odwzorowania i dla manipulatorów o strukturze szeregowej jest znacznie od niego trudniejsze. Jest to zadanie fundamentalne, ponieważ dla znanego położenia narzędzia w przestrzeni kartezjańskiej należy wyznaczyć wartości współrzędnych uogólnionych tak, aby możliwa była realizacja ruchu końcówki roboczej wzdłuż zadanej trajektorii. Ideę zadania kinematyki odwrotnej można przedstawić następująco. Mając zadane wartości elementów macierzy przekształcenia jednorodnego między końcówką manipulatora, a układem bazowym, należy odnaleźć jedno lub wszystkie rozwiązania równania (2.1), dającego łącznie 16 równań z n niewiadomymi postaci (2.2). Równania odpowiadające ostatniemu wierszowi macierzy H są trywialne, dzięki czemu układ upraszcza się do 12 nieliniowych równań, których rozwiązaniem są wartości zmiennych przegubowych odpowiadających zadanej konfiguracji.

$$H = T_n^0(q_1, q_2, \dots, q_n) = \begin{pmatrix} R_n^0 & l_n^0 \\ 0 & 1 \end{pmatrix} \quad (2.1)$$

$$\forall i \in (1 \dots 3) j \in (1 \dots 4) : T_{ij}(q_1, q_2, \dots, q_n) = h_{ij} \quad (2.2)$$

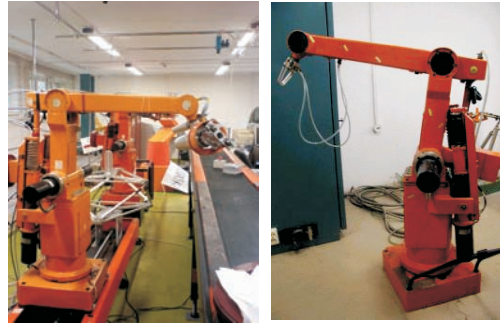
$$\forall k \in (1 \dots n) : q_k = f_k(h_{11}, \dots, h_{34}) \quad (2.3)$$

Rozwiązanie przedstawionych równań jest trudne z kilku powodów. Po pierwsze ich silna nieliniowość uniemożliwia najczęściej znalezienie rozwiązań w postaci rozwikłanej. Po drugie może istnieć wiele rozwiązań, z których należy wybrać najlepsze, a w przypadku manipulatorów redundantnych może istnieć ich nieskończona liczba. Po trzecie, rozwiązanie to, dla określonej struktury kinematycznej, może w ogóle nie istnieć. Metoda rozwiązania odwrotnego zadania kinematyki musi również uwzględniać ograniczenia nałożone na poszczególne zmienne przegubowe, tak aby rozwiązanie zawierało się w przestrzeni roboczej mechanizmu [5]. Rozwiązanie odwrotnego zadania kinematyki wymaga dużej intuicji algebraicznej, pomagającej w optymalnym obraniu kolejności i metod rozwiązywania równań. Pomocna jest tu również intuicja geometryczna, dzięki której w rozwiązaniu uwzględniane są cechy charakterystyczne struktury geometrycznej manipulatora. Rozwiązanie problemu kinematyki odwrotnej musi być podejmowane osobno dla każdej struktury kinematycznej, ponieważ wynika ono w głównej mierze ze specyfiki tej struktury. Podejmowane są jednakże próby znalezienia uniwersalnych metod rozwiązania tego problemu, takie jak oprogramowanie SRAST (Symbolic Robot Arm Solution Tool) [18], są one jednak mało wydajne.

Podczas rozwiązywania zadania kinematyki odwrotnej jesteśmy raczej zainteresowani znalezieniem wyniku w postaci rozwikłanej (2.3), niż rozwiązaniem numerycznym. Równania w tej postaci są wygodniejsze z dwóch powodów, po pierwsze metody te są szybsze obliczeniowo od metod iteracyjnych, po drugie w przypadku niejednoznaczności zadania łatwiej jest wybrać wartości właściwe.

2.2 Opis manipulatora IRp-6

Manipulator IRp-6 jest polskiej produkcji robotem przemysłowym, o strukturze antropomorficznej. W laboratorium Instytutu Automatyki znajdują się dwa manipulatory tego typu: sześciostopniowy¹, zamocowany na postumencie, oraz redundantny, którego podstawa znajduje się na ruchomej taśmie. W prezentowanej pracy analizowany jest pierwszy z tych robotów. Poniżej zaprezentowano opis poszczególnych ogniw robota, wraz z przyporządkowaniem, do poszczególnych par kinematycznych, układów współrzędnych.



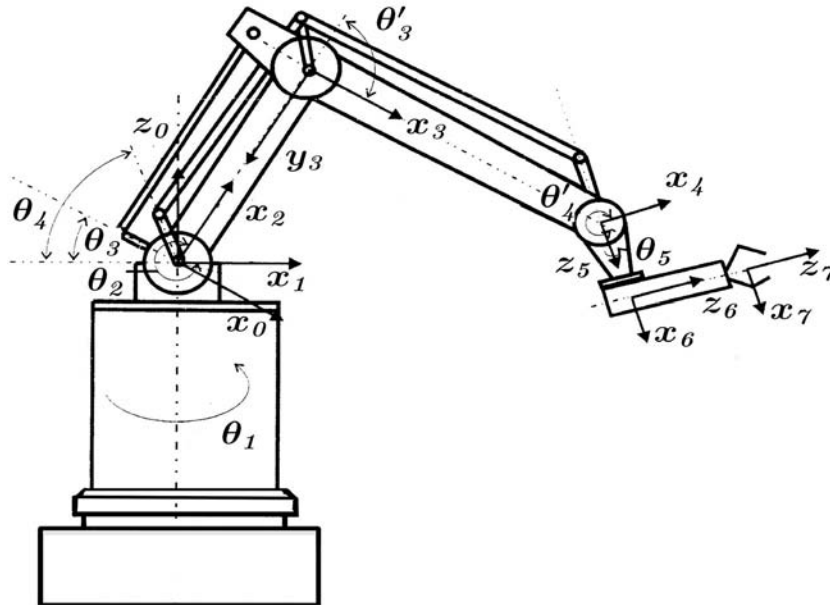
Rysunek 2.1: Manipulator IRp-6

2.2.1 Opis ogniw manipulatora IRp-6

- Kolumna robota ma wysokość $0,7\text{ m}$. U jej podstawy znajduje się przegub powodujący rotację całego manipulatora wokół osi Oz układu związanego z podłożem. Punkt zaczepienia układu współrzędnych tego złącza O_1 pokrywa się z układem bazowym O_0 , który to, dla uproszczenia opisu, znajduje się na górze kolumny. Dzięki temu założeniu w modelu matematycznym manipulatora pomijana jest wysokość kolumny oraz wszelkie parametry z nią związane. Obrót kolumny θ_1 odbywa się w zakresie od -170° do 170° .
- Złącze związane z ruchem przedramienia zaczepione jest w tym samym punkcie, co złącze wywołujące ruch opisanej wcześniej kolumny. Orientacja układu współrzędnych dla tego przegubu jest obrócona o kąt -90° przez co kierunek osi obrotu Oz_2 pokrywa oś Oy_1 . Ruch złącza łączącego przedramię z kolumną θ_2 wywołuje ruch ramienia w płaszczyźnie prostopadłej do podłoża i odbywa się w zakresie od -130° do -50° . Długość przedramienia oznaczona w dalszej części pracy jako a_2 wynosi 0.45 m .

¹tzn. o sześciu stopniach swobody

- Silnik napędzający ruch trzeciego przegubu θ_3 zaczepiony jest w tym samym punkcie co drugiego θ_2 , a jego ruch przekładany jest za pomocą mechanizmu równoległobocznego na ruch ramienia. Do opisu notacji ZDH używany jest względny ruch pomiędzy kolejnymi członami, który wyniesie dla tego przegubu $\theta'_3 = \theta_3 - \theta_2$ (rys.2.2). Kąt obrotu θ_3 może zmieniać się w zakresie od -25° do 40° , a długość tego ramienia oznaczona jako a_3 wynosi 0.67 m .
- Kolejny człon, podobnie jak poprzedni, zbudowany jest na zasadzie równoległoboku, a oś obrotu θ_4 pokrywa się z osiami obrotu θ_2 i θ_3 . Względny ruch pomiędzy czwartym członem, a ramieniem wyznaczamy z zależności $\theta'_4 = \theta_4 - \theta_3 - \frac{\pi}{2}$ (rys. 2.2), gdzie θ_4 zmienia się od -90° do 90° .
- Piąty przegub znajduje się w odległości $d_5 = 0.15$ od poprzedniego z ogniów, a oś jego obrotu pokrywa się z osią Oy poprzedniego ogniwa. Ruch θ_5 ograniczony jest od -180° do 180° .
- Szóste i ostatnie złącze odpowiada za zmianę orientacji końcówki roboczej. Podobnie jak w przypadku kolumny do jego opisu dokonano uproszczenia polegającego na umiejscowieniu początku układu współrzędnych w punkcie zaczepienia piątego układu, a nie środka chwytaka w odległości $d_7 = 0.2$. Kąt obrotu θ_6 zmienia się od -180° do 180° .



Rysunek 2.2: Przyporządkowanie układów współrzędnych do ogniów manipulatora IRp-6

2.2.2 Parametry Denavita-Hartenberga dla IRp-6

Na podstawie powyższego opisu, oraz rysunku (2.2) wyznaczono parametry zmodyfikowanej notacji Denavita-Hartenberg manipulatora IRp-6.

i	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	$-\frac{\pi}{2}$	0	θ_2
3	a_2	0	0	$\theta'_3 = \theta_3 - \theta_2$
4	a_3	0	0	$\theta'_4 = \theta_4 - \theta_3 - \frac{\pi}{2}$
5	0	$-\frac{\pi}{2}$	d_5	θ_5
6	0	$\frac{\pi}{2}$	0	θ_6

Tabela 2.1: zestawienie parametrów ZDH dla manipulatora IRp-6

2.3 Równania kinematyki prostej manipulatora IRp-6

Korzystając z tabeli parametrów ZDH dla manipulatora IRp-6 (2.1) oraz wzoru (1.71) wyznaczamy macierze przekształcenia jednorodnego dla kolejnych par kinematycznych robota.

$$T_1^0 = \begin{pmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.4)$$

$$T_2^1 = \begin{pmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin(\theta_2) & -\cos(\theta_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.5)$$

$$T_3^2 = \begin{pmatrix} \cos(\theta_3 - \theta_2) & -\sin(\theta_3 - \theta_2) & 0 & a_2 \\ \sin(\theta_3 - \theta_2) & \cos(\theta_3 - \theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.6)$$

Aby uprościć postać kolejnej macierzy podstawione zostały następujące przekształcenia, dotyczące kolejnej zmiennej przegubowej, związanej poprzez mechanizm równoległoboczny z czwartym ogniwem $\theta'_4 = \theta_4 - \theta_3 - \frac{\pi}{2}$:

$$\cos\left(-\left(\frac{\pi}{2} + \theta_3 - \theta_4\right)\right) = \cos\left(\frac{\pi}{2} + \theta_3 - \theta_4\right) = -\sin(\theta_3 - \theta_4)$$

$$\sin\left(-\left(\frac{\pi}{2} + \theta_3 - \theta_4\right)\right) = -\sin\left(\frac{\pi}{2} + \theta_3 - \theta_4\right) = -\cos(\theta_3 - \theta_4)$$

$$T_4^3 = \begin{pmatrix} -\sin(\theta_3 - \theta_4) & \cos(\theta_3 - \theta_4) & 0 & a_3 \\ -\cos(\theta_3 - \theta_4) & -\sin(\theta_3 - \theta_4) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.7)$$

$$T_5^4 = \begin{pmatrix} \cos(\theta_5) & -\sin(\theta_5) & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ -\sin(\theta_5) & -\cos(\theta_5) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.8)$$

$$T_6^5 = \begin{pmatrix} \cos(\theta_6) & -\sin(\theta_6) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin(\theta_6) & \cos(\theta_6) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.9)$$

Po złożeniu kolejnych przekształceń według zależności (1.37) otrzymamy macierze przekształcenia jednorodnego dla kolejnych ogniw manipulatora, względem bazowego układu współrzędnych. Dla skrócenia zapisu zostały użyte oznaczenia: $c_i = \cos \theta_i$, $s_i = \sin \theta_i$, oraz $c_{32} = \cos(\theta_3 - \theta_2)$, $s_{32} = \sin(\theta_3 - \theta_2)$.

$$T_2^0 = \begin{pmatrix} c_1 c_2 & -c_1 s_2 & -s_1 & 0 \\ s_1 c_2 & -s_1 s_2 & c_1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.10)$$

$$T_3^0 = \begin{pmatrix} c_1 (c_2 c_{32} - s_2 s_{32}) & -c_1 (c_2 s_{32} + s_2 c_{32}) & -s_1 & c_1 c_2 a_2 \\ s_1 (c_2 c_{32} - s_2 s_{32}) & -s_1 (c_2 s_{32} + s_2 c_{32}) & c_1 & s_1 c_2 a_2 \\ -c_2 s_{32} - s_2 c_{32} & s_2 s_{32} - c_2 c_{32} & 0 & -s_2 a_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.11)$$

$$T_4^0 = \begin{pmatrix} c_1 s_4 & c_1 c_4 & -s_1 & c_1 (a_3 c_3 + a_2 c_2) \\ s_1 s_4 & s_1 c_4 & c_1 & s_1 (a_3 c_3 + a_2 c_2) \\ c_4 & -s_4 & 0 & -a_3 s_3 - a_2 s_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.12)$$

$$T_5^0 = \begin{pmatrix} c_1 s_4 c_5 + s_1 s_5 & -c_1 s_4 s_5 + s_1 c_5 & c_1 c_4 & c_1 (c_4 d_5 + a_3 c_3 + a_2 c_2) \\ s_1 s_4 c_5 - c_1 s_5 & -s_1 s_4 s_5 - c_1 c_5 & s_1 c_4 & s_1 (c_4 d_5 + a_3 c_3 + a_2 c_2) \\ c_4 c_5 & -c_4 s_5 & -s_4 & -s_4 d_5 - a_3 s_3 - a_2 s_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.13)$$

Ostateczne równania określające rozwiązanie prostego zadania kinematyki przyjmują postać:

$$T_6^0 = \begin{pmatrix} (c_1 s_4 c_5 + s_1 s_5) c_6 + c_1 c_4 s_6 & -(c_1 s_4 c_5 + s_1 s_5) s_6 + c_1 c_4 c_6 & \\ (s_1 s_4 c_5 - c_1 s_5) c_6 + s_1 c_4 s_6 & -(s_1 s_4 c_5 - c_1 s_5) s_6 + s_1 c_4 c_6 & \dots \\ c_4 c_5 c_6 - s_4 s_6 & -c_4 c_5 s_6 - s_4 c_6 & \\ 0 & 0 & \\ c_1 s_4 s_5 - s_1 c_5 & c_1 (c_4 d_5 + a_3 c_3 + a_2 c_2) & \\ s_1 s_4 s_5 + c_1 c_5 & s_1 (c_4 d_5 + a_3 c_3 + a_2 c_2) & \\ c_4 s_5 & -s_4 d_5 - a_3 s_3 - a_2 s_2 & \\ 0 & 1 & \end{pmatrix} \quad (2.14)$$

Alternatywną metodą wyznaczenia równań kinematyki prostej jest wykorzystanie przestrzennego operatora pozycji postaci (1.63). Poprzez wymnożenie operatorów związanych z kolejnymi parami kinematycznymi otrzymane zostanie rozwiązanie prostego zadania kinematyki w postaci (2.15).

$$L(\theta, 6) = \begin{pmatrix} R_6^0 & S(l_6^0) R_6^0 \\ O & R_6^0 \end{pmatrix} \quad (2.15)$$

2.4 Równania kinematyki odwrotnej manipulatora IRp-6

Jednym ze sposobów podejścia do zadania kinematyki odwrotnej jest niezależne wyprowadzenie dwóch zależności opisujących macierz przekształcenia jednorodnego między tymi samymi układami. W kolejnym kroku przyrównywane są odpowiadające sobie współczynniki obu macierzy [26, 46].

Według tej zasady wyprowadzone zostały dwa wzory opisujące macierz T_6^1 . Pierwsza z nich otrzymana została przez wymnożenie macierzy przekształcenia jednorodnego układów 1–6 (2.16), drugą natomiast z zależności (2.17). W przekształceniu tym wykorzystano własności odwrotności macierzy przekształcenia jednorodnego (1.38), a symbolem H oznaczono macierz opisującą położenie i orientację końcówki manipulatora (2.1) tzn. dane wejściowe zadania. Omówione macierze mają postać (2.18) i (2.19).

$$(T_6^1)_a = T_2^1 T_3^2 T_4^3 T_5^4 T_6^5 \quad (2.16)$$

$$(T_6^1)_b = T_0^1 H = (T_1^0)^{-1} H \quad (2.17)$$

$$(T_6^1)_a = \begin{pmatrix} s_4 c_5 c_6 + c_4 s_6 & -s_4 c_5 s_6 + c_4 c_6 & s_4 s_5 & c_4 d_5 + a_3 c_3 + a_2 c_2 \\ -s_5 c_6 & s_5 s_6 & c_5 & 0 \\ c_4 c_5 c_6 - s_4 s_6 & -c_4 c_5 s_6 - s_4 c_6 & c_4 s_5 & -s_4 d_5 - a_3 s_3 - a_2 s_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.18)$$

$$\begin{aligned}
(T_6^I)_b &= \begin{pmatrix} c_1 & s_1 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \\
&= \begin{pmatrix} c_1 r_{11} + s_1 r_{21} & c_1 r_{12} + s_1 r_{22} & c_1 r_{13} + s_1 r_{23} & c_1 p_x + s_1 p_y \\ c_1 r_{21} - s_1 r_{11} & c_1 r_{22} - s_1 r_{12} & c_1 r_{23} - s_1 r_{13} & c_1 p_y - s_1 p_x \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.19)
\end{aligned}$$

Porównując poszczególne wartości obu macierzy wyprowadzamy następujące zależności określające wartości zmiennych wewnętrznych:

- Z elementów położonych w drugim wierszu ostatniej kolumny wynika zależność: $c_1 p_y - s_1 p_x = 0$, tak więc $c_1 p_y = s_1 p_x$, a co za tym idzie $\frac{s_1}{c_1} = \frac{p_y}{p_x}$. Wyliczając z danych wartości funkcję arcus tangens² otrzymana zostanie wartość pierwszej zmiennej przegubowej:

$$\theta_1 = \arctan\left(\frac{p_y}{p_x}\right) \quad (2.20)$$

- Kolejna zależność wyznaczona zostanie na podstawie elementów położonych w drugim wierszu trzeciej kolumny: $c_1 r_{23} - s_1 r_{13} = c_5$. Wartości c_1 i s_1 zostały wyznaczone w poprzednim punkcie, natomiast s_5 wyliczyć można z własności jedynki trygonometrycznej. Dzięki temu wartość drugiej zmiennej wewnętrznej zadana jest następującą zależnością:

$$\theta_5 = \arctan\left(\frac{\pm\sqrt{1 - (c_1 r_{23} - s_1 r_{13})^2}}{c_1 r_{23} - s_1 r_{13}}\right) \quad (2.21)$$

- Po porównaniu elementów pierwszej i drugiej kolumny drugiego wiersza otrzymane zostanie równanie: $\frac{c_1 r_{22} - s_1 r_{12}}{c_1 r_{21} - s_1 r_{11}} = \frac{s_5 s_6}{-s_5 c_6}$. Na jego podstawie, przy założeniu iż $s_5 \neq 0$, wyznaczona zostanie kolejna wartość wektora współrzędnych uogólnionych:

$$\theta_6 = \arctan\left(\frac{c_1 r_{22} - s_1 r_{12}}{-(c_1 r_{21} - s_1 r_{11})}\right) \quad (2.22)$$

²W implementacji tych równań zastosowano omówioną wcześniej funkcję $\text{Atan2}(x, y)$.

- Postępując analogicznie, na podstawie pierwszego i trzeciego wiersza trzeciej kolumny wyznaczona zostanie zależność $\frac{c_1 r_{13} + s_1 r_{23}}{r_{33}} = \frac{s_4 s_5}{c_4 s_5}$. Po założeniu iż $s_5 \neq 0$ wyliczone zostanie:

$$\theta_4 = \arctan\left(\frac{c_1 r_{13} + s_1 r_{23}}{r_{33}}\right) \quad (2.23)$$

- W przypadku gdy wartość $\theta_5 = 0$ lub $\theta_5 = \pi$, a co za tym idzie $s_5 = 0$, mamy do czynienia z konfiguracją osobliwą, a macierz T_6^I przyjmuje następującą postać:

$$T_6^I = \begin{pmatrix} \pm s_4 c_6 + c_4 s_6 & \pm s_4 s_6 + c_4 c_6 & 0 & a_2 c_2 + a_3 c_3 \\ 0 & 0 & \pm 1 & 0 \\ \pm c_4 c_6 - s_4 s_6 & \pm c_4 s_6 - s_4 c_6 & 0 & -a_2 s_2 - a_3 s_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.24)$$

Korzystając z wzorów sumy i różnicy funkcji trygonometrycznych, macierz tą przekształcić można do następującej postaci:

$$T_6^I = \begin{pmatrix} \sin(\theta_6 \pm \theta_4) & \cos(\theta_6 \pm \theta_4) & 0 & a_2 c_2 + a_3 c_3 \\ 0 & 0 & \pm 1 & 0 \\ \pm \cos(\theta_6 \pm \theta_4) & \pm \sin(\theta_6 \pm \theta_4) & 0 & -a_2 s_2 - a_3 s_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.25)$$

Dla macierzy tej wyznaczyć można zależności $\frac{c_1 r_{11} + s_1 r_{21}}{c_1 r_{12} + s_1 r_{22}} = \frac{\sin(\theta_6 \pm \theta_4)}{\cos(\theta_6 \pm \theta_4)}$, tak więc:

$$(\theta_6 \pm \theta_4) = \arctan\left(\frac{c_1 r_{11} + s_1 r_{21}}{c_1 r_{12} + s_1 r_{22}}\right) \quad (2.26)$$

Jak widać rozwiązanie odwrotnego zadania kinematyki dla tej konfiguracji jest niejednoznaczne i istnieje nieskończona liczba wartości θ_4 i θ_6 spełniających zadaną zależność. Dla tej sytuacji przyjęto konwencję iż przegub, którego obrót wymaga większej pracy, pozostaje w spoczynku, podczas gdy obrót drugiego z ogniw wyliczany jest z powyższego równania.

- Wyliczenie kolejnego z kątów wiąże się z rozwiązaniem układu równań, wynikających z postaci pierwszego i trzeciego wiersza ostatniej kolumny macierzy

$$\begin{cases} c_4 d_5 + a_3 c_3 + a_2 c_2 & = c_1 p_x + s_1 p_y \\ -s_4 d_5 - a_3 s_3 - a_2 s_2 & = p_z \end{cases} \quad (2.27)$$

$$\begin{cases} a_3 c_3 + a_2 c_2 &= c_1 p_x + s_1 p_y - c_4 d_5 \\ a_3 s_3 + a_2 s_2 &= -(p_z + s_4 d_5) \end{cases}$$

Wprowadzając dodatkowe oznaczenia:

$$\begin{aligned} E &= c_1 p_x + s_1 p_y - c_4 d_5 \\ F &= -(p_z + s_4 d_5) \end{aligned} \quad (2.28)$$

$$\begin{cases} a_3 c_3 &= E - a_2 c_2 \\ a_3 s_3 &= F - a_2 s_2 \end{cases} \quad (2.29)$$

oraz podnosząc obie strony równań do kwadratu:

$$\begin{cases} (a_3 c_3)^2 &= E^2 + (a_2 c_2)^2 - 2E a_2 c_2 \\ (a_3 s_3)^2 &= F^2 + (a_2 s_2)^2 - 2F a_2 s_2 \end{cases}$$

a następnie dodając stronami, otrzymane zostanie równanie postaci:

$$\begin{aligned} a_3^2 &= E^2 + F^2 + a_2^2 - 2E a_2 c_2 - 2F a_2 s_2 \\ E^2 + F^2 + a_2^2 - a_3^2 &= 2E a_2 c_2 + 2F a_2 s_2 \end{aligned} \quad (2.30)$$

Rozpatrując własności geometryczne manipulatora, zauważyć można iż ruch ramion, odpowiadający obrotowi drugiego, trzeciego i czwartego przegubu, odbywa się w jednej płaszczyźnie. Jedną z osi płaszczyzny tej pokrywa się z osią obrotu kolumny, druga natomiast leży w płaszczyźnie wyznaczonej przez Ox i Oy układu bazowego. Wyznaczone powyżej parametry E i F stanowią współrzędne kartezjańskie punktu zaczepienia czwartego ogniwa w zadanej płaszczyźnie, dzięki czemu ich wartości mogą zostać opisane współrzędnymi biegunowymi ζ i φ .

$$\begin{cases} G = 2E a_2 = \zeta \sin(\varphi) \\ H = 2F a_2 = \zeta \cos(\varphi) \end{cases}$$

$$\begin{cases} \zeta = \sqrt{G^2 + H^2} \\ \varphi = \arctan\left(\frac{G}{H}\right) \end{cases}$$

Przekształcając następnie równanie (2.30), poprzez wprowadzenie do niego opisanych zmiennych, otrzymana zostanie ostateczna zależność na θ_2 (2.33)

$$K = E^2 + F^2 + a_2^2 - a_3^2$$

$$K = \zeta \sin(\varphi) \cos(\theta_2) + \zeta \cos(\varphi) \sin(\theta_2) \quad (2.31)$$

$$\begin{cases} \sin(\theta_2 + \varphi) = \frac{K}{\zeta} \\ \cos(\theta_2 + \varphi) = \sqrt{1 - \left(\frac{K}{\zeta}\right)^2} \end{cases} \quad (2.32)$$

$$\theta_2 + \varphi = \arctan\left(\frac{\frac{K}{\zeta}}{\pm\sqrt{1 - \left(\frac{K}{\zeta}\right)^2}}\right)$$

$$\theta_2 = \arctan\left(\frac{\frac{K}{\zeta}}{\pm\sqrt{1 - \left(\frac{K}{\zeta}\right)^2}}\right) - \arctan\left(\frac{G}{H}\right) \quad (2.33)$$

- Korzystając z wyprowadzonego w poprzednim punkcie układu równań, wyliczona zostanie zależność na ostatnią ze zmiennych przegubowych:

$$\begin{cases} a_3 c_3 = E - a_2 c_2 \\ a_3 s_3 = F - a_2 s_2 \end{cases} \quad (2.34)$$

$$\frac{F - a_2 s_2}{E - a_2 c_2} = \frac{s_3}{c_3} \quad (2.35)$$

$$\theta_3 = \arctan\left(\frac{F - a_2 s_2}{E - a_2 c_2}\right) \quad (2.36)$$

Rozdział 3

Kinematyka różniczkowa

3.1 Sterowanie oparte na prędkości

Z matematycznego punktu widzenia równania kinematyki prostej określają funkcję pomiędzy przestrzenią pozycji l_n^0 i orientacji R_n^0 kartezjańskich końcówki roboczej, a przestrzenią pozycji przegubów q . Związki prędkościowe są z kolei wyznaczone poprzez pochodną, a więc jakobian¹, tej funkcji. Jakobian manipulatora o n stopniach swobody jest funkcją macierzową o wymiarze $6 \times n$ i może być traktowany jako wersja wektorowa zwykłej pochodnej funkcji skalarnej. Trzy pierwsze wiersze macierzy jakobianu odpowiadają przekształceniu prędkości kątowej, trzy kolejne prędkości liniowej efektora robota.

Jeżeli równania kinematyki prostej opisane są za pomocą macierzy przekształcenia jednorodnego postaci (3.1), to pochodne tego przekształcenia dają nam wyrażenia określające prędkość kątową (3.2) i liniową (3.3) końcówki, gdzie $S(\omega)$ oznacza omówioną we wstępie pracy macierz skośniesymetryczną wektora prędkości kątowej. Równania te mogą zostać zapisane w interesującej nas postaci, uwzględniającej strukturę jakobianu (3.5).

$$T_n^0(q) = \begin{pmatrix} R_n^0(q) & l_n^0(q) \\ 0 & 1 \end{pmatrix} \quad (3.1)$$

$$S(\omega) = \dot{R}_n^0 (R_n^0)^T \quad (3.2)$$

$$v_n^0 = \dot{l}_n^0 \quad (3.3)$$

$$\begin{cases} \omega_n^0 = J_\omega \dot{q} \\ v_n^0 = J_v \dot{q} \end{cases} \quad (3.4)$$

¹Dla przypomnienia jakobian w robotyce, inaczej niż w matematyce, oznacza macierz jakobiego (macierz pochodnych pierwszego rzędu), a nie jej wyznacznik.

$$\begin{pmatrix} \omega_n^0 \\ v_n^0 \end{pmatrix} = \begin{pmatrix} J_\omega \\ J_v \end{pmatrix} \dot{q} = J_n^0 \dot{q} \quad (3.5)$$

Jeżeli jacobian jest macierzą kwadratową i nieosobliwą, to postać dualną zadania kinematyki różniczkowej otrzymamy poprzez wyznaczenie odwrotności jacobianu.

$$\dot{q} = (J_n^0)^{-1} \begin{pmatrix} \omega_n^0 \\ v_n^0 \end{pmatrix} \quad (3.6)$$

3.2 Wyprowadzenie jacobianu na podstawie macierzy przekształcenia jednorodnego

Poniższe dwa rozdziały przedstawiają dwie metody wyznaczenia jacobianu manipulatora. Jako pierwszą przedstawiono metodę wykorzystującą postać macierzy przekształcenia jednorodnego pomiędzy kolejnymi układami współrzędnych, związanych z poszczególnymi złączami, a układem bazowym. Osobno przedstawione zostały przekształcenia związane z prędkością kątową i liniową. Druga z przedstawionych metod opiera się na operatorach algebry przestrzennej, a konkretnie macierzy pozycji. Wynik końcowy dla obu metod jest identyczny, a wyprowadzenie jacobianu na dwa sposoby umożliwia zweryfikowanie poprawności równań.

3.2.1 Prędkość obrotowa

Zakładamy iż p jest wektorem ustalonym w układzie O_1 (takim że współrzędne $p_1 = \text{const}$), który obraca się względem układu O_0 . Współrzędne tego wektora (p_0) w układzie O_0 wyrażamy następującą zależnością:

$$p_0 = R_1^0(t)p_1 + l_1^0 \quad (3.7)$$

$$\dot{p}_0 = \dot{R}_1^0(t)p_1 + \dot{l}_1^0 \quad (3.8)$$

Z drugiej strony prędkość końcówki wektora p względem układu O_0 (\dot{p}_0) wyrazić można jako złożenie prędkości liniowej całego układu O_1 (v), oraz iloczynu wektorowego prędkości kątowej układu O_1 (ω) i ramienia (r), będącego odległością między końcówką wektora, a początkiem układu O_1 , wyrażoną w układzie O_0 ($r = R_1^0 p_1$).

$$\dot{p}_0 = \omega \times r + v = S(\omega)r + v = S(\omega)R_1^0(t)p_1 + v \quad (3.9)$$

Z porównania zależności (3.8) i (3.9) widać iż:

$$v = \dot{l}_1^0 \quad (3.10)$$

$$\dot{R}_1^0(t) = S(\omega)R_1^0(t) \quad (3.11)$$

$$S(\omega) = \dot{R}(t)R(t)^T \quad (3.12)$$

$$S(\omega)^T = R(t)\dot{R}(t)^T \quad (3.13)$$

Wychodząc z warunku ortogonalności macierzy rotacji $R(t)$ można bezpośrednio dowieść, iż macierz określona równaniem (3.12) spełnia warunki macierzy skośniesymetrycznej.

$$R(t)R(t)^T = I \quad (3.14)$$

$$\dot{R}(t)R(t)^T + R(t)\dot{R}(t)^T = 0 \quad (3.15)$$

$$S(\omega) + S(\omega)^T = 0 \quad (3.16)$$

Jak zaznaczono w poprzednich rozdziałach, operacji złożenia rotacji dokonujemy przy pomocy zależności (1.15). Wyliczając jego pochodną otrzymujemy wzór (3.18). Człon występujący po lewej stronie tej zależności możemy zapisać w postaci (3.19), a kolejne dwa po prawej za pomocą zależności (3.20) i (3.21). W przekształceniach tych skorzystano z wyprowadzonego wcześniej wzoru (1.49), oraz własności (1.48) macierzy skośniesymetrycznej.

$$R_2^0 = R_1^0 R_2^1 \quad (3.17)$$

$$\dot{R}_2^0 = \dot{R}_1^0 R_2^1 + R_1^0 \dot{R}_2^1 \quad (3.18)$$

$$\dot{R}_2^0 = S(\omega_2^0) R_2^0 \quad (3.19)$$

$$\dot{R}_1^0 R_2^1 = S(\omega_1^0) R_1^0 R_2^1 = S(\omega_1^0) R_2^0 \quad (3.20)$$

$$\begin{aligned} R_1^0 \dot{R}_2^1 &= R_1^0 S(\omega_2^1) R_2^1 = R_1^0 S(\omega_2^1) (R_1^0)^T R_1^0 R_2^1 = \\ &= S(R_1^0 \omega_2^1) R_1^0 R_2^1 = S(R_1^0 \omega_2^1) R_2^0 \end{aligned} \quad (3.21)$$

$$S(\omega_2^0) R_2^0 = (S(\omega_1^0) + S(R_1^0 \omega_2^1)) R_2^0 = S(\omega_1^0 + R_1^0 \omega_2^1) R_2^0 \quad (3.22)$$

Na podstawie powyższych przekształceń wyznaczony został wzór definiujący operację złożenia prędkości kątowych (3.24) mówiący, iż mogą one zostać zsumowane pod warunkiem, iż wyrażone są względem tego samego układu współrzędnych.

$$\omega_2^0 = \omega_1^0 + R_1^0 \omega_2^1 \quad (3.23)$$

$$\omega_n^0 = \omega_1^0 + R_1^0 \omega_2^1 + R_2^0 \omega_3^2 + \dots + R_{n-1}^0 \omega_n^{n-1} \quad (3.24)$$

Prędkość kątowna związana z dowolnym przegubem zadana jest następującą wartością, gdzie k jest osią obrotu przegubu:

$$\begin{cases} \omega_i^{i-1} = 0 & \text{przegub przesuwany} \\ \omega_i^{i-1} = \dot{q}_i k & \text{przegub obrotowy} \end{cases} \quad (3.25)$$

Podstawiając te wartości do równania (3.24) otrzymana zostanie zależność (3.26), gdzie $z_i = R_i^0 k$, jest wersorem osi Oz_i wyrażonym w układzie bazowym, a σ_i przyjmuje wartość 1 dla przegubu obrotowego i 0 dla przesuwnego.

$$\omega_n^0 = \sigma_1 \dot{q}_1 R_1^0 k + \sigma_2 \dot{q}_2 R_2^0 k + \dots + \sigma_n \dot{q}_n R_n^0 k = \sum_{i=1}^n \sigma_i \dot{q}_i z_i \quad (3.26)$$

Reasumując powyższe rozważania stwierdzimy, iż trzy pierwsze wiersze Jakobianu, odpowiadające prędkości kątovej, przyjmą następującą postać (3.27). Do ich wyznaczenia wystarczy znajomość wektorów z_i zdefiniowanych przez trzecią kolumnę macierzy rotacji R_i^0 .

$$J_\omega = (\sigma_1 z_1 \dots \sigma_n z_n) \quad (3.27)$$

3.2.2 Prędkość Liniowa

Wartość prędkości liniowej końcówki wyznaczana jest z zależności (3.3). Na podstawie reguły na pochodną superpozycji odwzorowań otrzymujemy zależność (3.28) uwzględniającą prędkości związane ze wszystkimi zmiennymi przegubowymi manipulatora.

$$\dot{l}_n^0 = \sum_{i=1}^n \frac{\partial l_n^0}{\partial q_i} \dot{q}_i \quad (3.28)$$

W przypadku gdy przegub i jest translacyjny, to prędkość związana z jego ruchem przekłada się na ruch końcówki, o tej samej prędkości i kierunku wyznaczonym przez wartość osi przesuwu z_i . Kolumna i macierzy J_v , odpowiadająca temu przegubowi przyjmie wówczas postać:

$$J_{vi} = R_i^0 k = z_i \quad (3.29)$$

Natomiast, w momencie gdy przegub i jest obrotowy i porusza się z prędkością ω_i , wynikowa prędkość liniowa końcówki wyliczona zostanie z iloczynu wektorowego ω_i , oraz ramienia łączącego oba układy: $v_n = \omega_i \times l_n^i$. Podsumowując odpowiedni element macierzy jakobianu wyniesie:

$$J_{vi} = z_i \times (l_n^0 - l_i^0) \quad (3.30)$$

Wartość wektora l_i^0 wyznaczona jest przez ostatnią kolumnę macierzy przekształcenia jednorodnego T_i^0 , natomiast wartość z_i , analogicznie jak powyżej, przez trzecią kolumnę macierzy rotacji R_i^0 . Tak więc macierz jacobianowa może być w całości wyznaczona poprzez złożenie trzecich i czwartych kolumn macierzy przekształcenia jednorodnego między kolejnymi przegubami manipulatora, a układem bazowym [44].

3.3 Wyprowadzenie jacobianu na podstawie operatorów algebry przestrzennej

Przestrzenna macierz pozycji $L(k-1, k)$ określona zależnością (1.63) jest złożeniem przestrzennej macierzy odległości $\Phi(k-1, k)$, będącej opisem odległości układów O_{k-1} i O_k , oraz macierzy kosinusów kierunkowych $A(k-1, k)$, odpowiadającej opisowi ich wzajemnej orientacji. Przy pomocy macierzy $L(k-1, k)$ przełożenie wektora prędkości pomiędzy kolejnymi przegubami (3.31) opisane może zostać następującym zwięzłym zapisem (3.32).

$$\begin{cases} \omega_k = R_k^{k-1} \omega_{k-1} \\ v_k = R_k^{k-1} v_{k-1} + R_k^{k-1} \omega_{k-1} \times l_k^{k-1} \end{cases} \quad (3.31)$$

$$V_k = (A(k-1, k) \Phi(k-1, k))^T V_{k-1} = (L(k-1, k))^T V_{k-1} \quad (3.32)$$

Jeżeli w powyższej zależności uwzględniona zostanie prędkość wprowadzona przez k -te złącze (\dot{q}_k) to przyjmie ona postać (3.33), gdzie H jest wektorem rzutowania złącza, którego wartość zadana jest równaniami (1.66).

$$V_k = (L(k-1, k))^T V_{k-1} + H_k^T \dot{q}_k \quad (3.33)$$

Równanie (3.33) umożliwia rekurencyjne przełożenie wektorów prędkości związanych z kolejnymi złączami manipulatora, na końcówkę robota [20]. Oczywiście postać ta jest mało wygodna w pętli sterowania, gdzie żądana jest raczej jego rozwinięta, zawierająca jacobian postać, która uzyskana zostanie poprzez rozwinięcie i uporządkowanie równania (3.33) względem kolejnych zmiennych przegubowych. Końcowy wzór określający jacobian przyjmuje następującą postać:

$$J = \begin{pmatrix} L(1, n)^T H_1^T & L(2, n)^T H_2^T & \dots & L(n-1, n)^T H_{n-1}^T \end{pmatrix} \quad (3.34)$$

Wyliczone przy pomocy równań (3.33) prędkości są wyrażone względem lokalnych układów współrzędnych, tak więc wyznaczony jacobian jest wyrażony względem ostatniego (szóstego) układu, związanego z końcówką roboczą. Aby wyrazić macierz jacobianową względem dowolnego (w szczególności bazowego)

układu współrzędnych, należy wymnożyć ją lewostronnie poprzez odpowiednią macierz kosinusów kierunkowych.

$$J_n^0 = A(\theta, n) J_0^n \quad (3.35)$$

Zasadniczą zaletą przedstawionej metody wyprowadzenia jakobianu jest ujęcie, w jednej zależności, wszystkich wierszy jakobianu (odpowiadających prędkości liniowej i kątowej), oraz zwarty i czytelny analityczny zapis metody.

3.4 Postać jakobianu dla manipulatora IRp-6

Wykorzystując obydwie powyższe metody wyznaczony został jakobian manipulatora IRp-6 o sześciu stopniach swobody. Poniżej przedstawiono pośrednie wyniki obu metod. W zależnościach tych, w celu uproszczenia zapisu, użyto następujące oznaczenia: $c_i = \cos(\theta_i)$, $s_i = \sin(\theta_i)$, $c_{32} = \cos(\theta_3 - \theta_2)$, $s_{32} = \sin(\theta_3 - \theta_2)$, $c_{34} = \cos(\theta_3 - \theta_4)$, $s_{34} = \sin(\theta_3 - \theta_4)$.

• Metoda 1

Wszystkie przeguby rozważanego manipulatora IRp-6 są obrotowe, tak więc poszczególne elementy macierzy jakobianu wyznaczone zostaną na podstawie zależności (3.27), (3.30). Wektory $z_i = R_i^0 k$ będące, wyrażonymi w układzie bazowym, wersorami osi obrotu przegubów, oraz l_i^0 będące wektorami odległości między układem związanym z i -tym przegubem, a układem bazowym, wyznaczone zostały na podstawie, otrzymanych przy rozwiązywaniu prostego zadania kinematyki, macierzy przekształcenia jednorodnego T_i^0 , dla $i \in (1 \dots 6)$: (2.4), (2.10)-(2.14). Wektory z_i odpowiadają trzem pierwszym elementom trzeciej kolumny macierzy T_i^0 , natomiast l_i^0 trzem pierwszym elementom ostatniej kolumny.

– z_i - Równoważne trzem pierwszym wierszom jakobianu - J_ω .

$$J_\omega = (z_1 \quad z_2 \quad z_3 \quad z_4 \quad z_5 \quad z_6)$$

$$z_1 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad z_2 = \begin{pmatrix} -s_1 \\ c_1 \\ 0 \end{pmatrix} \quad z_3 = \begin{pmatrix} -s_1 \\ c_1 \\ 0 \end{pmatrix}$$

$$z_4 = \begin{pmatrix} -s_1 \\ c_1 \\ 0 \end{pmatrix} \quad z_5 = \begin{pmatrix} c_1 c_4 \\ s_1 c_4 \\ -s_4 \end{pmatrix} \quad z_6 = \begin{pmatrix} c_1 s_4 s_5 - s_1 c_5 \\ s_1 s_4 s_5 + c_1 c_5 \\ c_4 s_5 \end{pmatrix}$$

- \mathbf{l}_i^0 - Wektor odległości pomiędzy układem bazowym, a układami współrzędnych związanych z kolejnymi złączami robota.

$$l_1^0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$l_2^0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$l_3^0 = \begin{pmatrix} c_1 c_2 a_2 \\ s_1 c_2 a_2 \\ -s_2 a_2 \end{pmatrix}$$

$$l_4^0 = \begin{pmatrix} c_1 (a_3 c_3 + a_2 c_2) \\ s_1 (a_3 c_3 + a_2 c_2) \\ -a_3 s_3 - a_2 s_2 \end{pmatrix}$$

$$l_5^0 = \begin{pmatrix} c_1 (c_4 d_5 + a_3 c_3 + a_2 c_2) \\ s_1 (c_4 d_5 + a_3 c_3 + a_2 c_2) \\ -s_4 d_5 - a_3 s_3 - a_2 s_2 \end{pmatrix} \quad l_6^0 = \begin{pmatrix} c_1 (c_4 d_5 + a_3 c_3 + a_2 c_2) \\ s_1 (c_4 d_5 + a_3 c_3 + a_2 c_2) \\ -s_4 d_5 - a_3 s_3 - a_2 s_2 \end{pmatrix}$$

- $\mathbf{l}_6^i = \mathbf{l}_6^0 - \mathbf{l}_i^0$ - Wektor odległości pomiędzy układem związanym z końcówką manipulatora, a układami współrzędnych związanych z kolejnymi złączami.

$$l_6^0 - l_1^0 = \begin{pmatrix} c_1 (c_4 d_5 + a_3 c_3 + a_2 c_2) \\ s_1 (c_4 d_5 + a_3 c_3 + a_2 c_2) \\ -s_4 d_5 - a_3 s_3 - a_2 s_2 \end{pmatrix}$$

$$l_6^0 - l_2^0 = \begin{pmatrix} c_1 (c_4 d_5 + a_3 c_3 + a_2 c_2) \\ s_1 (c_4 d_5 + a_3 c_3 + a_2 c_2) \\ -s_4 d_5 - a_3 s_3 - a_2 s_2 \end{pmatrix}$$

$$l_6^0 - l_3^0 = \begin{pmatrix} c_1 (c_4 d_5 + a_3 c_3) \\ s_1 (c_4 d_5 + a_3 c_3) \\ -s_4 d_5 - a_3 s_3 \end{pmatrix}$$

$$l_6^0 - l_4^0 = \begin{pmatrix} c_1 c_4 d_5 \\ s_1 c_4 d_5 \\ -s_4 d_5 \end{pmatrix}$$

$$l_6^0 - l_5^0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$l_6^0 - l_6^0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

- $\mathbf{z}_i \times \mathbf{l}_6^i$ - Równoważne kolejnym trzem wierszom jakobianu - J_v .

$$J_v = \begin{pmatrix} z_1 \times l_6^1 & z_2 \times l_6^2 & z_3 \times l_6^3 & z_4 \times l_6^4 & z_5 \times l_6^5 & z_6 \times l_6^6 \end{pmatrix}$$

$$z_1 \times l_6^1 = \begin{pmatrix} -s_1 (c_4 d_5 + a_3 c_3 + a_2 c_2) \\ c_1 (c_4 d_5 + a_3 c_3 + a_2 c_2) \\ 0 \end{pmatrix}$$

$$z_2 \times l_6^2 = \begin{pmatrix} -c_1 (s_4 d_5 + a_3 s_3 + a_2 s_2) \\ -s_1 (s_4 d_5 + a_3 s_3 + a_2 s_2) \\ -(c_4 d_5 + a_3 c_3 + a_2 c_2) \end{pmatrix}$$

$$z_3 \times l_6^3 = \begin{pmatrix} -c_1 (s_4 d_5 + a_3 s_3) \\ -s_1 (s_4 d_5 + a_3 s_3) \\ -(c_4 d_5 + a_3 c_3) \end{pmatrix}$$

$$z_4 \times l_6^4 = \begin{pmatrix} -c_1 s_4 d_5 \\ -s_1 s_4 d_5 \\ -c_4 d_5 \end{pmatrix}$$

$$z_5 \times l_6^5 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$z_6 \times l_6^6 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

• Metoda 2

W metodzie opartej na operatorach algebry przestrzennej macierz jacobianu, wyrażona w układzie związanym z końcówką roboczą, otrzymana zostanie na podstawie zależności (3.34). Aby uzyskać postać jacobianu wyrażonego względem bazowego układu współrzędnych, należy ją wymnożyć lewostronnie poprzez macierz kosinusów kierunkowych $A(0, 6)$ (3.35).

- \mathbf{H}_i - Wektor rzutowania prędkości związanej z i-tym złączem.

$$H_1 = H_2 = H_3 = H_4 = H_5 = H_6 = (001000)$$

- $\mathbf{L}(\mathbf{k} - \mathbf{1}, \mathbf{k})$ - Przestrzenna macierz pozycji opisująca położenie i orientację układu O_{k-1} względem O_k .

$$L(0, 1) = \begin{pmatrix} c_1 & -s_1 & 0 & 0 & 0 & 0 \\ s_1 & c_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_1 & -s_1 & 0 \\ 0 & 0 & 0 & s_1 & c_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$L(1,2) = \begin{pmatrix} c_2 & -s_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -s_2 & -c_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_2 & -s_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -s_2 & -c_2 & 0 \end{pmatrix}$$

$$L(2,3) = \begin{pmatrix} c_{32} & -s_{32} & 0 & 0 & 0 & 0 \\ s_{32} & c_{32} & 0 & 0 & 0 & -a_2 \\ 0 & 0 & 1 & -a_2 s_{32} & a_2 c_{32} & 0 \\ 0 & 0 & 0 & c_{32} & -s_{32} & 0 \\ 0 & 0 & 0 & s_{32} & c_{32} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$L(3,4) = \begin{pmatrix} -s_{34} & c_{34} & 0 & 0 & 0 & 0 \\ -c_{34} & -s_{34} & 0 & 0 & 0 & -a_3 \\ 0 & 0 & 1 & -a_3 c_{34} & -a_3 s_{34} & 0 \\ 0 & 0 & 0 & -s_{34} & c_{34} & 0 \\ 0 & 0 & 0 & -c_{34} & -s_{34} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$L(4,5) = \begin{pmatrix} c_5 & -s_5 & 0 & -d_5 s_5 & -d_5 c_5 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -s_5 & -c_5 & 0 & -d_5 c_5 & d_5 s_5 & 0 \\ 0 & 0 & 0 & c_5 & -s_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -s_5 & -c_5 & 0 \end{pmatrix}$$

$$L(5,6) = \begin{pmatrix} c_6 & -s_6 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ s_6 & c_6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_6 & -s_6 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & s_6 & c_6 & 0 \end{pmatrix}$$

- $\mathbf{L}(\mathbf{i}, \mathbf{6})^T \mathbf{H}_i^T$ - Kolejne kolumny jacobianu wyrażonego w układzie związanym z końcówką robota.

$$J_0^n = (j_1 \quad j_2 \quad j_3 \quad j_4 \quad j_5 \quad j_6)$$

$$j_1 = L(1, 6)^T H_1^T = \begin{pmatrix} c_6 c_4 c_5 - s_4 s_6 \\ -c_4 c_5 s_6 - s_4 c_6 \\ c_4 s_5 \\ -c_6 s_5 (a_3 c_3 + c_2 a_2 + d_5 c_4) \\ s_5 s_6 (a_3 c_3 + c_2 a_2 + d_5 c_4) \\ c_5 (a_3 c_3 + c_2 a_2 + d_5 c_4) \end{pmatrix}$$

$$j_2 = L(2, 6)^T H_2^T =$$

$$\begin{pmatrix} -c_6 s_5 \\ s_5 s_6 \\ c_5 \\ -(d_5 + a_3 c_{34} - a_2 s_{32} s_{34} + a_2 c_{32} c_{34}) c_5 c_6 - (a_3 s_{34} + a_2 s_{32} c_{34} + a_2 c_{32} s_{34}) s_6 \\ (d_5 + a_3 c_{34} - a_2 s_{32} s_{34} + a_2 c_{32} c_{34}) c_5 s_6 - (a_3 s_{34} + a_2 s_{32} c_{34} + a_2 c_{32} s_{34}) c_6 \\ -d_5 s_5 + (-a_3 c_{34} + a_2 s_{32} s_{34} - a_2 c_{32} c_{34}) s_5 \end{pmatrix}$$

$$j_3 = L(3, 6)^T H_3^T = \begin{pmatrix} -c_6 s_5 \\ s_5 s_6 \\ c_5 \\ (-d_5 c_5 - a_3 c_{34} c_5) c_6 - a_3 s_{34} s_6 \\ -(-d_5 c_5 - a_3 c_{34} c_5) s_6 - a_3 s_{34} c_6 \\ -d_5 s_5 - a_3 c_{34} s_5 \end{pmatrix}$$

$$j_4 = L(4, 6)^T H_4^T = \begin{pmatrix} -c_6 s_5 \\ s_5 s_6 \\ c_5 \\ -c_6 d_5 c_5 \\ s_6 d_5 c_5 \\ -d_5 s_5 \end{pmatrix}$$

$$j_5 = L(5,6)^T H_5^T = \begin{pmatrix} s_6 \\ c_6 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad j_6 = L(6,6)^T H_6^T = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

- **A(0, 6)** - Kolejne kolumny macierzy kosinusów kierunkowych pomiędzy układem bazowym, a układem związanym końcówką manipulatora.

$$A(0,6) = (a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6)$$

$$a_1 = \begin{pmatrix} c_6 c_5 s_4 c_1 + s_6 c_4 c_1 + c_6 s_1 s_5 \\ s_6 c_4 s_1 + c_6 c_5 s_4 s_1 - c_6 c_1 s_5 \\ c_6 c_4 c_5 - s_4 s_6 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$a_2 = \begin{pmatrix} c_6 c_4 c_1 - s_6 c_5 s_4 c_1 - s_6 s_1 s_5 \\ -s_6 c_5 s_4 s_1 + c_6 c_4 s_1 + s_6 c_1 s_5 \\ -s_6 c_4 c_5 - s_4 c_6 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$a_3 = \begin{pmatrix} s_5 s_4 c_1 - s_1 c_5 \\ s_5 s_4 s_1 + c_1 c_5 \\ c_4 s_5 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$a_4 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ c_6 c_5 s_4 c_1 + s_6 c_4 c_1 + c_6 s_1 s_5 \\ s_6 c_4 s_1 + c_6 c_5 s_4 s_1 - c_6 c_1 s_5 \\ c_6 c_4 c_5 - s_4 s_6 \end{pmatrix}$$

$$a_5 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ c_6 c_4 c_1 - s_6 c_5 s_4 c_1 - s_6 s_1 s_5 \\ -s_6 c_5 s_4 s_1 + c_6 c_4 s_1 + s_6 c_1 s_5 \\ -s_6 c_4 c_5 - s_4 c_6 \end{pmatrix}$$

$$a_6 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ s_5 s_4 c_1 - s_1 c_5 \\ s_5 s_4 s_1 + c_1 c_5 \\ c_4 s_5 \end{pmatrix}$$

• Końcowa postać macierzy jacobianu

Reasumując, przy wykorzystaniu obydwu powyższych metod, wyznaczone zostały następujące końcowe równania określające macierz jacobianu względem bazowego układu współrzędnych:

$$J'_1 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ -s_1 (c_4 d_5 + a_3 c_3 + a_2 c_2) \\ c_1 (c_4 d_5 + a_3 c_3 + a_2 c_2) \\ 0 \end{pmatrix} \quad J'_2 = \begin{pmatrix} -s_1 \\ c_1 \\ 0 \\ -c_1 (s_4 d_5 + a_3 s_3 + a_2 s_2) \\ -s_1 (s_4 d_5 + a_3 s_3 + a_2 s_2) \\ -(c_4 d_5 + a_3 c_3 + a_2 c_2) \end{pmatrix}$$

$$\begin{aligned}
J'_3 &= \begin{pmatrix} -s_1 \\ c_1 \\ 0 \\ -c_1(s_4 d_5 + a_3 s_3) \\ -s_1(s_4 d_5 + a_3 s_3) \\ -(c_4 d_5 + a_3 c_3) \end{pmatrix} & J'_4 &= \begin{pmatrix} -s_1 \\ c_1 \\ 0 \\ -c_1 s_4 d_5 \\ -s_1 s_4 d_5 \\ -c_4 d_5 \end{pmatrix} \\
J'_5 &= \begin{pmatrix} c_1 c_4 \\ s_1 c_4 \\ -s_4 \\ 0 \\ 0 \\ 0 \end{pmatrix} & J'_6 &= \begin{pmatrix} c_1 s_4 s_5 - s_1 c_5 \\ s_1 s_4 s_5 + c_1 c_5 \\ c_4 s_5 \\ 0 \\ 0 \\ 0 \end{pmatrix}
\end{aligned} \tag{3.36}$$

Struktura sterowania oparta na jakobianie opisana jest zależnością (3.5), a jako q_i rozumiana jest w tym równaniu zmienna przegubowa łącząca układy związane z ogniwnem (i) i $(i+1)$. W przypadku manipulatora IRp-6 sprawa jest nieco bardziej skomplikowana, z uwagi na strukturę równoległoboczną opisaną na stronie (38). Obrót trzeciego i czwartego przegubu powoduje rotację układu związanego z tym przegubem względem układu związanego z kolumną, a nie układu bezpośrednio go poprzedzającego. Zmienne wewnętrzne manipulatora IRp-6, oraz ich pochodne, uwzględniające strukturę równoległoboczną przyjmują następującą postać (3.37)

$$q' = \begin{pmatrix} q_1 \\ q_2 \\ q_3 - q_2 \\ q_4 - q_3 - \frac{\pi}{2} \\ q_5 \\ q_6 \end{pmatrix} \quad \dot{q}' = \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 - \dot{q}_2 \\ \dot{q}_4 - \dot{q}_3 \\ \dot{q}_5 \\ \dot{q}_6 \end{pmatrix} \tag{3.37}$$

Poprzez wyznaczenie i uporządkowanie zależności (3.5) względem poszczególnych zmiennych przegubowych, otrzymamy alternatywną postać jakobianu (3.38). Struktura oparta na jakobianie w postaci (3.39) umożliwia zbudowanie struktury sterowania, w której każda z zmiennych przegubowych występuje w postaci niezależnej.

$$\begin{pmatrix} \omega \\ v \end{pmatrix} = J' \dot{q}' = J \dot{q} \tag{3.38}$$

$$\begin{aligned}
J_1 &= \begin{pmatrix} 0 \\ 0 \\ 1 \\ -s_1(c_4d_5 + a_3c_3 + a_2c_2) \\ c_1(c_4d_5 + a_3c_3 + a_2c_2) \\ 0 \end{pmatrix} & J_2 &= \begin{pmatrix} 0 \\ 0 \\ 0 \\ -a_2c_1s_2 \\ -a_2s_1s_2 \\ -a_2c_2 \end{pmatrix} \\
J_3 &= \begin{pmatrix} 0 \\ 0 \\ 0 \\ -a_3c_1s_3 \\ -a_3s_1s_3 \\ -a_3c_3 \end{pmatrix} & J_4 &= \begin{pmatrix} -s_1 \\ c_1 \\ 0 \\ -c_1s_4d_5 \\ -s_1s_4d_5 \\ -c_4d_5 \end{pmatrix} \\
J_5 &= \begin{pmatrix} c_1c_4 \\ s_1c_4 \\ -s_4 \\ 0 \\ 0 \\ 0 \end{pmatrix} & J_6 &= \begin{pmatrix} c_1s_4s_5 - s_1c_5 \\ s_1s_4s_5 + c_1c_5 \\ c_4s_5 \\ 0 \\ 0 \\ 0 \end{pmatrix}
\end{aligned} \tag{3.39}$$

Jeżeli interesuje nas przełożenie wektora prędkości uogólnionych na końcówkę przyczepionego do chwytaka narzędzia, a nie, jak poprzednio, umownego punktu zaczepienia ostatniego ogniwa, to macierz jakobianu (3.39) przemnożona być musi przez przestrzenną macierz odległości (3.42). W macierzy tej elementy macierzy R_7^6 określają zadaną rotację narzędzia, natomiast poprzez składowe wektora l' oznaczono przesunięcie końcówki narzędzia od środka efektoru (l) wyrażone względem bazowego układu współrzędnych. W wektorze tym należy również uwzględnić faktyczną długość ostatniego ogniwa d_7 , mierzoną wzdłuż osi obrotu chwytaka.

$$l = \begin{pmatrix} l_x \\ l_y \\ l_z + d_7 \end{pmatrix} \tag{3.40}$$

$$l' = R_6^0 R_7^6 l \tag{3.41}$$

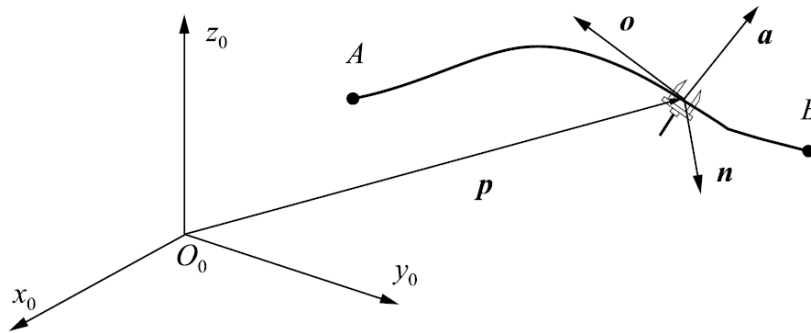
$$L(6, 7) = \begin{pmatrix} R_7^6 & S(l') R_7^6 \\ 0 & R_7^6 \end{pmatrix} \tag{3.42}$$

3.5 Jakobian analityczny

W poprzedniej części pracy wyprowadzono strukturę jacobianu geometrycznego, który jest ściśle związany z parametrami kinematycznymi manipulatora. W niniejszym podrozdziale omówiona zostanie struktura jacobianu analitycznego oraz jego związek z jacobianem geometrycznym.

Jednym z podstawowych zadań dotyczących programowania manipulatorów jest problem planowania trajektorii, polegający na przeprowadzeniu końcówki roboczej robota wzdłuż zadanej trajektorii w przestrzeni kartezjańskiej. Aby tego dokonać konieczne jest wyznaczenie położenia i orientacji chwytaka, względem bazowego układu współrzędnych, w każdym punkcie tej trajektorii. Jeżeli pozycje te są zadawane w postaci macierzy przekształcenia jednorodnego, to do ich wyznaczenia konieczne jest podanie wartości czterech trójwymiarowych wektorów:

- **p** - wektor odległości między końcówką roboczą, a początkiem układu bazowego, wyznaczony przez wektor przesunięcia l_n^0 .
- **a** - wektor zbliżenia, który jest równoległy do płaszczyzny wyznaczonej przez szczęki chwytaka i wyznacza kierunek zbliżania się do obiektu. Wektor ten wyznaczony jest przez trzecią kolumnę macierzy rotacji R_n^0 .
- **o** - wektor orientacji, który leży w płaszczyźnie chwytaka i wyznacza kierunek otwierania i zamykania szczęk, opisany przez drugą kolumnę macierzy rotacji R_n^0 .
- **n** - wektor normalny uzupełniający oba poprzednie wektory do prawoskrętnego układu współrzędnych.



Rysunek 3.1: Wyznaczanie trajektorii w układzie kartezjańskim [25]

O ile wyznaczenie wektora p nie nastęcza żadnych trudności, to przyporządkowanie wartości pozostałym parametrom stanowi istotny problem. Wynika on z konieczności uwzględnienia ograniczeń związanych z ortonormalnością wektorów ($n o a$). Ponadto wyobrażenia projektanta może zawieść przy próbie poprawnego przypisania tychże wektorów na trajektorii.

Aby rozwiązać ten problem, przy planowaniu trajektorii stosowane są na ogół inne, niż macierz rotacji reprezentacje obrotów, wymagające mniejszej ilości parametrów. Najczęściej stosowanymi w tym przypadku reprezentacjami są opisane w rozdziale pierwszym zestawy kątów Eulera. Pozycja, przy tym założeniu, opisana jest przez wektor (3.43), gdzie Ω jest minimalnym (trójelementowym) opisem orientacji, a p wektorem odległości. Każda składowa wektora g wyznaczana jest niezależnie, więc zadanie docelowej trajektorii nie stanowi już problemu.

$$g = \begin{pmatrix} \Omega \\ p \end{pmatrix} \quad (3.43)$$

Wektor g zdefiniowany jest w przestrzeni operacyjnej manipulatora, a więc przestrzeni związanej z wykonywanym zadaniem. Bezpośrednie odwzorowanie pomiędzy tym wektorem, a przestrzenią współrzędnych uogólnionych ($f_A(\cdot)$), jest w większości przypadków trudne do wyznaczenia. Aby wyliczyć tę funkcję konieczne jest na ogół przeliczenie, poprzez proste zadanie kinematyki pozycyjnej ($f_P(\cdot)$), wektorów ($n o a$), a następnie przekształcenie ich na odpowiadający im wektor Ω (przekształcenia te ($f_L(\cdot)$) zadane są zależnościami (1.22)-(1.23) i (1.25)-(1.26)) (3.44). Prędkości związane z wykonywanym zadaniem wyznaczmy przez zróżniczkowanie wektora g (3.46).

$$g = f_A(q) = f_L(f_P(q)) \quad (3.44)$$

$$\begin{cases} \dot{\Omega} = \frac{\partial \Omega}{\partial q} \dot{q} = J_o(q) \dot{q} \\ \dot{p} = \frac{\partial p}{\partial q} \dot{q} = J_p(q) \dot{q} \end{cases} \quad (3.45)$$

$$\dot{g} = \begin{pmatrix} \dot{\Omega} \\ \dot{p} \end{pmatrix} = \begin{pmatrix} J_o(q) \\ J_p(q) \end{pmatrix} \dot{q} = J_A(q) \dot{q} \quad (3.46)$$

J_A w powyższym wzorze oznacza jakobian analityczny, który w ogólności różnić może się od jakobianu geometrycznego (3.47). Rozbieżność ta wynika z różnicy pochodnych orientacji wyrażonych w przestrzeni operacyjnej ($\dot{\Omega}$) i

przestrzeni złącz (ω). Należy w tym miejscu zaznaczyć, iż rozbieżności te zależą jedynie od przyjętej reprezentacji orientacji, a więc część odpowiadająca wektorowi położenia obu jacobianów jest identyczna.

$$\begin{cases} J = \frac{\partial f_P(q)}{\partial q} & \text{(jacobian geometryczny)} \\ J_A = \frac{\partial f_A(q)}{\partial q} & \text{(jacobian analityczny)} \end{cases} \quad (3.47)$$

Przekształcenie między prędkością kątową ω i prędkością $\dot{\Omega}$ uzyskamy na podstawie zależności (3.48), z której wynika wprost zależność między wierszami odpowiadającymi orientacji jacobianu analitycznego i geometrycznego (3.49). Macierz R_A wyliczana jest przez rzutowanie w układ bazowy pochodnych elementów Ω . Oczywiście postać R_A jest odmienna dla każdej reprezentacji orientacji. Na przykład dla zestawu kątów Eulera ZYZ (φ, ϑ, ψ) przybiera ona postać (3.50)².

$$\omega = R_A(\Omega) \dot{\Omega} \quad (3.48)$$

$$J = R_A(\Omega) J_A \quad (3.49)$$

$$R_A(\Omega) = \begin{bmatrix} 0 & -\sin(\varphi) & \cos(\varphi) \sin(\vartheta) \\ 0 & \cos(\varphi) & \sin(\varphi) \sin(\vartheta) \\ 1 & 0 & \cos(\vartheta) \end{bmatrix} \quad (3.50)$$

3.6 Struktura sterowania oparta na odwrotności jacobianu

W kolejnym kroku wyprowadzona zostanie struktura sterowania oparta na odwrotności Jacobianu. Możliwe są dwie wersje tej struktury. Pierwsza wynikająca bezpośrednio z zależności (3.51), polegająca na wyliczeniu w jednym kroku żądanej wartości zmiennych przegubowych, oraz druga z wbudowaną pętlą sprzężenia zwrotnego.

$$\dot{q} = J^{-1}(q) v \quad (3.51)$$

Celem odwrotnego zadania kinematyki jest wyznaczenie przebiegu w czasie wartości zmiennych przegubowych. Przy znajomości warunków początkowych

²Ze względu na specyfikę zaimplementowanych do tej pory funkcji składowych struktury ramowej MRROC++, w dalszej części pracy rozpatrywane będą jedynie algorytmy sterowania oparte na geometrycznej postaci jacobianu.

zadania (tzn. pozycji startowej $q(0)$) można je wyznaczyć przez scałkowanie wektora prędkości uogólnionych.

$$q(t) = \int_0^t \dot{q}(\xi) d\xi + q(0) \quad (3.52)$$

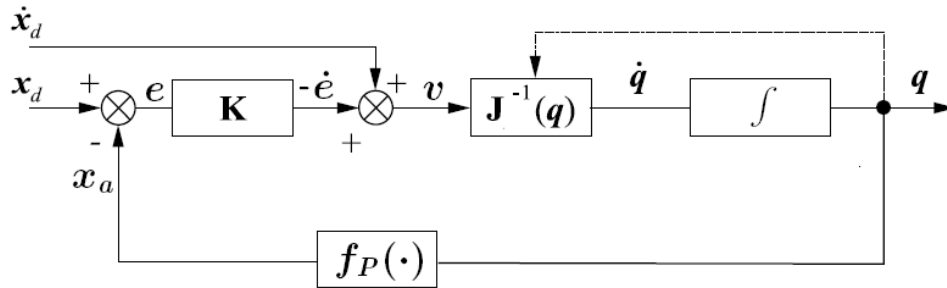
Po dyskretyzacji tego równania dla dostatecznie małych wartości Δt otrzymamy zależność (3.53), gdzie $t_{k+1} = t_k + \Delta t$.

$$q(t_{k+1}) = q(t_k) + \dot{q}(t_k) \Delta t \quad (3.53)$$

Po uwzględnieniu w ostatnim równaniu zależności (3.51) otrzymamy algorytm rozwiązania odwrotnego zadania kinematyki różniczkowej.

$$q(t_{k+1}) = q(t_k) + J^{-1}(q(t_k)) v(t_k) \Delta t \quad (3.54)$$

Algorytm rozwiązania oparty na tym równaniu jest mało dokładny ze względu na propagację błędu numerycznego i uchybów związanych z dyskretyzacją, których wartość wzrasta liniowo wraz z wzrostem wartości Δt . Aby poprawić jakość algorytmu do struktury rozwiązania dodawana jest pętla sprzężenia zwrotnego (rys. 3.2).



Rysunek 3.2: Struktura algorytmu iteracyjnego rozwiązania odwrotnego zadania kinetyki różniczkowej w oparciu o odwrotność jacobianu

W pętli tej poprzez x_a i x_d rozumiemy reprezentację aktualnej i zadanej pozycji końcówki robota. W momencie gdy w strukturze (3.54) zaszyty jest jacobian analityczny to wartości te reprezentowane są zależnością (3.43) ($x = g$), zaś w przypadku jacobianu geometrycznego przyjmują one postać macierzy przekształcenia jednorodnego ($x = T$).

Różnicę powyższych wartości $e = x_d - x_a$ nazywamy błędem, a jej pochodną wyznaczamy z zależności (3.55), gdzie x_a wyznaczone jest w pętli sprzężenia zwrotnego poprzez rozwiązanie prostego zadania kinematyki (3.44) lub (2.14). Dzięki temu wartość pochodnej błędu wyliczona może zostać z równania (3.56).

$$\dot{e} = \dot{x}_d - \dot{x}_a \quad (3.55)$$

$$\dot{e} = \dot{x}_d - J(q)\dot{q} \quad (3.56)$$

Jeżeli przyjęte zostanie, iż wartość \dot{e} jest liniową zależnością błędu e , czyli taką, że spełniona jest zależność (3.57), gdzie współczynniki dodatnio określonej macierzy K dobierane są doświadczalnie, to wartość sygnału \dot{q} wyznaczana jest z zależności (4.13).

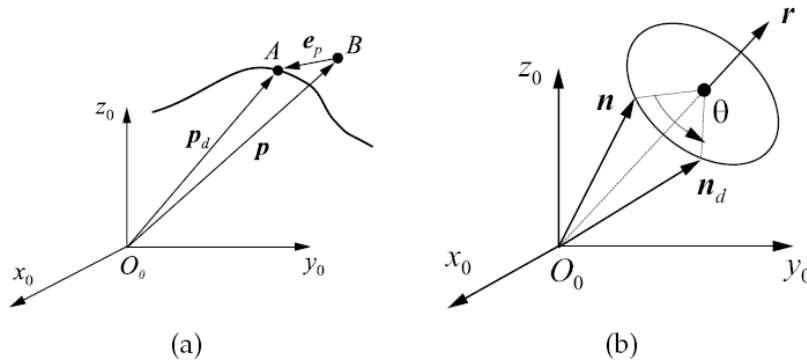
$$Ke + \dot{e} = 0 \quad (3.57)$$

$$\dot{q} = J^{-1}(q)(\dot{x}_d + Ke) \quad (3.58)$$

W przypadku rozważanego w pracy dyskretnego systemu sterowania manipulatorem IRp-6 mamy oczywiście do czynienia ze stałą wartością pozycji zadanej ($x_d = \text{const}$), a więc w dalszych rozdziałach przyjęta zostanie zerowa wartość pochodnej pozycji zadanej ($\dot{x}_d = 0$).

Pozostaje jeszcze problem wyznaczenia wartości uchybu pozycji efektora manipulatora, na podstawie zadanego i aktualnego układu reprezentowanego przez przekształcenia jednorodne: T_d (pozycja zadana) i T_a (pozycja aktualna). Wiersze wektora e_p , odpowiadające błędowi położenia wyznaczane są z różnicy wektorów położenia - ostatnich kolumn macierzy T_a i T_d .

Różnica między zadaną i aktualną macierzą rotacji wyznaczana jest z kolei na podstawie zależności (3.59), gdzie $R_{b,\theta}$ oznacza macierz rotacji odpowiadającą szukanemu uchybowi. Aby przedstawić ten obrót w niezależnej, dla każdej z osi układu, postaci macierz $R_{b,\theta}$ przedstawiona jest w reprezentacji oś-kąt (1.3.3). Wektory n , o , a oznaczają tu wersory składowe poszczególnych macierzy rotacji.



Rysunek 3.3: Obliczenie błędu położenia (a) i orientacji (b) [25]

$$R_{b,\theta} = R_d R_a^{-1} \quad (3.59)$$

$$R_{b,\theta} = \quad (3.60)$$

$$\begin{pmatrix} b_x^2(1 - \cos \theta) + \cos \theta & b_x b_y(1 - \cos \theta) - b_z \sin \theta & b_x b_z(1 - \cos \theta) + b_y \sin \theta \\ b_x b_y(1 - \cos \theta) + b_z \sin \theta & b_y^2(1 - \cos \theta) + \cos \theta & b_y b_z(1 - \cos \theta) - b_x \sin \theta \\ b_x b_z(1 - \cos \theta) - b_y \sin \theta & b_y b_z(1 - \cos \theta) + b_x \sin \theta & b_z^2(1 - \cos \theta) + \cos \theta \end{pmatrix}$$

$$R_d R_a^{-1} = \quad (3.61)$$

$$\begin{pmatrix} n_{dx}n_x + o_{dx}o_x + a_{dx}a_x & n_{dx}n_y + o_{dx}o_y + a_{dx}a_y & n_{dx}n_z + o_{dx}o_z + a_{dx}a_z \\ n_{dy}n_x + o_{dy}o_x + a_{dy}a_x & n_{dy}n_y + o_{dy}o_y + a_{dy}a_y & n_{dy}n_z + o_{dy}o_z + a_{dy}a_z \\ n_{dz}n_x + o_{dz}o_x + a_{dz}a_x & n_{dz}n_y + o_{dz}o_y + a_{dz}a_y & n_{dz}n_z + o_{dz}o_z + a_{dz}a_z \end{pmatrix}$$

Porównując elementy obu stron równania (3.59) otrzymane zostaną zależności na wartość poszukiwanego błędu pozycji. Poprzez odjęcie pierwszego elementu drugiej kolumny od drugiego elementu pierwszej kolumny otrzymany zostanie ostatni element wektora błędu orientacji:

$$b_x b_y(1 - \cos \theta) + b_z \sin \theta - b_x b_y(1 - \cos \theta) + b_z \sin \theta = \quad (3.62)$$

$$= n_{dy}n_x + o_{dy}o_x + a_{dy}a_x - n_{dx}n_y - o_{dx}o_y - a_{dx}a_y$$

$$2b_z \sin \theta = (n_{dy}n_x - n_{dx}n_y) + (o_{dy}o_x - o_{dx}o_y) + (a_{dy}a_x - a_{dx}a_y) \quad (3.63)$$

$$b_z \sin \theta = \frac{1}{2}((n_d \times n)_z + (o_d \times o)_z + (a_d \times a)_z) \quad (3.64)$$

Postępując analogicznie jak powyżej otrzymane zostaną kolejne zależności:

$$b_y \sin \theta = \frac{1}{2}((n_d \times n)_y + (o_d \times o)_y + (a_d \times a)_y) \quad (3.65)$$

$$b_x \sin \theta = \frac{1}{2}((n_d \times n)_x + (o_d \times o)_x + (a_d \times a)_x) \quad (3.66)$$

Tak więc końcowa zależność określająca wartość szukanego błędu pozycji przyjmuje następującą postać³:

$$\begin{cases} e_o = b \sin(\theta) = \frac{1}{2}(n_a \times n_d + o_a \times o_d + a_a \times a_d) \\ e_p = p_d - p_a \end{cases} \quad (3.67)$$

³Oczywiście postać ta jest prawdziwa jedynie dla małych wielkości kątów dla których $\sin x \cong x$.

3.7 Odwrotność jacobianu

Do wyznaczenia dualnego zadania kinematyki różniczkowej konieczne jest wyprowadzenie odwrotności jacobianu, co przy wymiarze macierzy 6×6 nie jest zadaniem banalnym. Poniżej przedstawione i porównane zostały dwa podejścia do tego problemu. Po pierwsze, poprzez algebraiczne przekształcenia, wyznaczona została analityczna postać równań określających J^{-1} , a następnie zaprezentowane zostało jedno w możliwych podejść numerycznych - metoda LU z eliminacją Gaussa [23]. Jeszcze inną, pominiętą w niniejszej pracy, stosowaną w tym przypadku metodą jest SMJM (Simple Modified Jacobian Matrix) w której jacobian jest przekształcany, poprzez zerowanie wybranych elementów, w macierz diagonalną [41]. Metoda ta jednak wnosi nie akceptowalnie duży błąd.

3.7.1 Postać analityczna

Z odwrotnością macierzy związany jest szereg pojęć algebraicznych. Poniżej wymienione zostały podstawowe definicje, których znajomość jest konieczna do wyznaczenia J^{-1} .

Definicja 1. Permutacją n -elementową nazywamy wzajemnie jednoznaczne przekształcenie zbioru n -elementowego na siebie.

$$\alpha(1\ 2\ \dots\ n) = (k_1\ k_2\ \dots\ k_n) \quad (3.68)$$

$$\alpha(i) = k_i \quad (3.69)$$

Definicja 2. Wyznacznik macierzy kwadratowej $A = [a_{ji}]_{n \times n}$ wyliczany jest z zależności (3.70), gdzie Π oznacza zbiór wszystkich możliwych permutacji macierzy, zaś k determinuje znak danej permutacji (liczba inwersji).

$$\det A = \sum_{\alpha \in \Pi} (-1)^k a_{1\alpha(1)} \cdot a_{2\alpha(2)} \cdot \dots \cdot a_{n\alpha(n)} \quad (3.70)$$

Definicja 3. Minorem stopnia k macierzy $A = [a_{ji}]_{m \times n}$, $k \leq \min(m, n)$ nazywamy wyznacznik macierzy kwadratowej stopnia k powstałej z macierzy A przez skreślenie $(m - k)$ wierszy, oraz $(n - k)$ kolumn.

Definicja 4. Dopełnieniem algebraicznym A_{ji} elementu a_{ji} macierzy $A = [a_{ji}]_{n \times n}$ nazywamy iloczyn minora stopnia $(n - 1)$ macierzy powstałej przez skreślenie j -tego wiersza i i -tej kolumny macierzy wejściowej oraz wartości $(-1)^{i+j}$.

Twierdzenie 1. (Laplace'a) Wyznacznik macierzy kwadratowej $A = [a_{ji}]_{n \times n}$ może być wyznaczony na podstawie jednej z zależności rekurencyjnych (3.71).

$$\begin{cases} \det A = \sum_{j=1}^n a_{1j} A_{1j} \\ \det A = \sum_{j=1}^n a_{j1} A_{j1} \end{cases} \quad (3.71)$$

Definicja 5. Macierz transponowana jest macierzą powstałą przez wzajemną zamianę wierszy i kolumny macierzy wejściowej:

$$A = [a_{ji}]_{m \times n} \mapsto A^T = [a_{ij}]_{n \times m} \quad (3.72)$$

Definicja 6. Macierzą dołączoną macierzy $A = [a_{ji}]_{n \times n}$ nazywamy transpozycję macierzy dopełnień algebraicznych $A^D = [A_{ji}]_{n \times n}^T$. Dla dowolnych macierzy kwadratowych zachodzi $AA^D = A^D A = \det[A]I$

Definicja 7. Macierz odwrotna do zadanej macierzy kwadratowej A jest macierzą A^{-1} dla której prawdziwa jest równość $AA^{-1} = A^{-1}A = I$. Macierz odwrotna istnieje jedynie dla $\det[A] \neq 0$ i wyliczana jest z wzoru (3.73).

$$A^{-1} = \frac{1}{\det[A]} A^D \quad (3.73)$$

Wyznacznik jacobianu manipulatora IRp-6 wyznaczony został przy pomocy twierdzenia Laplace'a. Operacja ta, po uwzględnieniu zerowych elementów macierzy J , wymagała wyznaczenia 10 minorów 3 stopnia. Ostateczny, po znaczącym uproszczeniu, wzór definiujący wyznacznik jacobianu przybiera postać:

$$\det J = a_2 a_3 s_5 (s_2 c_3 - c_2 s_3) (c_4 d_5 + a_3 c_3 + a_2 c_2) \quad (3.74)$$

Poniżej przedstawiono analityczną postać macierz J^{-1} .

$$J^{-1} = \begin{pmatrix} J_{11} & J_{12} & \cdots & J_{16} \\ J_{21} & J_{22} & \cdots & J_{26} \\ \vdots & \vdots & \ddots & \vdots \\ J_{61} & J_{62} & \cdots & J_{66} \end{pmatrix} \quad (3.75)$$

$$J_{11} = 0 \quad (3.76)$$

$$J_{21} = \frac{(s_1 c_4 s_3 s_5 - c_3 s_1 s_4 s_5 + s_3 c_4 s_4 c_1 c_5 - c_3 c_1 c_5 + c_3 c_1 c_5 c_4^2) d_5}{(s_3 c_2 - s_2 c_3) a_2 s_5} \quad (3.77)$$

$$J_{31} = \frac{(c_2 s_1 s_4 s_5 + c_2 c_1 c_5 - c_2 c_1 c_5 c_4^2 - s_1 c_4 s_5 s_2 - s_2 c_4 s_4 c_1 c_5) d_5}{(s_3 c_2 - s_2 c_3) a_3 s_5} \quad (3.78)$$

$$J_{41} = -\frac{s_1 s_5 + s_4 c_1 c_5}{s_5} \quad (3.79)$$

$$J_{51} = c_1 c_4 \quad (3.80)$$

$$J_{61} = \frac{s_4 c_1}{s_5} \quad (3.81)$$

$$J_{12} = 0 \quad (3.82)$$

$$J_{22} = \frac{(s_1 s_3 c_4 s_4 c_5 - c_3 s_1 c_5 + c_3 s_1 c_5 c_4^2 + c_3 c_1 s_4 s_5 - c_4 c_1 s_3 s_5) d_5}{(s_3 c_2 - s_2 c_3) a_2 s_5} \quad (3.83)$$

$$J_{32} = \frac{(c_2 s_1 c_5 - c_2 s_1 c_5 c_4^2 - c_2 c_1 s_4 s_5 - s_1 s_2 c_4 s_4 c_5 + c_4 c_1 s_5 s_2) d_5}{(s_3 c_2 - s_2 c_3) a_3 s_5} \quad (3.84)$$

$$J_{42} = \frac{c_1 s_5 - s_4 s_1 c_5}{s_5} \quad (3.85)$$

$$J_{52} = s_1 c_4 \quad (3.86)$$

$$J_{62} = \frac{s_4 s_1}{s_5} \quad (3.87)$$

$$J_{13} = 0 \quad (3.88)$$

$$J_{23} = \frac{(c_4 s_3 - s_4 c_3) c_4 d_5 c_5}{(s_3 c_2 - s_2 c_3) a_2 s_5} \quad (3.89)$$

$$J_{33} = \frac{(s_4 c_2 - c_4 s_2) c_4 d_5 c_5}{(s_3 c_2 - s_2 c_3) a_3 s_5} \quad (3.90)$$

$$J_{43} = -\frac{c_4 c_5}{s_5} \quad (3.91)$$

$$J_{53} = -s_4 \quad (3.92)$$

$$J_{63} = \frac{c_4}{s_5} \quad (3.93)$$

$$J_{14} = -\frac{s_1}{c_4 d_5 + c_3 a_3 + c_2 a_2} \quad (3.94)$$

$$J_{24} = \frac{c_4 d_5 (s_1 s_3 c_4 c_5 - s_1 c_3 s_4 c_5 + c_1 c_3 s_5) + c_1 c_3 s_5 (c_3 a_3 + c_2 a_2)}{(c_4 d_5 + c_3 a_3 + c_2 a_2) (s_3 c_2 - s_2 c_3) a_2 s_5} \quad (3.95)$$

$$J_{34} = -\frac{c_4 d_5 (s_1 s_2 c_4 c_5 - s_1 c_2 s_4 c_5 + c_1 c_2 s_5) + c_1 c_2 s_5 (c_3 a_3 + c_2 a_2)}{(c_4 d_5 + c_3 a_3 + c_2 a_2) (s_3 c_2 - s_2 c_3) a_3 s_5} \quad (3.96)$$

$$J_{44} = -\frac{s_1 c_4 c_5}{(c_4 d_5 + c_3 a_3 + c_2 a_2) s_5} \quad (3.97)$$

$$J_{54} = -\frac{s_4 s_1}{c_4 d_5 + c_3 a_3 + c_2 a_2} \quad (3.98)$$

$$J_{64} = \frac{s_1 c_4}{(c_4 d_5 + c_3 a_3 + c_2 a_2) s_5} \quad (3.99)$$

$$J_{15} = \frac{c_1}{c_4 d_5 + c_3 a_3 + c_2 a_2} \quad (3.100)$$

$$J_{25} = \frac{s_1 s_5 c_3 (c_4 d_5 + c_3 a_3 + c_2 a_2) - c_1 c_4 c_5 d_5 (s_3 c_4 + c_3 s_4)}{(c_4 d_5 + c_3 a_3 + c_2 a_2) (s_3 c_2 - s_2 c_3) a_2 s_5} \quad (3.101)$$

$$J_{35} = -\frac{s_1 s_5 c_2 (c_4 d_5 + c_3 a_3 + c_2 a_2) - c_1 c_4 c_5 d_5 (s_2 c_4 + c_2 s_4)}{(c_4 d_5 + c_3 a_3 + c_2 a_2) (s_3 c_2 - s_2 c_3) a_3 s_5} \quad (3.102)$$

$$J_{45} = \frac{c_1 c_4 c_5}{(c_4 d_5 + c_3 a_3 + c_2 a_2) s_5} \quad (3.103)$$

$$J_{55} = \frac{s_4 c_1}{c_4 d_5 + c_3 a_3 + c_2 a_2} \quad (3.104)$$

$$J_{65} = -\frac{c_1 c_4}{(c_4 d_5 + c_3 a_3 + c_2 a_2) s_5} \quad (3.105)$$

$$J_{16} = 0 \quad (3.106)$$

$$J_{26} = -\frac{s_3}{(s_3 c_2 - s_2 c_3) a_2} \quad (3.107)$$

$$J_{36} = \frac{s_2}{(s_3 c_2 - s_2 c_3) a_3} \quad (3.108)$$

$$J_{46} = 0 \quad (3.109)$$

$$J_{56} = 0 \quad (3.110)$$

$$J_{66} = 0 \quad (3.111)$$

3.7.2 Metoda LU

Zadanie odwrotne kinematyki różniczkowej sprowadza się do rozwiązania równania postaci (3.6). Stosując metodę analityczną w pierw wyznaczana jest wartość J^{-1} , a następnie wyliczany iloczyn odwrotności Jakobianu i wektora prędkości. Zaprezentowana poniżej metoda numeryczna umożliwia rozwiązanie tego równania, bez potrzeby wyliczenia J^{-1} , co w znacznym stopniu zmniejsza złożoność obliczeniową zadania i wielkość powstałego błędu numerycznego. Największą jednak zaletą podejścia numerycznego jest możliwość stworzenia, podczas implementacji algorytmów, uniwersalnej metody obsługującej alternatywne do przedstawionej postaci jacobianu, co nie jest możliwe w przypadku wprowadzonych „na sztywno” równań.

W metodzie LU macierz A , występująca w równaniu (3.112)⁴, rozkładana jest na iloczyn macierzy dolnej i górnej trójkątnej (3.113), dzięki czemu rozwiązanie równania, sprowadza się do układu dwóch równań zawierających macierze trójkątne (3.114). Dla ustalenia jednoznaczności rozkładu diagonalą macierzy L przyjmuje wartości 1. Samo rozwiązanie równania zawierającego macierz trójkątną jest zadaniem trywialnym.

$$y = Ax = LUx \quad (3.112)$$

$$L = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ l_{n1} & l_{n2} & \cdots & 1 \end{pmatrix} \quad U = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{pmatrix} \quad (3.113)$$

$$\begin{cases} y = Lz \\ z = Ux \end{cases} \quad (3.114)$$

Rozkładu macierzy można dokonać na dwa sposoby: metodą eliminacji Gaussa (rys. 3.4) oraz metodą Doolittle'a.

Z algorytmu tego widać, iż metoda nie osiągnie rozwiązania w momencie gdy $A_{kk} = 0$. Aby temu zapobiec stosowana jest zmodyfikowana metoda eliminacji Gaussa z częściowym lub pełnym wyborem elementu podstawowego. Ideą tej metody jest przestawienie wierszy macierzy wejściowej, tak aby elementem podstawowym stała się największa co do modułu wartość. W przypadku wyboru częściowego wybierany jest maksymalny element danej kolumny, natomiast wyboru pełnego - całej macierzy. Przestawienie kolumn macierzy A musi oczywiście pociągać za sobą odpowiednią permutację wektora y .

Tak więc ostatecznym rozwiązaniem zadania (3.112) będzie równanie (3.115), gdzie P jest macierzą permutacji zawierającą informację dotyczącą indeksów przestawionych elementów macierzy.

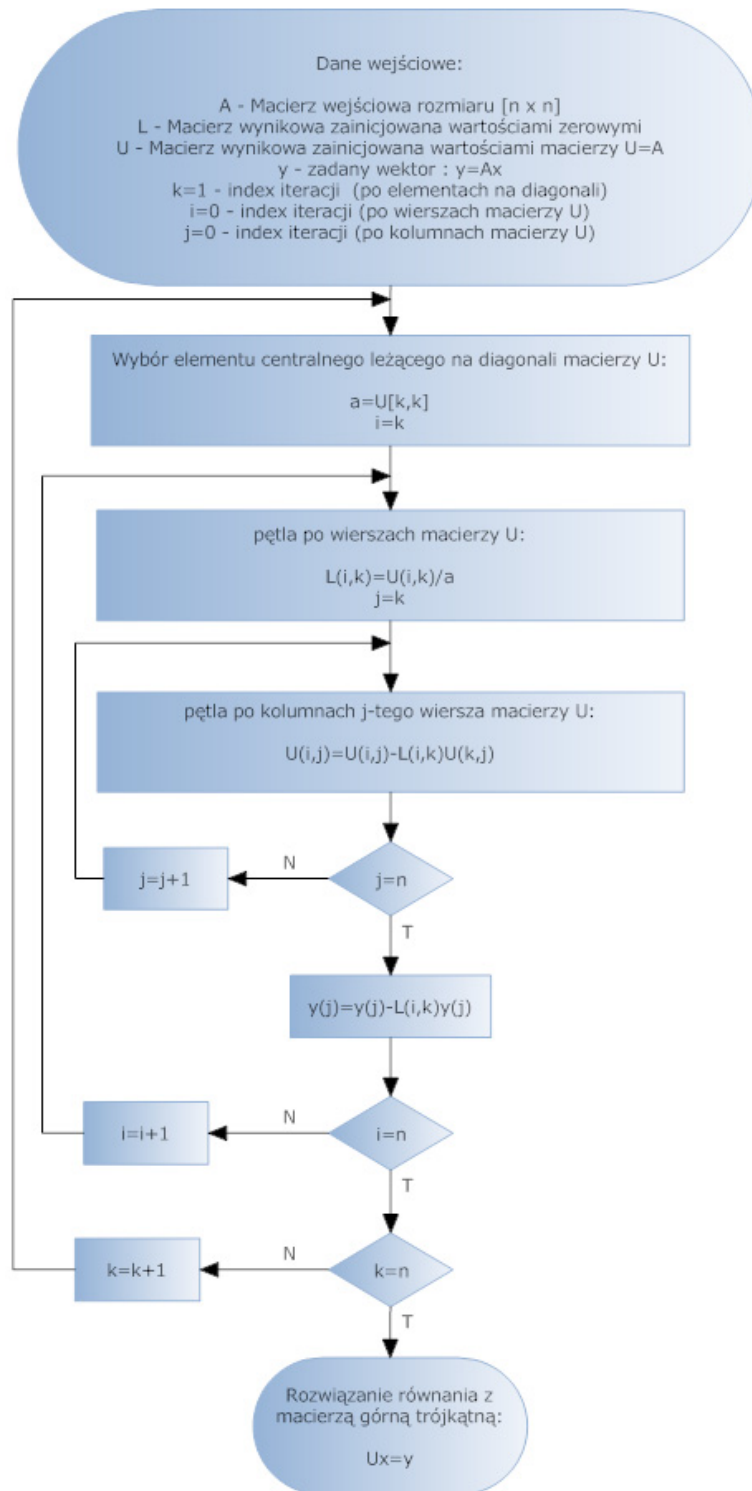
$$Ax = PLUx = y \quad (3.115)$$

Reasumując, rozwiązaniem odwrotnego zadania kinematyki różniczkowej (3.6), przy zastosowaniu metody LU, będzie układ równań postaci (3.117).

$$J = PLU \quad (3.116)$$

$$\begin{pmatrix} \omega \\ v \end{pmatrix} = (J) \dot{q} = (PLU) \dot{q} \quad (3.117)$$

⁴Rozwiązaniem tego równania jest wyznaczenie wartości x , przy zadanych wartościach wektora y i macierzy A



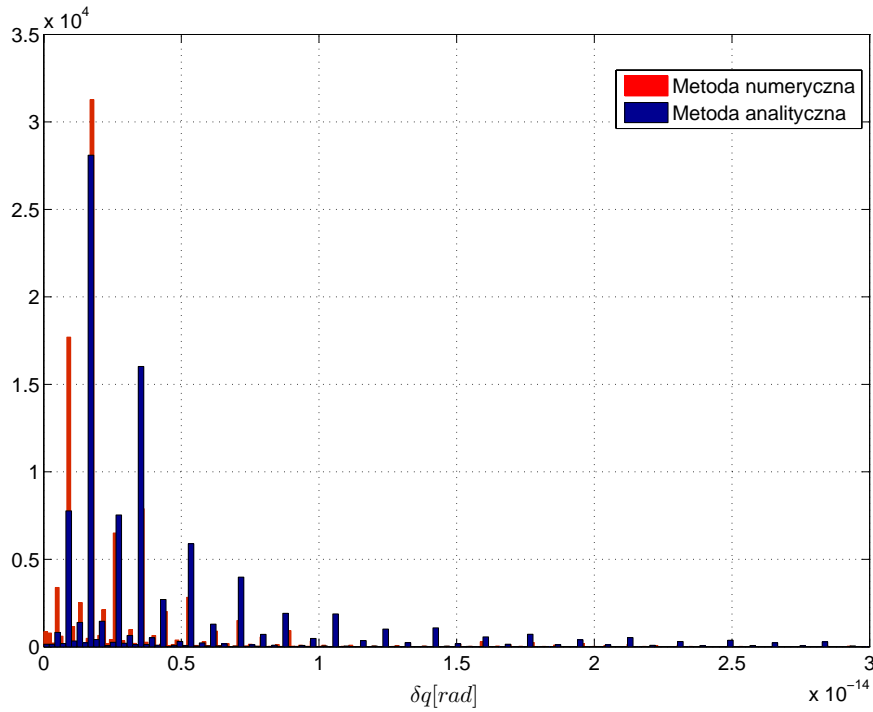
Rysunek 3.4: Schemat algorytmu eliminacji Gaussa

3.7.3 Porównanie metody analitycznej i numerycznej

W celu porównania metody analitycznej i numerycznej przeprowadzone zostały testy w środowisku Matlab. Dla 100000, z pominięciem osobliwych, losowych konfiguracji wyliczone zostało rozwiązanie zadania (3.6). Wyniki zostały porównane pod względem dokładności⁵, oraz czasu wykonania i zaprezentowane w postaci histogramów na poniższych wykresach. Widać iż metoda numeryczna daje dwa razy mniejsze odchylenia wyników, przy niewielkim wzroście zużycia czasu procesora.

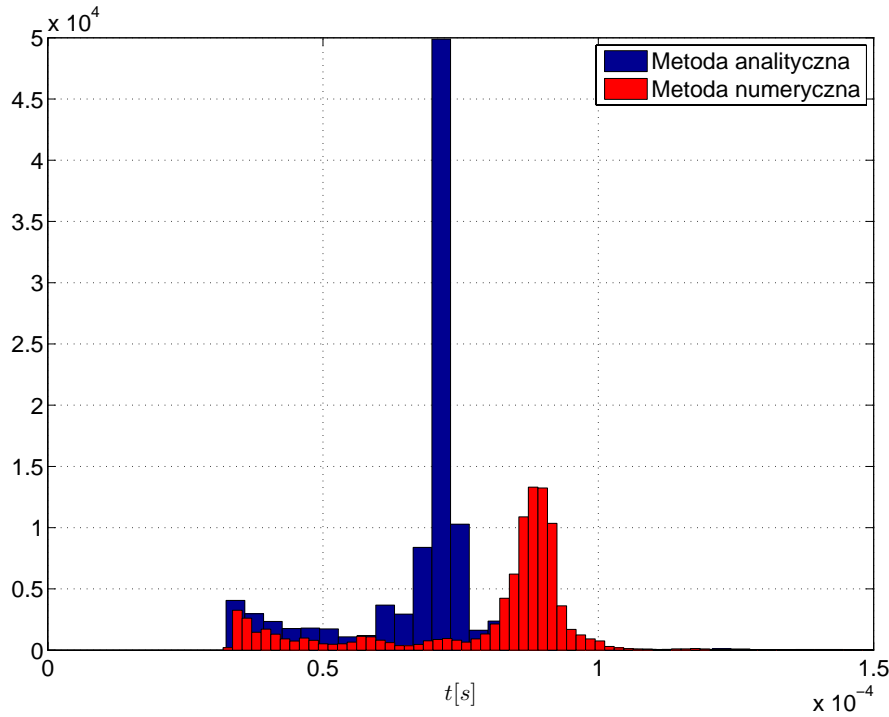
Zaprezentowane wyniki testów przeprowadzone zostały na domowej stacji roboczej i mają jedynie charakter porównawczy obu metod.

wartości średnie	błąd obliczeń	czas obliczeń
metoda analityczna	7.3738e-015	7.4138e-005
metoda numeryczna	4.1693e-015	8.2320e-005



Rysunek 3.5: Histogram porównawczy metody analitycznej i numerycznej pod względem dokonanego uchybu

⁵Błąd obliczeń wyznaczono jako różnicę zadanej wartości Δq_1 i wyliczonej $\Delta q_2 = \text{inv}(J) * V = \text{inv}(J) * J * \Delta q_1$, gdzie wartość odwrotności jacobianu wyliczono metodą analityczną lub metodą eliminacji Gaussa.



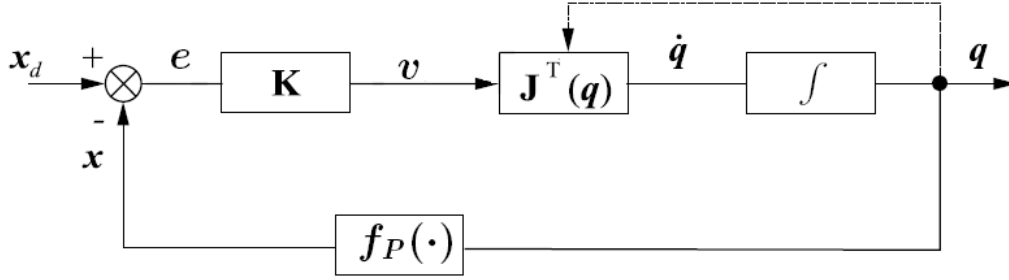
Rysunek 3.6: Histogram porównawczy metody analitycznej i numerycznej pod względem czasu zajętości procesora

3.8 Transpozycja jacobianu

Alternatywna, do podanej na str. 63 struktura rozwiązania odwrotnego zadania kinematyki opiera się na transpozycji, zamiast odwrotności jacobianu [15, 34, 37, 31], a wartość pochodnej zmiennych przegubowych wyliczana jest na podstawie zależności (3.118). Opisana struktura sterowania przybiera postać (rys. 3.7).

$$\dot{q} = J^T(q) K e. \quad (3.118)$$

Stabilności przedstawionego układu można dowieść bezpośrednią metodą Lapunowa. Metoda ta jest potężnym narzędziem w analizie układów nieliniowych i stanowi uogólnienie pojęcia energii mechanicznej. Idea metody Lapunowa sprowadza się do stwierdzenia, iż ruch dowolnego układu jest stabilny, o ile całkowita energia mechaniczna zmniejsza się przez cały czas jego działania. Używając bezpośredniej metody Lapunowa w analizie stabilności układów nieliniowych, tworzymy skalar energii nazywany funkcją Lapunowa - V i sprawdzamy, czy zmniejsza się ona wraz z upływem czasu. Twierdzenie Lapunowa, na którym opiera się owa metoda formułujemy w następujący sposób:



Rysunek 3.7: Struktura sterowania oparta na transpozycji jakobianu

Twierdzenie Układ nieliniowy opisany równaniem $\dot{x} = f(x)$ jest stabilny asymptotycznie w obszarze Ω , zawierającym początek układu współrzędnych (punkt równowagi $x_r = 0$), jeżeli można dobrać taką funkcję Lapunowa $V(x)$ dodatnio określoną w obszarze Ω , której pochodna względem czasu $\dot{V}(x)$ wzdłuż trajektorii stanu jest funkcją ujemnie określoną w tym obszarze Ω :

$$\begin{cases} V(x) > 0 & \forall x \neq 0 \\ V(0) = 0 \end{cases} \quad (3.119)$$

$$\begin{cases} \dot{V}(x) < 0 & \forall x \neq 0 \\ \dot{V}(0) = 0 \end{cases} \quad (3.120)$$

Dla badanego układu (rys. 3.7), zawierającego jakobian transponowany, skalarną funkcję Lapunowa zdefiniować możemy jako ważoną funkcję błędu (3.121). W wzorze tym K oznacza dodatnio określoną, symetryczną macierz kwadratową, dzięki czemu spełniony jest warunek dodatniej określoności (3.119).

$$V(e) = \frac{1}{2} e^T K e \quad (3.121)$$

Pochodna powyższej funkcji Lapunowa przyjmuje postacie (3.122-3.124).

$$\dot{V}(e) = e^T K \dot{e} \quad (3.122)$$

$$\dot{V}(e) = e^T K \dot{x}_d - e^T K \dot{x} \quad (3.123)$$

$$\dot{V}(e) = e^T K \dot{x}_d - e^T K J_A(q) \dot{q} \quad (3.124)$$

Jeżeli w ostatnim z tych równań w miejsce \dot{q} podstawić zależność (3.118) to ostateczny wzór określający pochodną funkcji Lapunowa przyjmie postać (3.125).

$$\dot{V}(e) = e^T K \dot{x}_d - e^T K J_A(q) J_A^T(q) K e \quad (3.125)$$

W przypadku gdy $\dot{x}_d = 0$, tzn. sygnał x_d jest stały, a macierz J_A jest pełnego rzędu to, $\dot{V}(e)$ spełnia warunki określone w twierdzeniu Lapunowa i cały układ jest asymptotycznie stabilny (trajektorie dążą jednostajnie do $e = 0$).

W momencie gdy $\dot{x}_d \neq 0$ (tzn. mamy do czynienia z układem ciągłym) układ traci asymptotyczną stabilność i trudno de facto przewidzieć jego zachowanie. Pierwszy człon równania (3.125) może przybrać dowolną wartość, przez co cała funkcja może przyjąć dodatnie wartości. Warunki pracy układu poprawić możemy manipulując wartościami macierzy K^6 .

3.9 Konfiguracje osobliwe manipulatora

Z kinematyką manipulatorów związany jest nierozłącznie problem osobliwości. Wynika on z tego, że przy pewnych konfiguracjach złącz manipulatora może zajść sytuacja, iż nie jest możliwe wymuszenie dowolnego ruchu końcówki, robot traci część stopni swobody lub małym zmianom prędkości końcówki odpowiadają duże zmiany prędkości w przestrzeni złącz. Konfiguracje te nazywamy osobliwymi i występują one w momencie gdy macierz jakobianu traci rząd, tzn. jej wyznacznik wynosi 0. W większości przypadków, dla konfiguracji osobliwych, rozwiązanie odwrotnego zadania kinematyki pozycyjnej ma nieskończenie wiele rozwiązań, a przez niemożność wyznaczenia odwrotności jakobianu struktura oparta na równaniu (3.6) jest niemożliwa do zastosowania.

Równanie określające wyznacznik manipulatora IRp-6 o sześciu stopniach swobody zostało wyznaczone w poprzedniej części pracy i ma następującą postać (3.126).

$$\det J = s_5 a_2 a_3 (s_2 c_3 - c_2 s_3) (c_4 d_5 + a_3 c_3 + a_2 c_2) \quad (3.126)$$

Z zależności tej wynika, iż możliwe są trzy konfiguracje osobliwe robota: jedna związana z kciścią i dwie z ramieniem. Pierwsza konfiguracja osobliwa występuje w momencie gdy $s_5 = 0$, a więc $\theta_5 = k\pi$ dla $k \in \mathbb{Z}$, przez co osie czwartego i szóstego przegubu są równoległe. Dla tego przypadku nie jest możliwe wyznaczenie jednoznacznego rozwiązania odwrotnego zadania kinematyki dla obrotu wyznaczonego przez osie tych przegubów. Problem ten został przedstawiony w rozdziale dotyczącym rozwiązania odwrotnego zadania kinematyki pozycyjnej.

Kolejne dwie konfiguracje osobliwe związane są z ułożeniem zmiennych przegubowych ramienia: $c_4 d_5 + a_3 c_3 + a_2 c_2 = 0$ lub $\tan(\theta_2) = \tan(\theta_3)$. Pierwsza z tych sytuacji nastąpi gdy drugi, trzeci i czwarty przegub przyjmie wartość:

⁶Pierwszy człon równania (3.125) jest liniową funkcją błędu, a drugi kwadratową, tak więc dla małych wartości e pierwsza część równania ma decydujący wpływ na znak rozwiązania. W tej sytuacji poprzez powiększenie normy macierzy K zwiększamy wpływ składnika kwadratowego, tak by wartość funkcji przyjęła ujemne wartości i układ odzyskał stabilność.

$\forall_{i \in (2,3,4)} \theta_i = k\pi + \frac{\pi}{2}$, tzn. całe ramie ustawione będzie prostopadle do płaszczyzny wyznaczonej przez osie Ox i Oy układu bazowego, a rzut kiści na podłoże wyniesie zero. Kolejna sytuacja wystąpi gdy różnica wartości obrotu drugiego i trzeciego przegubu jest wielokrotnością kąta półpełnego: $\theta_2 = \theta_3 + k\pi$, gdzie $k \in \mathbb{Z}$, a co za tym idzie kierunki wyznaczone przez ramiona (a_2) i (a_3) leżą na tej samej prostej. Jeżeli w powyższych rozważaniach uwzględnione zostaną ograniczenia ruchu poszczególnych zmiennych wewnętrznych⁷, to okaże się iż dla manipulatora IRp-6 możliwa jest tylko pierwsza z przedstawionych powyżej konfiguracji osobliwych.

Jak zostało poprzednio zaznaczone dla konfiguracji osobliwych struktura sterowania oparta na odwrotności jakobianu jest niemożliwa do zastosowania. Co więcej, nawet gdy manipulator zbliża się do tych konfiguracji, to z powodu złego uwarunkowania macierzy J popełniany przez metodę błąd jest nie akceptowalnie duży⁸. W celu rozwiązania problemu osobliwości jakobianu konieczne jest zastosowanie jednej z zaawansowanych metod matematycznych. Zdecydowana ich większość opiera się na wybraniu z jakobianu wierszy liniowo niezależnych i rozwiązaniu jedynie odpowiadającej im części układu równań. Do metod tych należy technika SVD [24, 43] (polegająca na rozkładzie według wartości własnych macierzy J) oraz DLS (odwrotność jakobianu przy tłumionych najmniejszych kwadratach) [6, 12, 27]. Inną metodą rozwiązania jest wyliczenie pseudo inwersji Jakobianu[29].

⁷ $\theta_2 \in \langle -130^\circ, -50^\circ \rangle$, $\theta_3 \in \langle -25^\circ, 40^\circ \rangle$, $\theta_4 \in \langle -90^\circ, 90^\circ \rangle$

⁸Doświadczalnie zostało wyznaczone, iż metoda zawodzi dla $\theta_5 = k\pi \pm 1^\circ$.

Rozdział 4

Wyniki symulacji w środowisku Matlab

4.1 Uchyb dla jednego kroku prostego zadania kinematyki różniczkowej

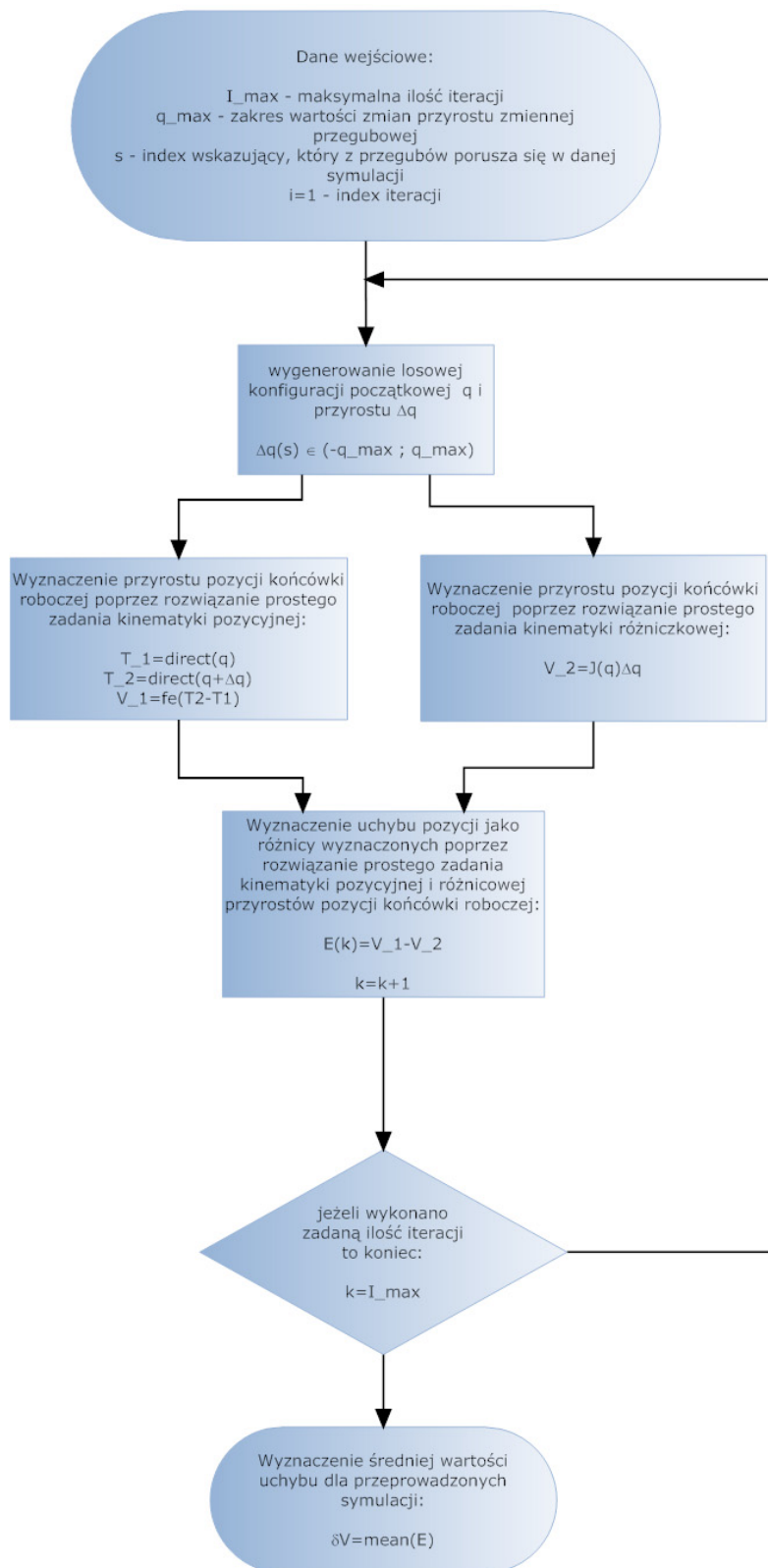
W celu zweryfikowania poprawności wyznaczonego jakobianu przeprowadzony został szereg testów. Po pierwsze, dla zadanej konfiguracji początkowej q i przyrostu współrzędnych wewnętrznych z zadanego zakresu $\Delta q \in (-q_{max}, q_{max})$ rozwiązano proste zadanie kinematyki pozycyjnej i różniczkowej, wyznaczając w ten sposób przyrost pozycji końcówki roboczej. W przypadku kinematyki pozycyjnej dla początkowej (q) i docelowej ($q + \Delta q$) konfiguracji wewnętrznej robota wyznaczono macierz przekształcenia jednorodnego (T_1, T_2), a następnie korzystając z zależności (3.67) wyznaczono różnicę pomiędzy tymi macierzami (V_1).

$$V_1 = f_e(T_2 - T_1) = \begin{cases} \frac{1}{2}(n_2 \times n_1 + o_2 \times o_1 + a_2 \times a_1) \\ p_2 - p_1 \end{cases} \quad (4.1)$$

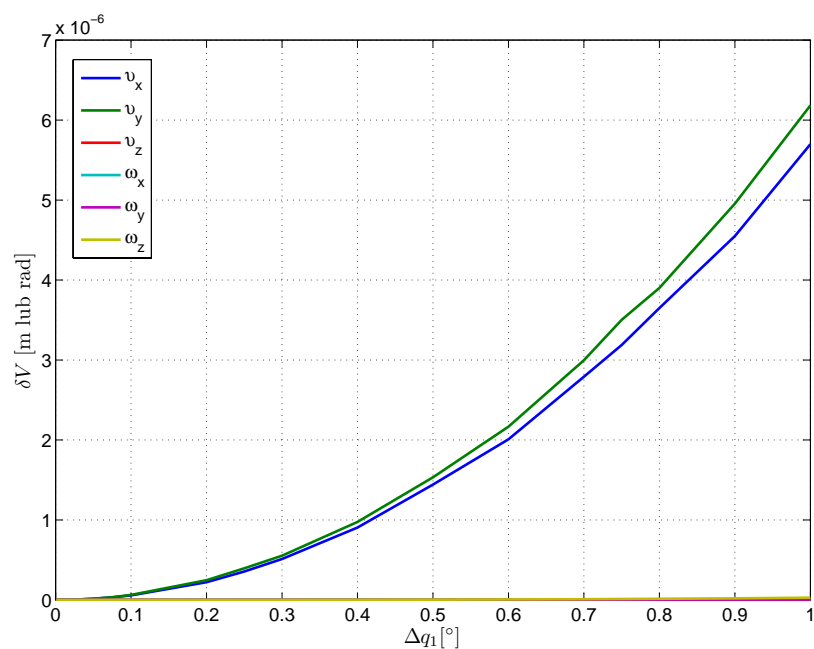
Z kolei dla prostego zadania kinematyki różniczkowej na podstawie aktualnej wartości zmiennych przegubowych wyznaczono macierz jakobianową ($J(q)$), a z wymnożenia jej przez przyrost współrzędnych wewnętrznych otrzymano przyrost pozycji końcówki roboczej ($J(q) \cdot \Delta q = V_2$). Uchyb metody wyznaczono jako różnicę rzeczywistego (V_1) i aproksymowanego (V_2) przyrostu ($E = V_1 - V_2$).

Powyższa procedura powtórzona została dla $I_{max} = 50\,000$ konfiguracji początkowych, a końcowy wynik otrzymano poprzez uśrednienie wyników pośrednich ($\delta V = mean(E)$). Otrzymane w ten sposób wartości zostały przedstawione na poniższych wykresach, w funkcji rosnących wartości q_{max} .

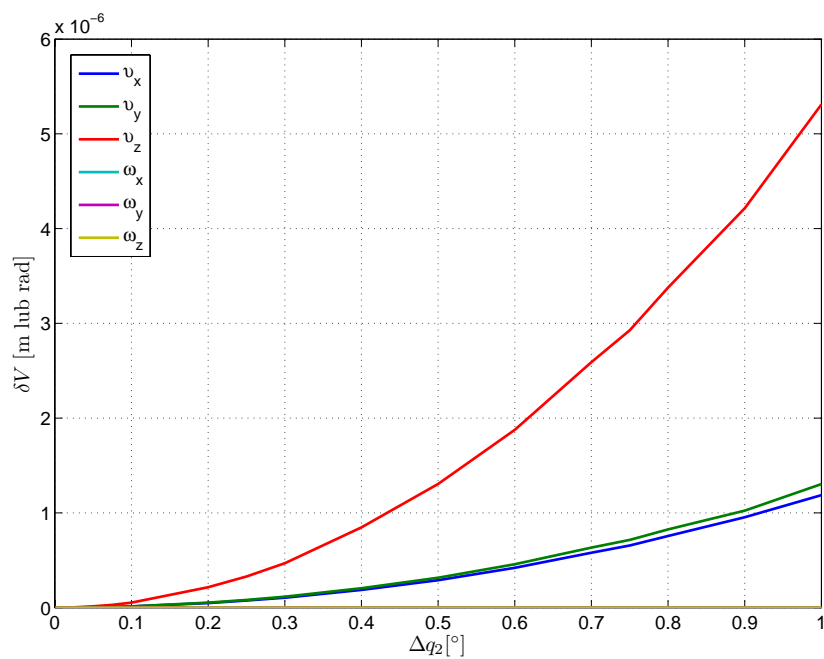
Zaprezentowane symulacje przeprowadzone zostały z osobna dla przyrostów każdej ze zmiennych przegubowych, dzięki czemu można zaobserwować wpływ każdej z nich na poszczególne elementy wektora prędkości V .



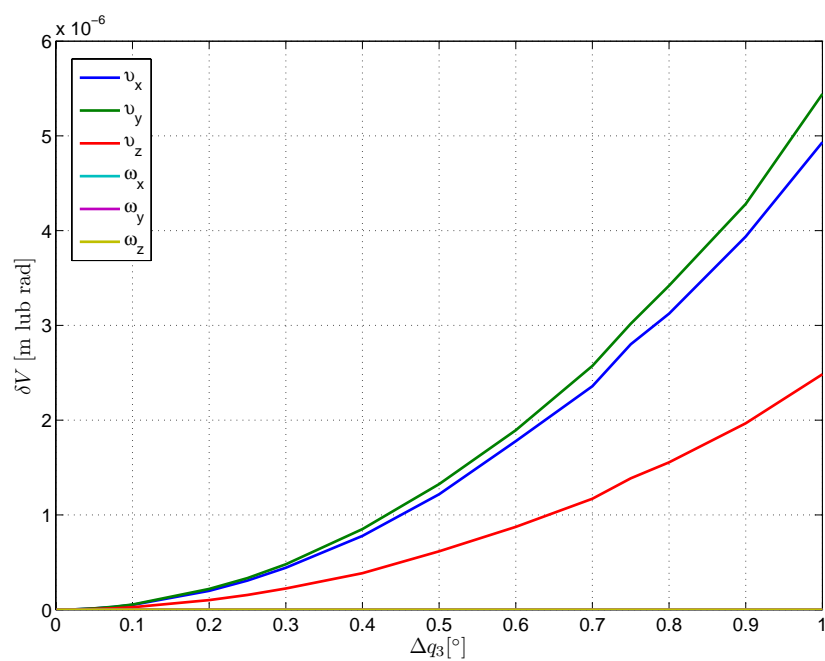
Rysunek 4.1: Schemat algorytmu symulacji prostego zadania kinematyki pozycyjnej i różniczkowej



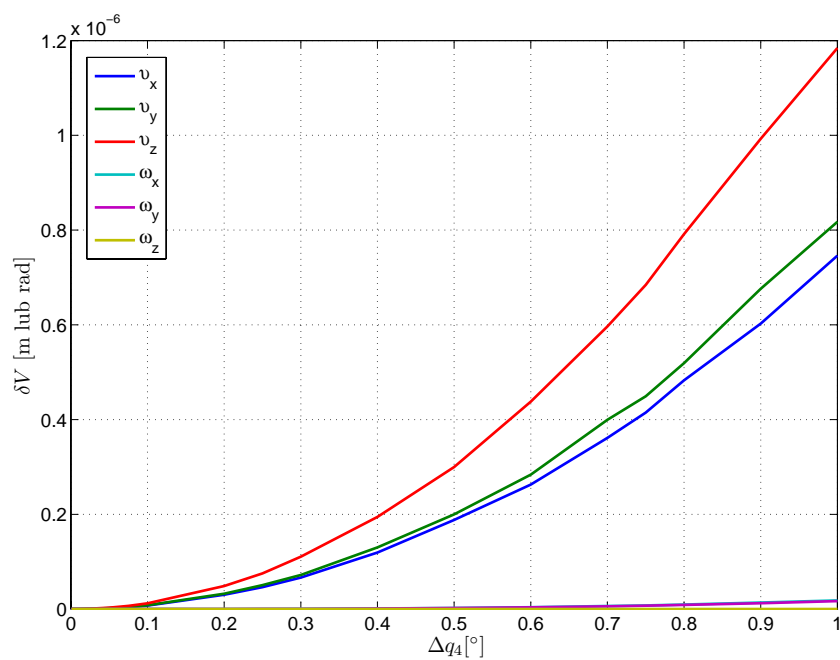
Rysunek 4.2: Uchyb wyznaczonego przesunięcia współrzędnych dla zadanego Δq_1



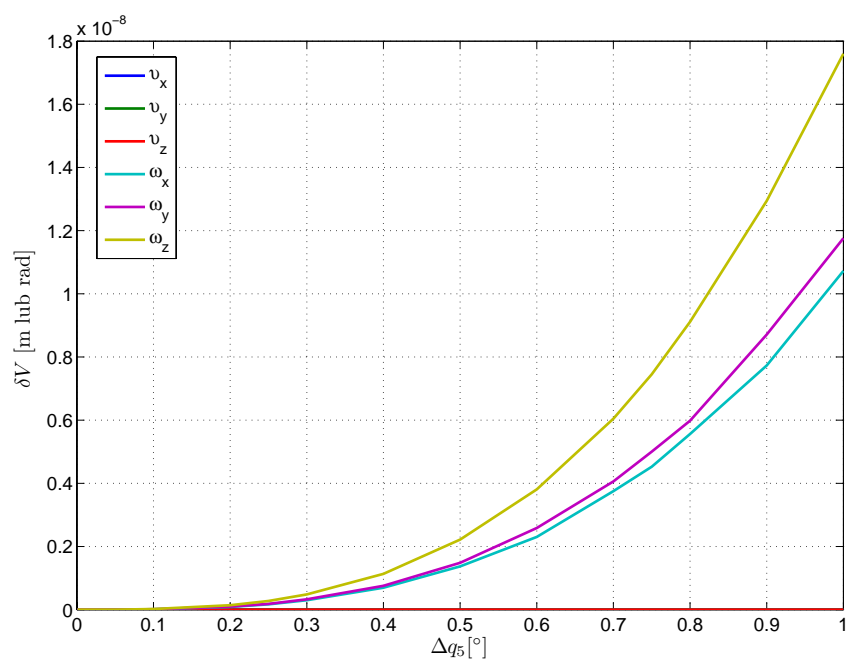
Rysunek 4.3: Uchyb wyznaczonego przesunięcia współrzędnych dla zadanego Δq_2



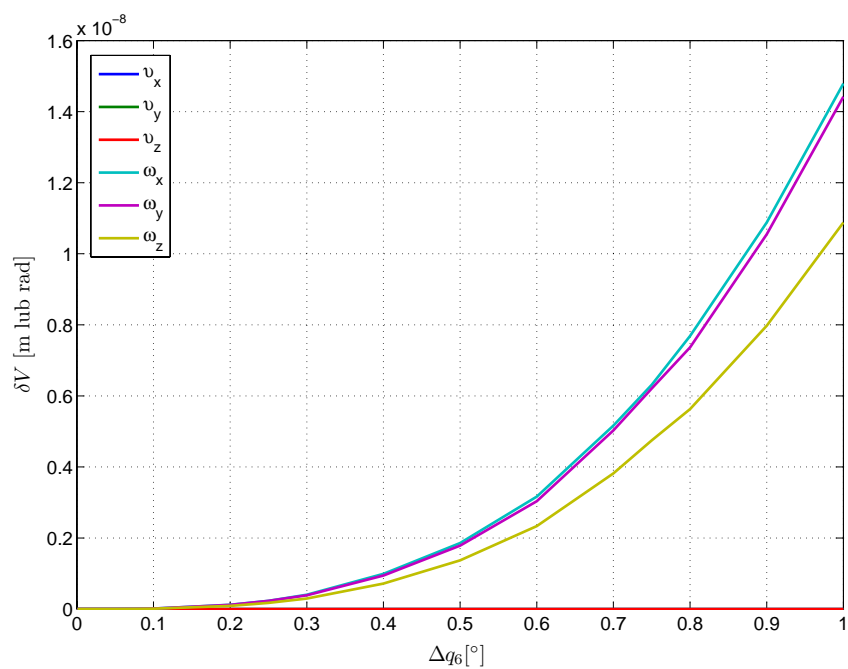
Rysunek 4.4: Uchyb wyznaczonego przesunięcia współrzędnych dla zadanego Δq_3



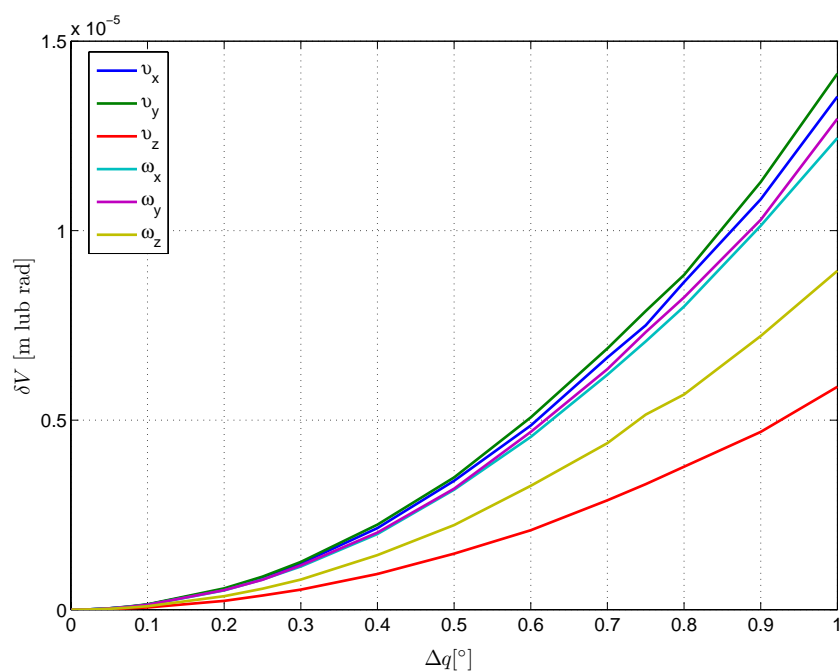
Rysunek 4.5: Uchyb wyznaczonego przesunięcia współrzędnych dla zadanego Δq_4



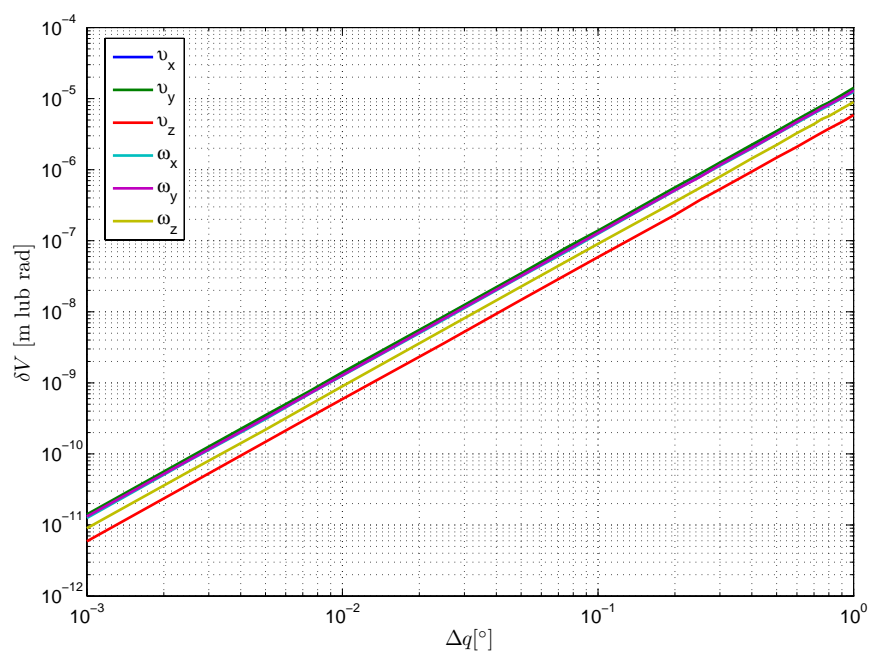
Rysunek 4.6: Uchyb wyznaczonego przesunięcia współrzędnych dla zadanego Δq_5



Rysunek 4.7: Uchyb wyznaczonego przesunięcia współrzędnych dla zadanego Δq_6



Rysunek 4.8: Uchyb wyznaczonego przesunięcia współrzędnych dla zadanego wektora Δq



Rysunek 4.9: Uchyb wyznaczonego przesunięcia współrzędnych dla zadanego wektora Δq przedstawiony w skali logarytmicznej

Z powyższych wykresów widać, iż wartość uchybu metody wzrasta wykładniczo wraz z wartością zakresu przyrostu zmiennych przegubowych q_{max} , a dla krańcowego z badanych przyrostów $\Delta q \sim 1^\circ$ można spodziewać się uchybu rzędu $1.4 * 10^{-5}$.

Jak zostało pokazane poprzez powyższe symulacje metoda oparta na jacobianie, w odróżnieniu od metody pozycyjnej, nie jest metodą dokładną. Wielkość błędu jednakże, zwłaszcza po uwzględnieniu pętli sprzężenia zwrotnego, jest akceptowalna.

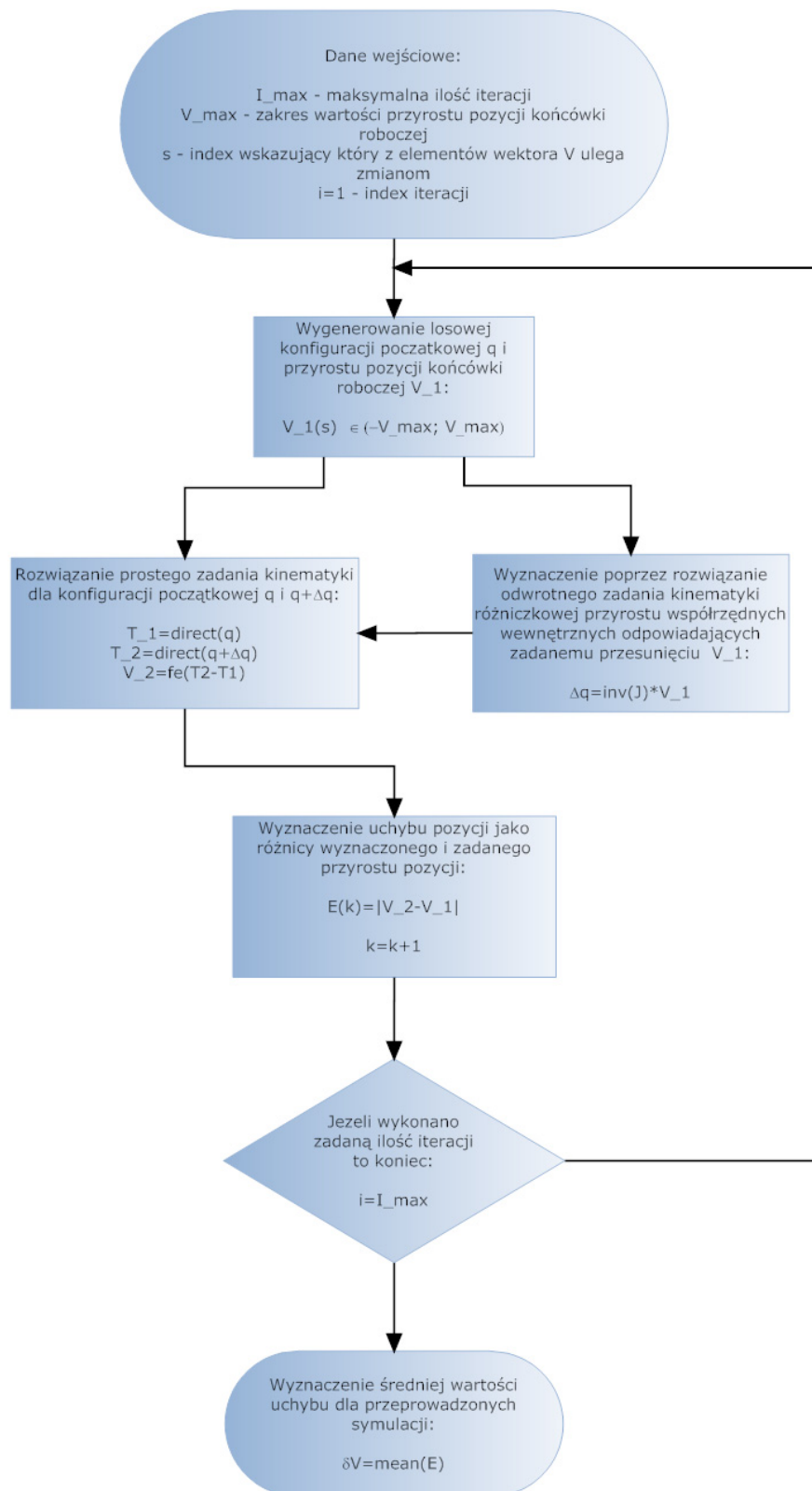
Dzięki temu, iż dla każdej ze zmiennych przegubowych Δq_i przeprowadzono odrębne testy, można zaobserwować wpływ każdej z nich na poszczególne elementy wektora prędkości. Z analizy tych wykresów wnioskujemy, iż ruch czterech pierwszych przegubów, odpowiadający przede wszystkim liniowej prędkości końcówki roboczej, wprowadza znacznie większą wartość błędu niż dwa pozostałe. Można również zauważyć, iż podczas ostatniej z symulacji, gdzie każdy z elementów wektora Δq przyjmuje niezerową wartość, uchyb każdego elementu wektora prędkości waha się na tym samym poziomie (prędkości liniowe wyrażone w metrach, a prędkości kątowe w radianach).

4.2 Uchyb dla jednego kroku odwrotnego zadania kinematyki różniczkowej

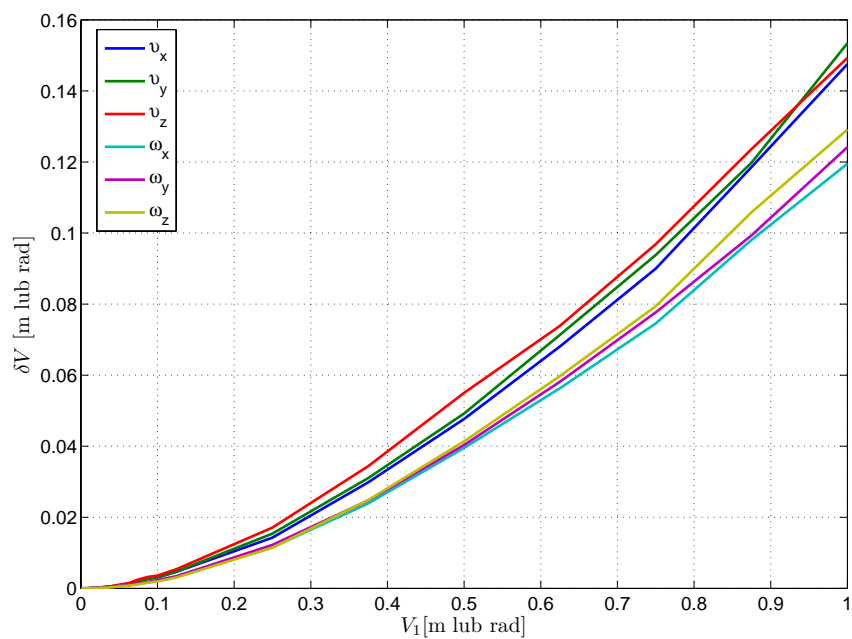
Analogiczne do powyższych testy przeprowadzone zostały dla odwrotnego zadania kinematyki.

Dla zadanej konfiguracji początkowej q i rzeczywistego przyrostu pozycji końcówki roboczej z zadanego zakresu $V_1 \in (-V_{max}, V_{max})$ rozwiązano odwrotne zadanie kinematyki różniczkowej (które odpowiada wymnożeniu zadanej wartości przesunięcia przez odwrotności jacobianu $J(q)^{-1} * V_1 = \Delta q$). Następnie dla początkowej (q) i wyznaczonej poprzez rozwiązanie odwrotnego zadania kinematyki różniczkowej ($q + \Delta q$) konfiguracji rozwiązano proste zadanie kinematyki pozycyjnej (T_1, T_2). Różnica tych macierzy ($V_2 = f_e(T_2 - T_1)$) odpowiada aproksymowanemu przyrostowi pozycji końcówki roboczej. Uchyb metody wyznaczono jako różnicę rzeczywistego (V_1) i wyznaczonego (V_2) przyrostu pozycji ($E = V_1 - V_2$).

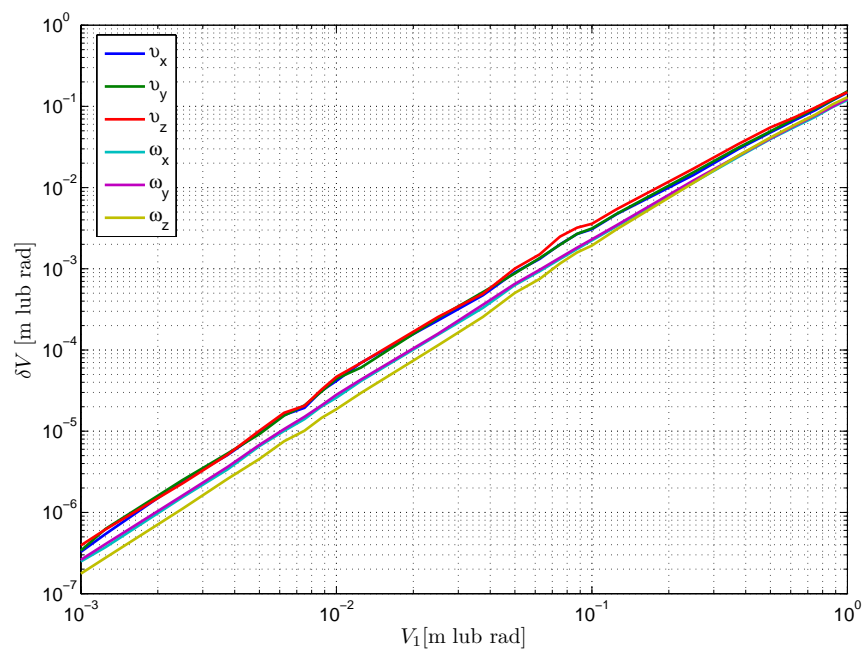
Powyższa procedura powtórzona została dla $I_{max} = 50\,000$ losowych, z pominięciem osobliwych, konfiguracji początkowych, a końcowy wynik otrzymano poprzez uśrednienie wyników pośrednich ($\delta V = mean(E)$). Otrzymane wartości uchybów przedstawione zostały na poniższych wykresach, w funkcji rosnących wartości V_{max} .



Rysunek 4.10: Schemat algorytmu symulacji odwrotnego zadania kinematyki pozycyjnej i różniczkowej



Rysunek 4.11: Uchyb wyznaczonego przesunięcia współrzędnych dla zadanego wektora V

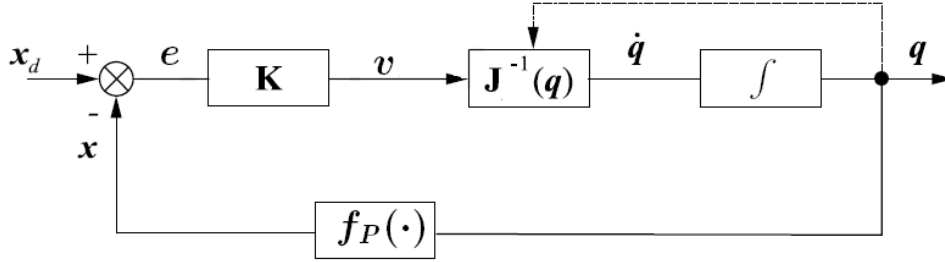


Rysunek 4.12: Uchyb wyznaczonego przesunięcia współrzędnych dla zadanego wektora V przedstawiony w skali logarytmicznej

Z przedstawionych wykresów wyczytać można, iż rząd wielkości popełnionego błędu metody jest w przybliżeniu kwadratem rzędu wielkości zadanego przyrostu pozycji.

4.3 Pętla sterowania oparta na odwrotności jakobianu J^{-1}

W kolejnych testach przebadana została struktura rozwiązania odwrotnego zadania kinematyki, opierająca się na odwrotności jakobianu (str. 63), dla przypadku gdy $x_d = \text{const}$.

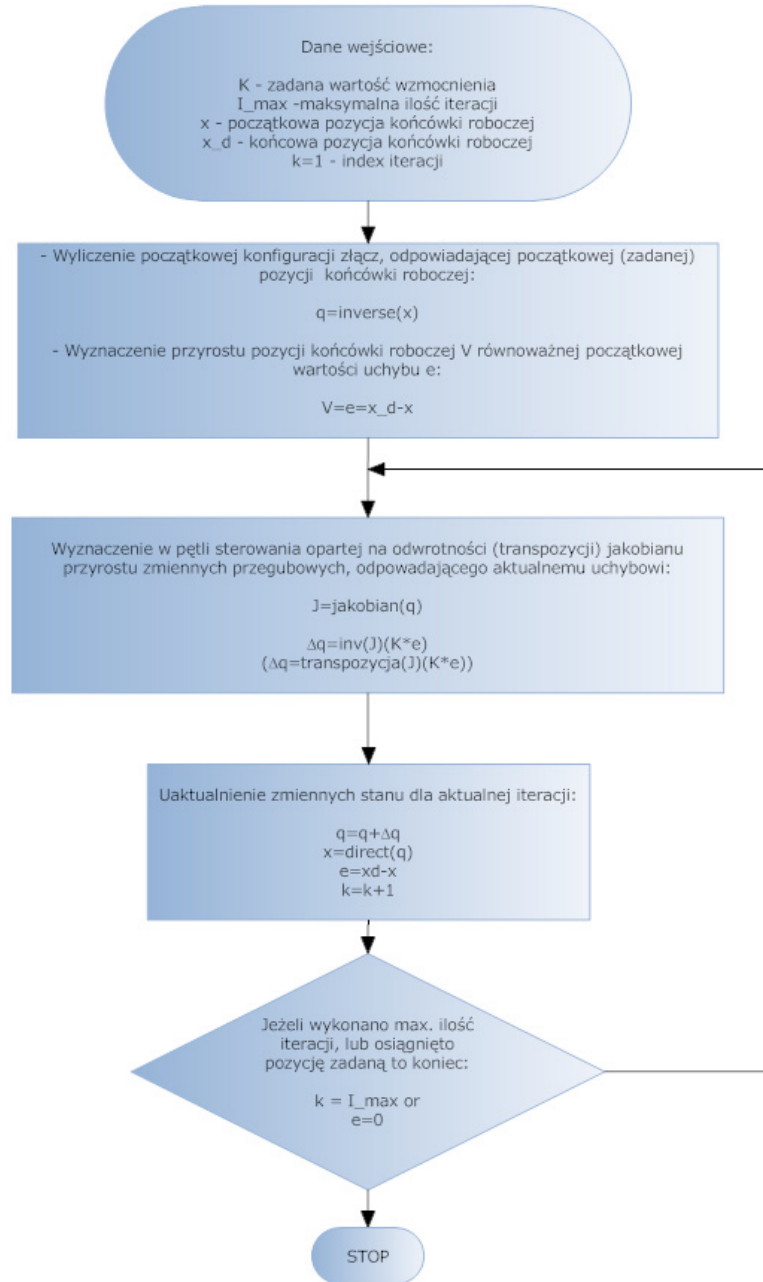


Rysunek 4.13: Struktura algorytmu oparta na odwrotności jakobianu

Danymi wejściowymi sformułowanego powyżej zadania jest startowa x i docelowa x_d pozycja końcówki roboczej, a co za tym idzie wartość zadanego przesunięcia $V = x_d - x$. Parametrem decydującym o przebiegu algorytmu jest wzmocnienie K , które według zależności (3.57) $\dot{e} = -Ke$ determinuje szybkość zmian wartości uchybu e . Im wartości macierzy K^1 bliższe jedności, tym metoda szybsza tzn. szybciej osiągnięta jest wartość docelowa pozycji, co wiąże się z coraz większymi, popełnianymi przez algorytm, błędami.

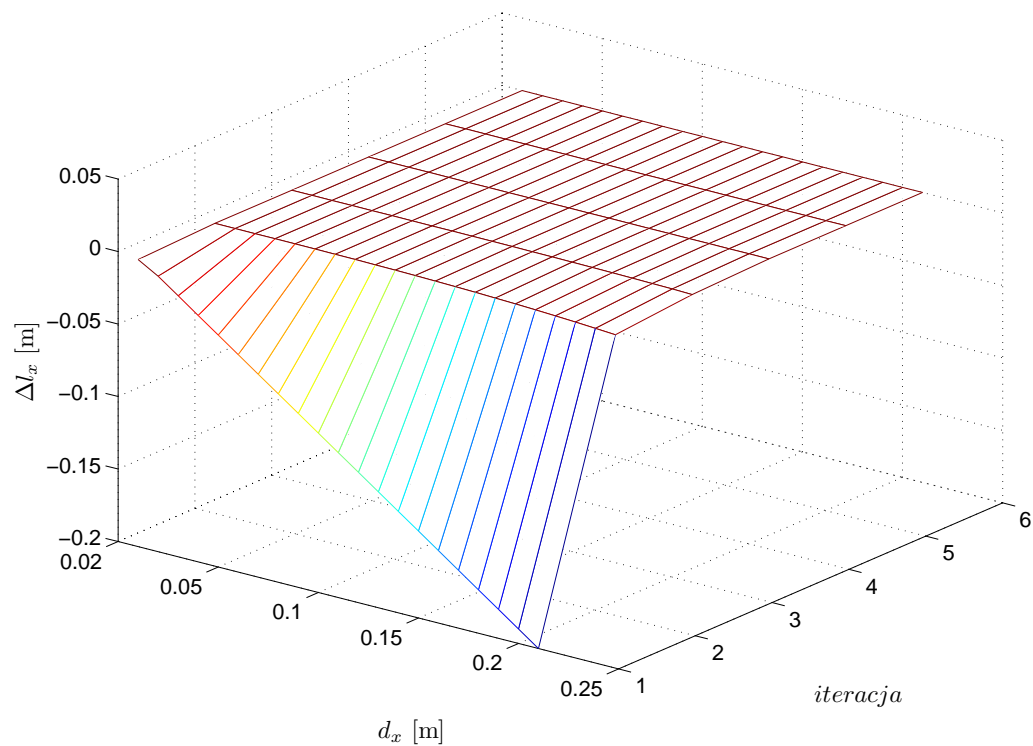
Dla każdego obiegu przedstawionej pętli sterowania wyliczona zostaje, poprzez rozwiązanie prostego zadania kinematyki pozycyjnej dla osiągniętej konfiguracji wewnętrznej, aktualna pozycja końcówki roboczej ($x = f_p(q)$) i odpowiadająca jej odległość od pozycji zadanej ($e = x_d - x$). Następnie wzmocniona wartość otrzymanego uchybu Ke podawana jest na wejście odwrotnego zadania kinematyki różniczkowej opartej na odwrotności J^{-1} jakobianu. Nowo wyznaczona wartość współrzędnych uogólnionych $q + \Delta q$ stanowi bazę dla kolejnego obiegu pętli, której warunkiem stopu jest osiągnięcie (z zadaną precyzją) przez końcówkę roboczą zadanej pozycji $e \cong 0$ lub wykonanie maksymalnej ilości iteracji.

¹W symulacjach przyjęto iż K jest macierzą diagonalną $K = I * \kappa$, gdzie I jest macierzą jednostkową.

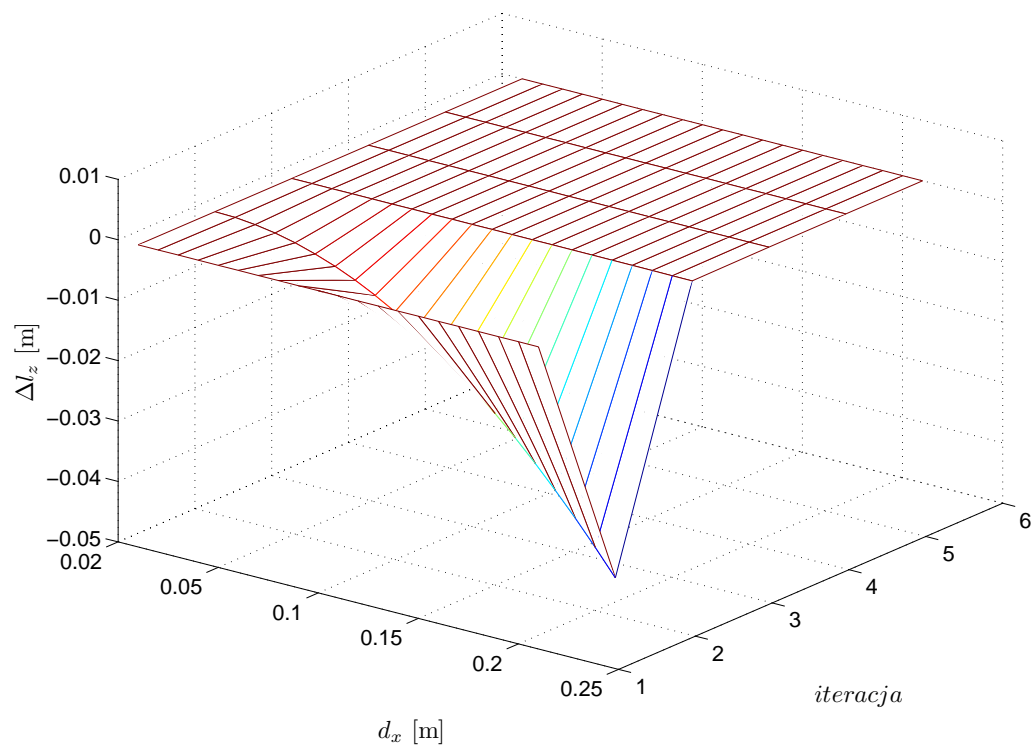


Rysunek 4.14: Schemat algorytmu symulacji pętli sterowania opartej na odwrotności (transpozycji) jacobianu dla $\dot{x}_d = 0$

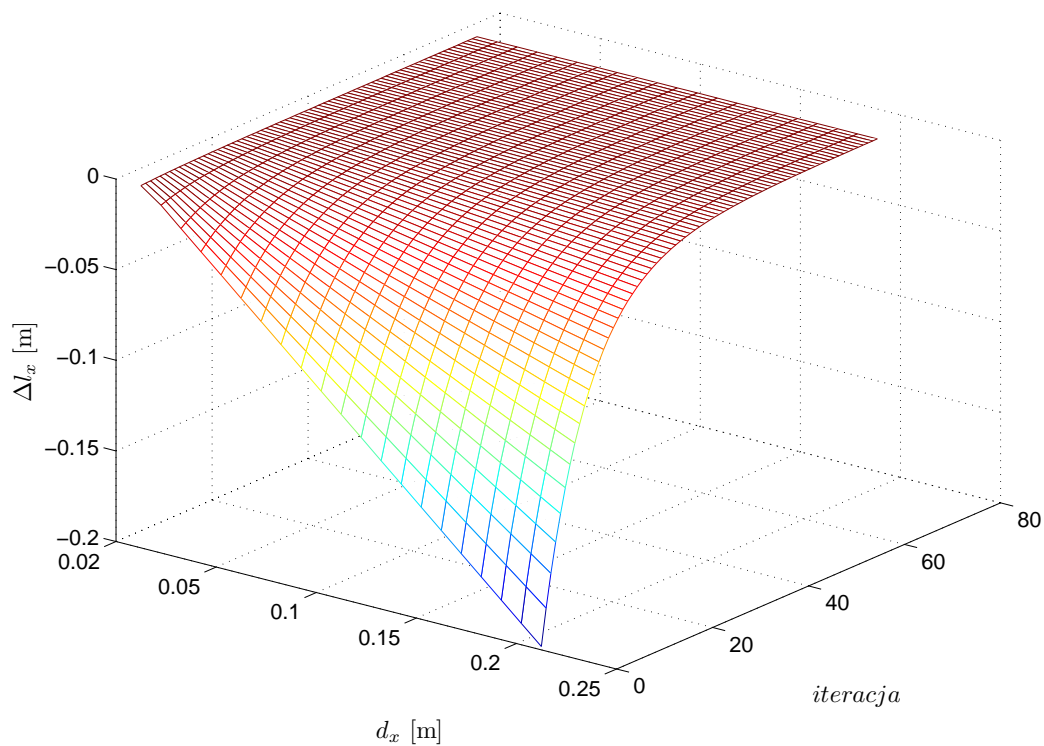
Na poniższych wykresach przedstawiono wyniki przykładowej symulacji dla początkowej konfiguracji wewnętrznej $q = [0; -\frac{\pi}{2}; 0; 0; \frac{\pi}{2}; 0]^T$ (co odpowiada pozycji $l_x = 0.82$, $l_y = 0$, $l_z = 0.45$, $\psi = 0$, $\vartheta = 0$, $\varphi = -\frac{\pi}{2}$) oraz coraz większego zadanego przesunięcia wzdłuż osi O_x ($d_x \in (0.02, 0.2)$). Poniższe symulacje przeprowadzono dla trzech różnych wartości wzmocnienia ($\kappa \in \{1, 0.1, 0.01\}$).



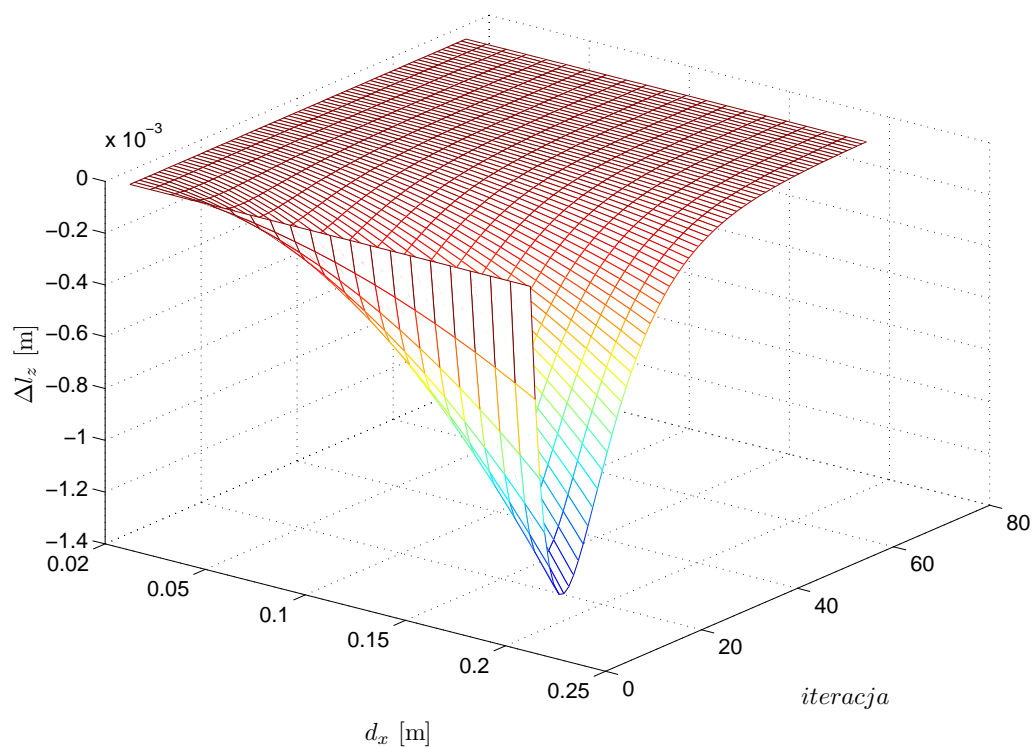
Rysunek 4.15: Uchyb położenia l_x końcówki roboczej dla pętli zawierającej J^{-1} i $\kappa = 1$



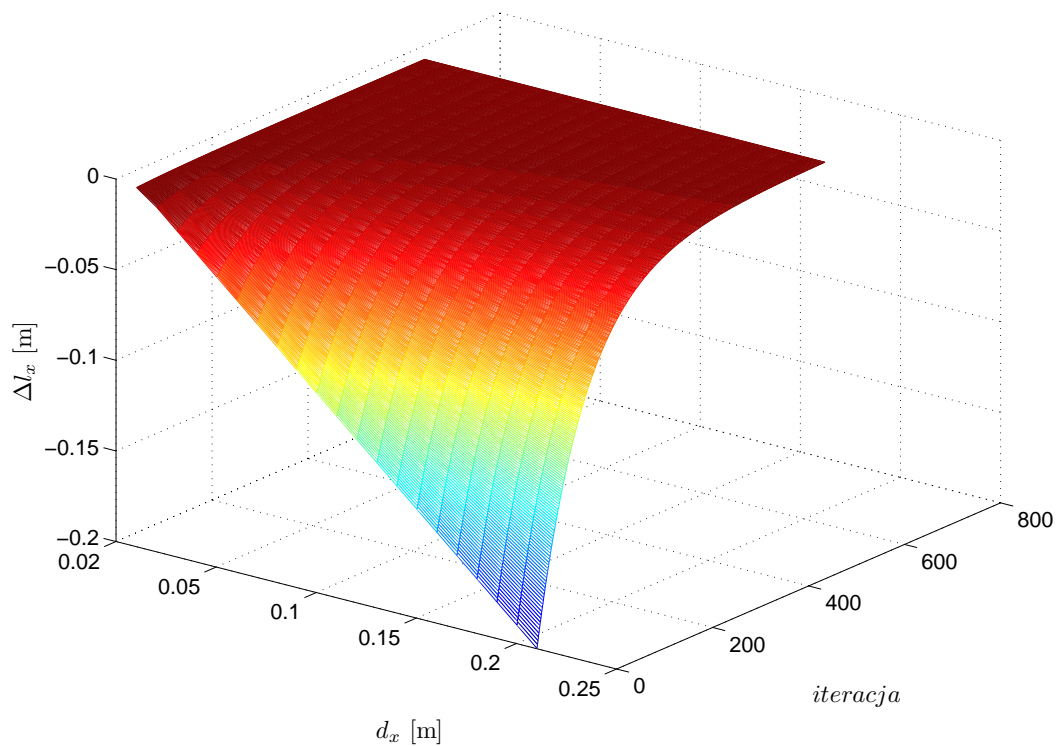
Rysunek 4.16: Uchyb położenia l_z końcówki roboczej dla pętli zawierającej J^{-1} i $\kappa = 1$



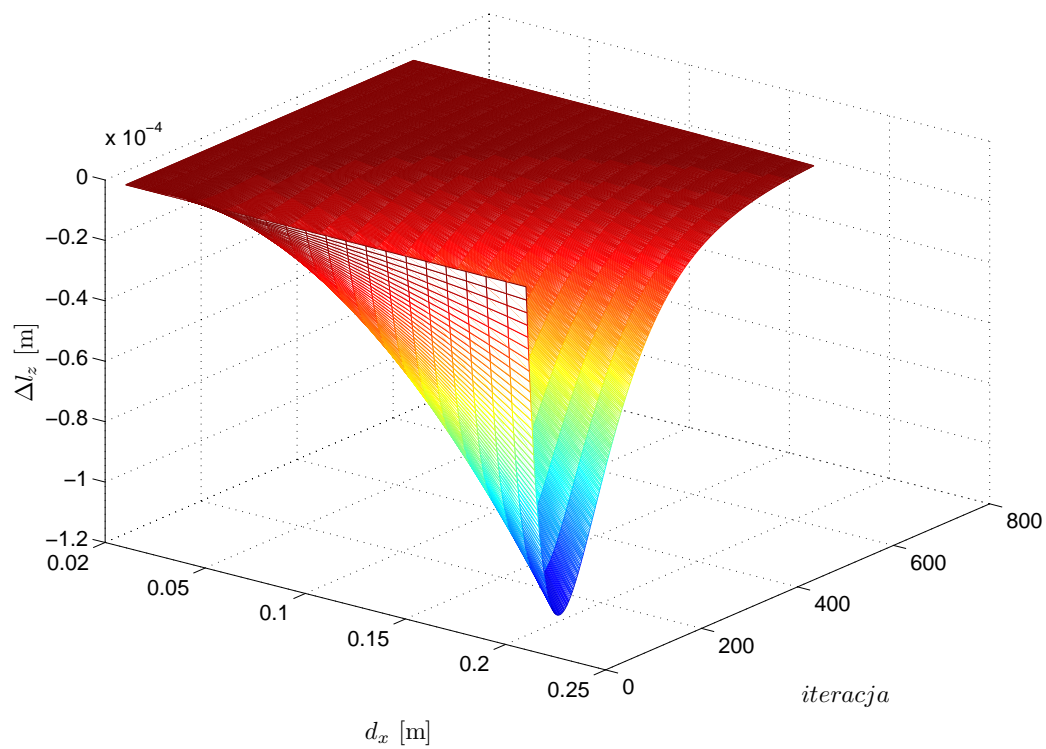
Rysunek 4.17: Uchyb położenia l_x końcówki roboczej dla pętli zawierającej J^{-1} i $\kappa = 0.1$



Rysunek 4.18: Uchyb położenia l_z końcówki roboczej dla pętli zawierającej J^{-1} i $\kappa = 0.1$



Rysunek 4.19: Uchyb położenia l_x końcówki roboczej dla pętli zawierającej J^{-1} i $\kappa = 0.01$



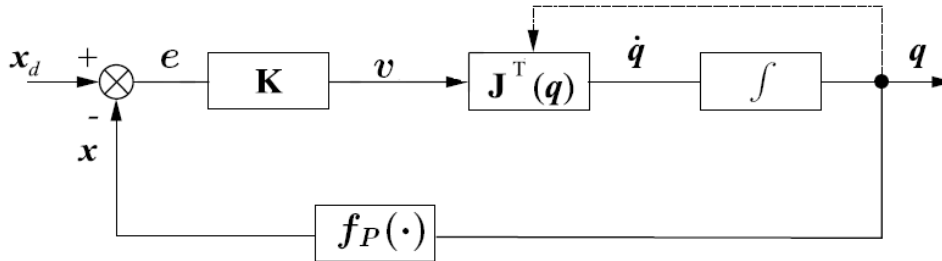
Rysunek 4.20: Uchyb położenia l_z końcówki roboczej dla pętli zawierającej J^{-1} i $\kappa = 0.01$

Widocznym na powyższych wykresach efektem ubocznym metody, opartej na odwrotnym zadaniu kinematyki różniczkowej, jest pojawienie się błędu względem innych, niż zadane poprzez przyrost V , elementów wektora pozycji. W przedstawionej symulacji dla $V = d_x$ widać wyraźny uchyb poczyniony wzdłuż osi O_z (błędy powstałe względem pozostałych elementów wektora pozycji pozostają zerowe). Z przedstawionych wyników wyczytać można, iż wartość uchybu (Δl_z) wzrasta, w przybliżeniu wykładniczo, wraz z wzrostem wielkości zadanego przesunięcia (d_x), oraz liniowo wraz z wartością wzmocnienia (κ).

Reasumując, z analizy przedstawionych wykresów widać jak wartości macierzy K , oraz wielkość zadanego przesunięcia V wpływa na warunki działania algorytmu - jego szybkość oraz wielkość popełnianych przez niego błędów.

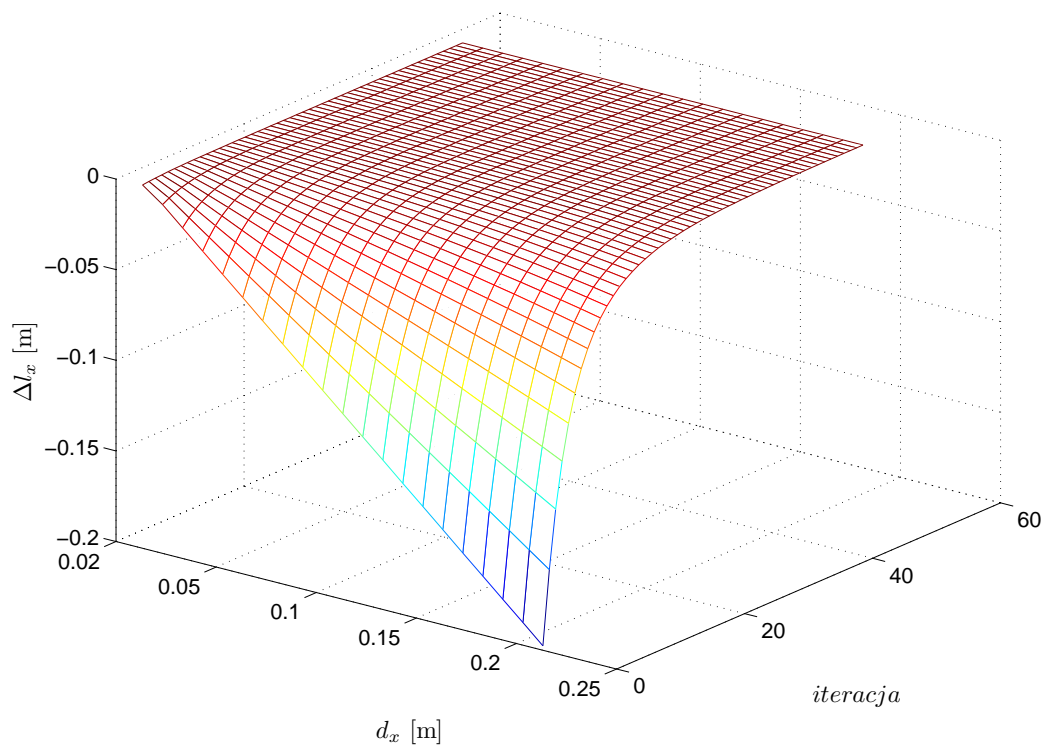
4.4 Pętla sterowania oparta na transpozycji jakobianu J^T

Analogiczne do powyższych symulacje przeprowadzone zostały dla pętli sterowania zawierającej transpozycję, w miejscu odwrotności, jakobianu. Symulacje przeprowadzono dla stałej pozycji zadanej $\dot{x}_d = 0$.

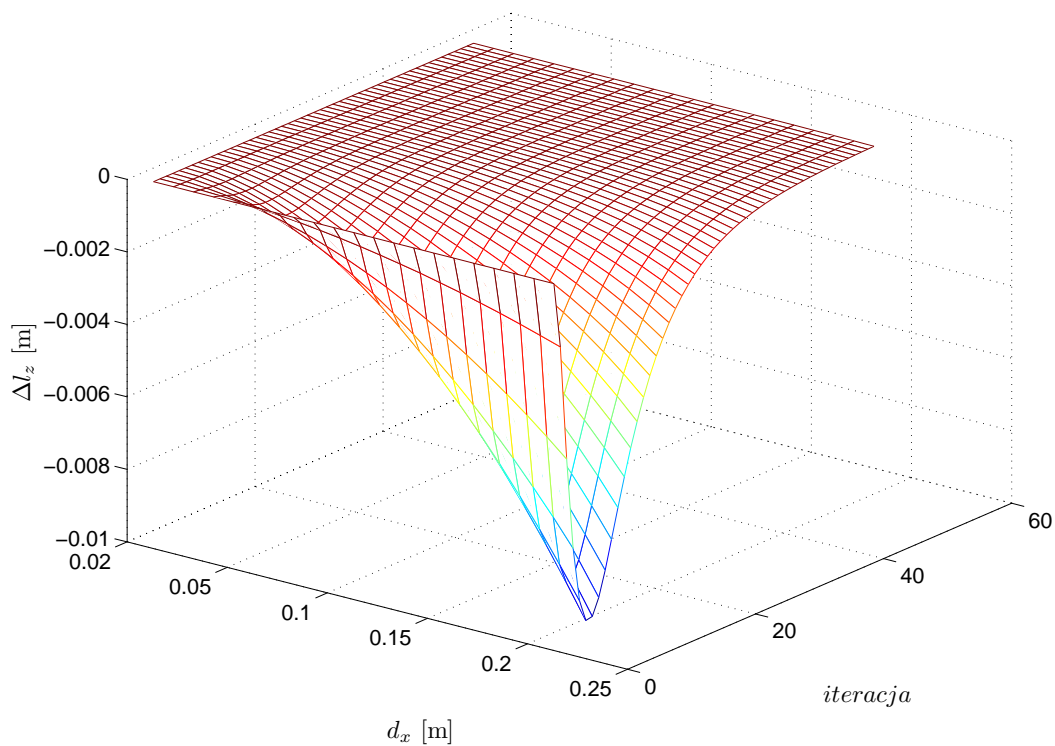


Rysunek 4.21: Struktura sterowania oparta na transpozycji jakobianu

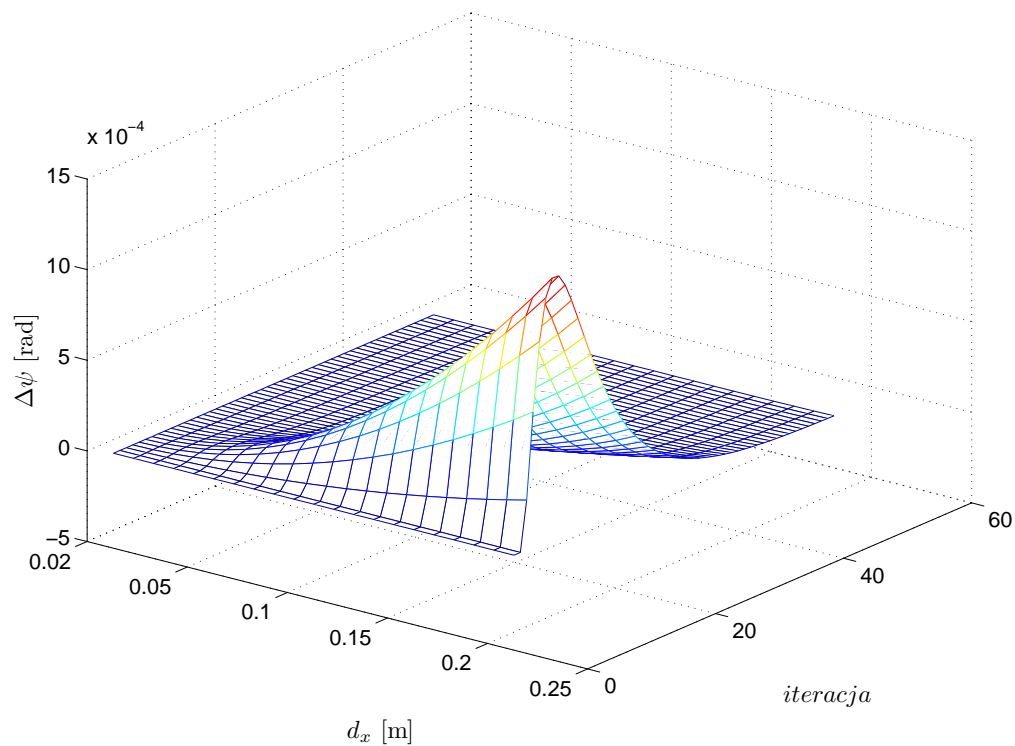
Poniżej przedstawiono wyniki symulacji, przeprowadzonych dla tych samych, co w przypadku odwrotności jakobianu, warunków początkowych. Symulacje przeprowadzono jak poprzednio dla ($\kappa \in \{1, 0.1, 0.01\}$). Jak widać wielkość uchybu Δl_z jest dużo większa niż w przypadku metody opartej na odwrotności jakobianu, a oprócz tego zauważyć można, iż uchyb względem wartości ψ przyjmuje zauważalną wielkość, podczas gdy dla odwrotności jakobianu był równy zero. Z przedstawionych wyników widać również, iż wielkość parametru κ nie wpływa na wielkość uchybu, a jedynie na szybkość algorytmu.



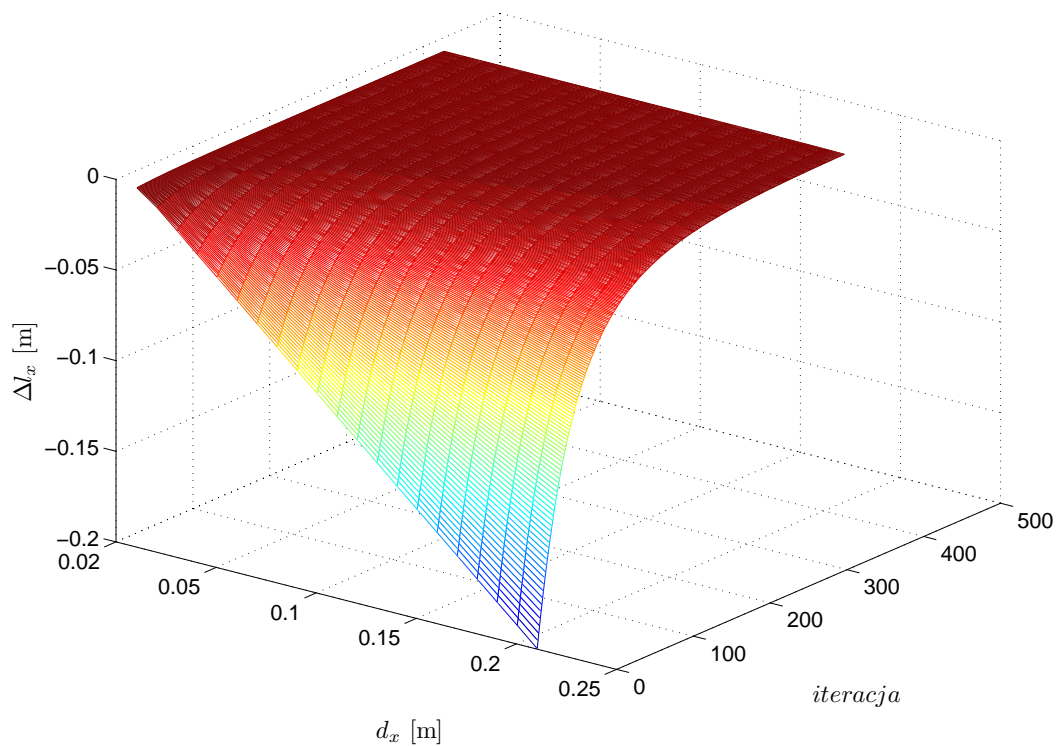
Rysunek 4.22: Uchyb położenia l_x końcówki roboczej dla pętli zawierającej J^T i $\kappa = 1$



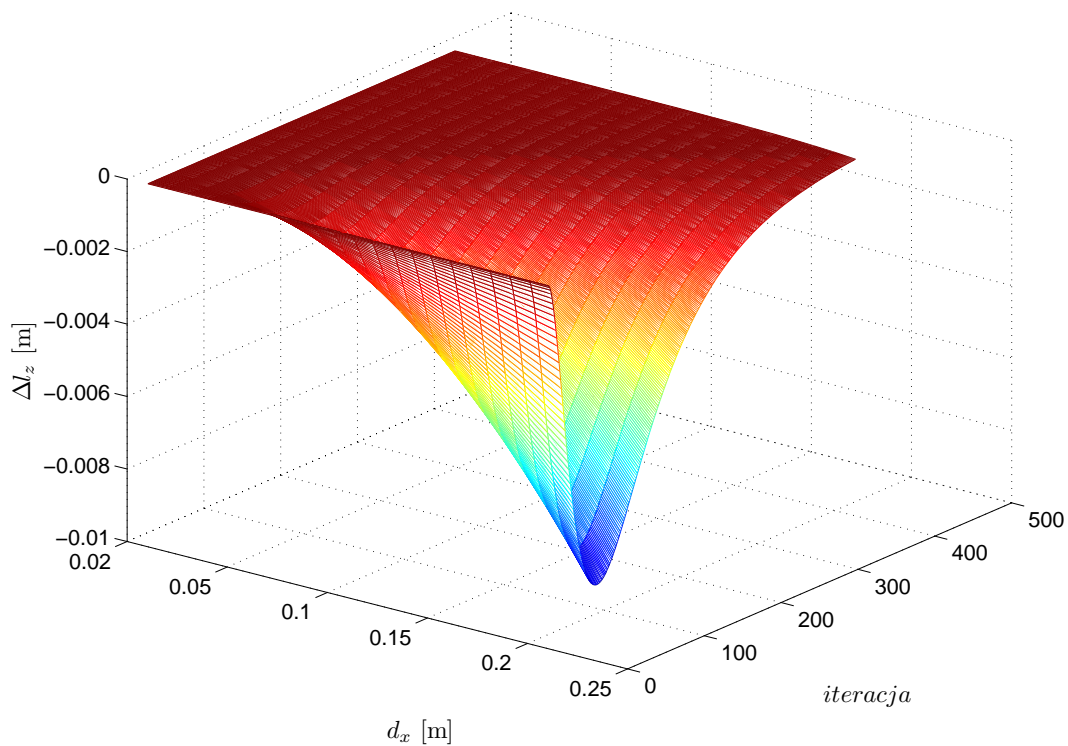
Rysunek 4.23: Uchyb położenia l_z końcówki roboczej dla pętli zawierającej J^T i $\kappa = 1$



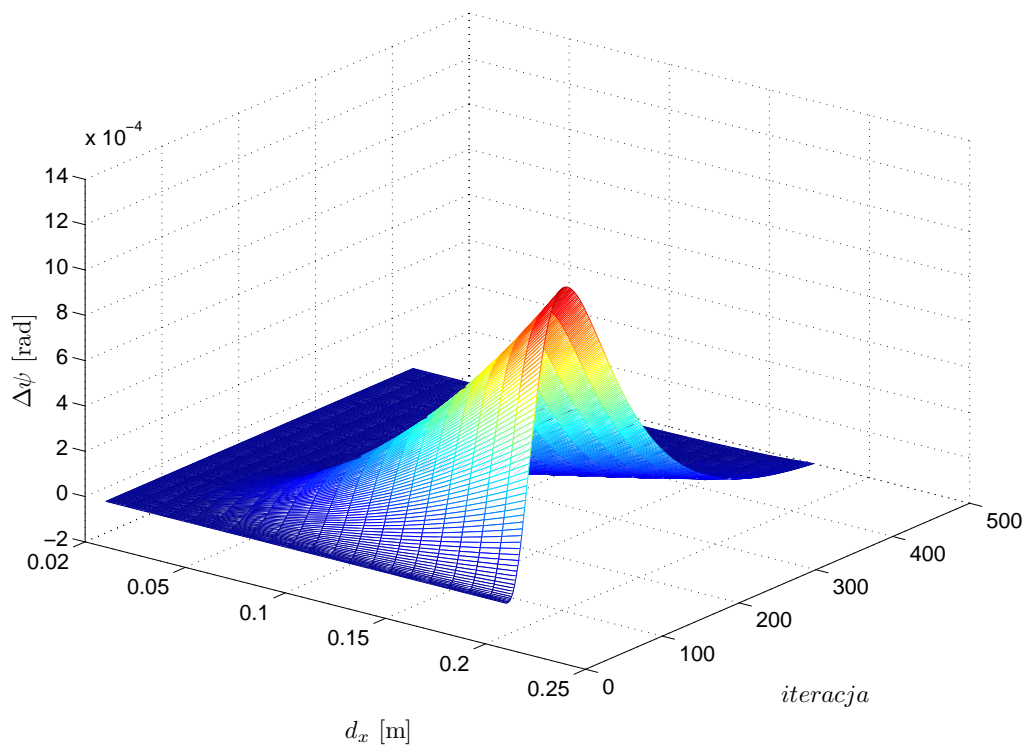
Rysunek 4.24: Uchyb położenia ψ końcówki roboczej dla pętli zawierającej J^T i $\kappa = 1$



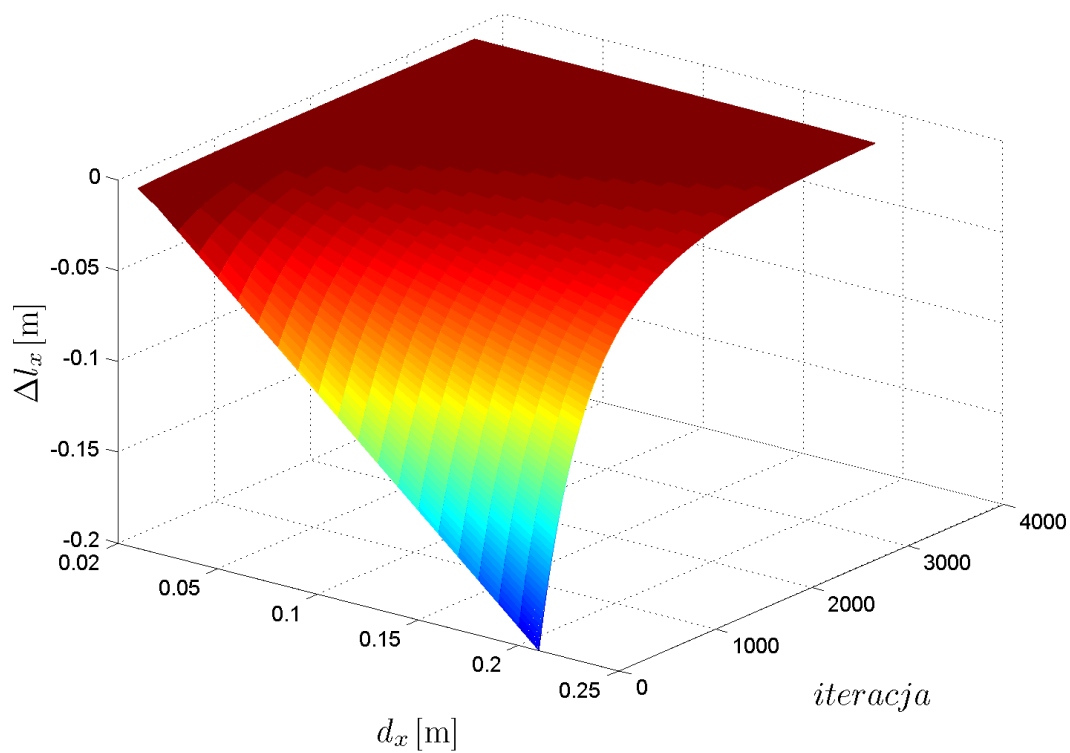
Rysunek 4.25: Uchyb położenia l_x końcówki roboczej dla pętli zawierającej J^T i $\kappa = 0.1$



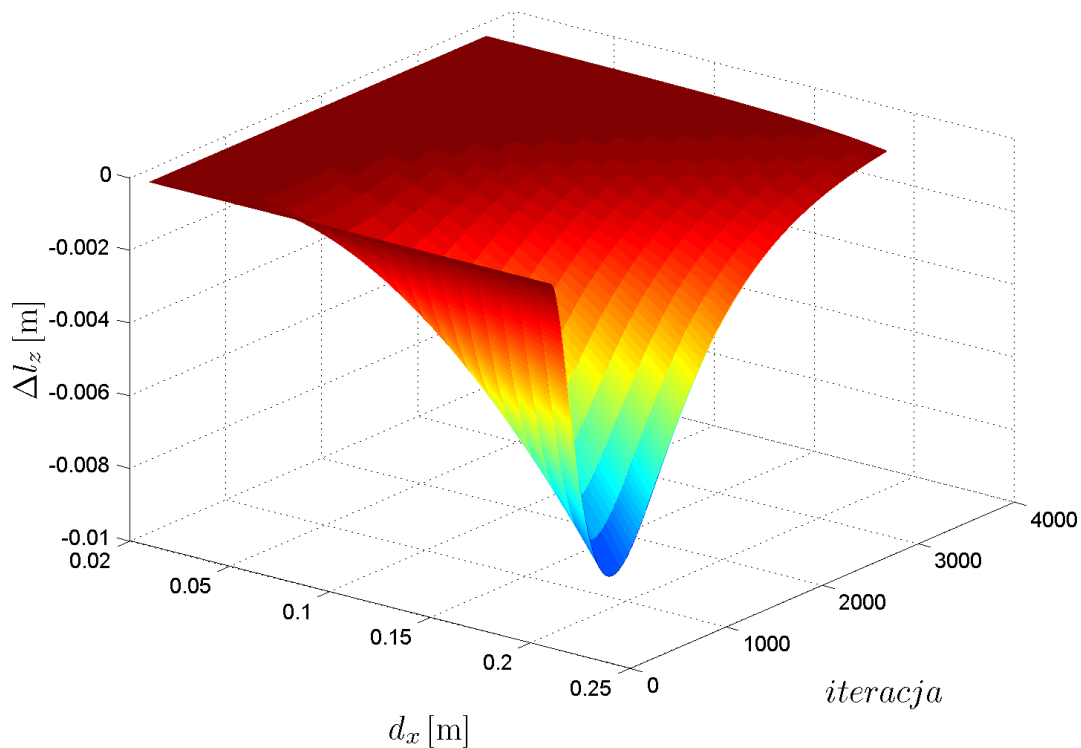
Rysunek 4.26: Uchyb położenia l_z końcówki roboczej dla pętli zawierającej J^T i $\kappa = 0.1$



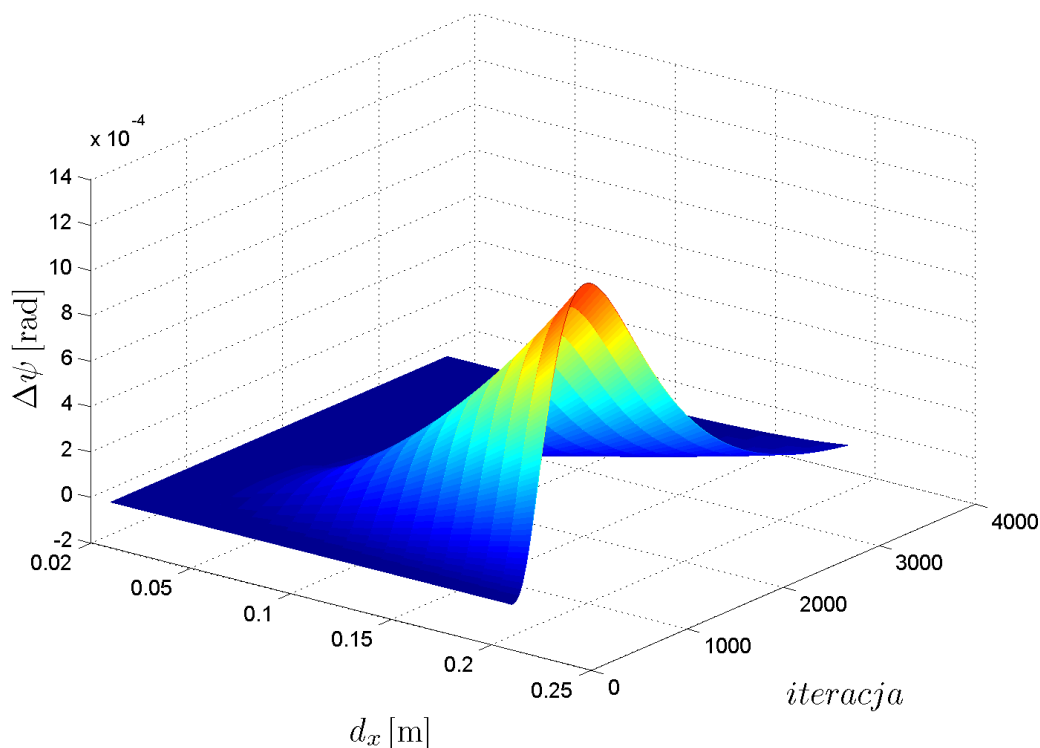
Rysunek 4.27: Uchyb położenia ψ końcówki roboczej dla pętli zawierającej J^T i $\kappa = 0.1$



Rysunek 4.28: Uchyb położenia l_x końcówki roboczej dla pętli zawierającej J^T i $\kappa = 0.01$



Rysunek 4.29: Uchyb położenia l_z końcówki roboczej dla pętli zawierającej J^T i $\kappa = 0.01$



Rysunek 4.30: Uchyb położenia ψ końcówki roboczej dla pętli zawierającej J^T i $\kappa = 0.01$

Reasumując otrzymane wyniki widać, iż warunki działania metody opartej na transpozycji jacobianu są dużo gorsze od warunków działania kinematyki różniczkowej opartej na odwrotności jacobianu - jest ona dużo od niej wolniejsza, a jej uchyby mają dużo większą wartość i rozrzut.

Rozdział 5

Wdrożenie wyznaczonych metod w strukturę MRROC++

MRROC++ jest zaimplementowaną w języku C++ programową strukturą ramową odpowiedzialną za sterowanie systemem wielorobotowego, w którego skład wchodzi, omówiony w pracy, robot IRp-6 na postumencie. Oprogramowanie MRROC++ umożliwia programowanie off-line i posiada strukturę otwartą, dzięki czemu dostęp i modyfikacja poszczególnych modułów systemu nie nastrocza wielu problemów [49, 48].

W celu wdrożenia w strukturę MRROC++ przedstawionych w pracy metod sterowania, w klasie opisującej model kinematyki dla robota IRp-6 na postumencie *kinematic_model_irp6p_with_wrist* zdefiniowane zostały poniższe metody odpowiedzialne za rozwiązanie prostego i odwrotnego zadania kinematyki pozycyjnej:

- *direct_kinematics_transform*

Rozwiązanie prostego zadania kinematyki pozycyjnej. Dla zadanych wartości wektora współrzędnych wewnętrznych (*local_current_joints*) wyznaczana jest, na podstawie zależności (2.14), macierz przekształcenia jednorodnego (*local_current_end_effector_frame*), opisująca aktualną pozycję końcówki robota.

- *inverse_kinematics_transform*

Rozwiązanie odwrotnego zadania kinematyki pozycyjnej. Na podstawie macierzy przekształcenia jednorodnego (*local_desired_end_effector_frame*), odpowiadającej zadanej pozycji końcówki roboczej wyznaczany jest, poprzez przekształcenia (2.20-2.36), docelowy wektor współrzędnych wewnętrznych (*local_desired_joints*). Dodatkowo mając dostępną informację dotyczącą aktualnych (*local_current_joints*) wartości współrzędnych przegubowych wyznaczane jest jedno z możliwych rozwiązań, w przypadku osiągnięcia konfiguracji osobliwej.

W celu wdrożenia w strukturę ramową MRROC++ opisanych w pracy algorytmów rozwiązania odwrotnego zadania kinematyki w oparciu o odwrotność i transpozycję jacobianu, zdefiniowane zostały dwie klasy dziedziczące po klasie *kinematic_model_irp6p_with_wrist* i nadpisujące wirtualną metodę *inverse_kinematics_transform*:

- *kinematic_model_irp6p_jacobian_with_wrist::inverse_kinematics_transform*
Rozwiązanie odwrotnego zadania kinematyki różniczkowej w oparciu o odwrotność jacobianu. W metodzie tej zaimplementowana została analogiczna do przedstawionej na schemacie (4.14) pętla sterowania.

W pierwszym kroku pętli dla aktualnej wartości wektora współrzędnych uogólnionych (*local_current_joints*) wyznaczana jest poprzez rozwiązanie prostego zadania kinematyki pozycyjnej (*kinematic_model_irp6p_with_wrist::direct_kinematics_transform*) postać aktualnej ramki przekształcenia jednorodnego¹. Następnie dla tak wyznaczonej macierzy oraz macierzy odpowiadającej zadanej pozycji (*local_desired_end_effector_frame*) wyznaczany jest według zależności (3.67) przyrost pozycji końcówki roboczej (*Ft_v_tr::position_distance*). W kolejnym kroku wyznaczane są wartości jacobianu dla aktualnej pozycji (*Jacobian_matrix::irp6_6dof_equations*), a w końcu metodą eliminacji Gaussa (*Jacobian_matrix::jacobian_inverse_gauss*) wyliczony zostaje przyrost położenia współrzędnych uogólnionych.

Dla nowo wyliczonych wartości współrzędnych wewnętrznych zostaje powtórnie wyznaczone proste zadanie kinematyki, oraz uchyb wyliczonej i zadanej pozycji końcówki roboczej. Powyższa pętla jest powtarzana dopóki maksymalna składowa wektora odległości (*Ft_v_tr::max_element*) nie będzie przewyższać zadanej dokładności. W momencie gdy spełniony jest powyższy warunek stopu ostatnie wyliczone wartości współrzędnych uogólnionych zostają przesłane do serwomechanizmu i manipulator wykonuje zadany krok.

- *kinematic_model_irp6p_jacobian_transpose_with_wrist::inverse_kinematics_transform*
Rozwiązanie odwrotnego zadania kinematyki różniczkowej w oparciu o transpozycję jacobianu. W metodzie tej zaimplementowana została analogiczna do powyższej pętla sterowania. Jediną różnicą pomiędzy tą, a poprzednią klasą jest zastosowanie w miejscu wywołania funkcji (*Jacobian_matrix::jacobian_inverse_gauss*), iloczynu (*Jacobian_matrix::operator**) przyrostu pozycji końcówki manipulatora i transpozycji jacobianu (*Jacobian_matrix::jacobian_transpose*).

¹Prezentowany w pracy jacobian wyznaczony został z pominięciem narzędzia traktowanego jako bryła sztywna, ergo podczas wyznaczania w tym miejscu macierzy przekształcenia jednorodnego narzędzie również zostało pominięte.

Sama struktura jacobianu została zdefiniowana w klasie zewnętrznej (*Jacobian_matrix*) dostępnej w pliku (*mathtr.cc*). W klasie tej dostępne są następujące metody:

- *Jacobian_matrix*
Domyślny konstruktor klasy inicjujący macierz jacobianu o wymiarze 6×6 wartościami zerowymi
- *irp6_6dof_equations*
Nadpisanie elementów jacobianu odpowiednimi wartościami dla manipulatora IRp-6 o sześciu stopniach swobody, zależnymi od aktualnych wartości wektora współrzędnych uogólnionych i zadanymi równaniami (3.39).
- *irp6_6dof_inverse_equations*
Wzory analityczne na odwrotność jacobianu dla manipulatora IRp-6 o sześciu stopniach swobody (3.76 - 3.111).
- *irp6_6dof_determinant*
Wyznaczenie wyznacznika jacobianu manipulatora IRp-6 o sześciu stopniach swobody według zależności (3.74).
- *jacobian_transpose*
Wyznaczenie transpozycji jacobianu.
- *jacobian_inverse_gauss*
Rozwiązanie układu równań postaci $Ax = y$ metodą eliminacji Gaussa. Na wejście metody podawana jest wartość aktualnie wyliczonego Jakobianu (A), oraz sześćoelementowy wektor odpowiadający zadanemu przesunięciu (y) będący obiektem klasy *Ft_v_tr*. W wyniku otrzymywany jest sześćoelementowy wektor (również klasy *Ft_v_tr*) odpowiadający zadanemu przesunięciu współrzędnych uogólnionych.
- *wypisz*
Wypisanie na konsole zawartości elementów macierzy jacobianu.
- *to_table*
Przepisanie wartości macierzy jacobianu do tablicy typu *double* o wymiarze 6×6 , zadanej jako argument wejściowy funkcji.
- *operator**
Przeddefiniowanie operacji mnożenia sześćoelementowego wektora przez macierz o wymiarze 6×6 . Wynikiem operacji jest sześćoelementowy wektor typu *Ft_v_tr*.

Ostatnim krokiem było dodanie do klasy opisującej sześćoelementowy wektor sił lub prędkości (*Ft_v_tr*) następujących metod:

- *position_distance*

Na podstawie aktualnej (*local_current_end_effector_frame*) i zadanej (*local_desired_end_effector_frame*) postaci ramki przekształcenia jednorodnego wyznaczany jest na podstawie zależności (3.67) wektor przesunięcia położenia końcówki roboczej.

- *max_element*

Wyznaczenie i zwrócenie jako wartość typu *double* maksymalnej wartości wektora.

Zmiany modelu kinematyki, a więc klasy odpowiedzialnej za rozwiązanie odwrotnego zadania kinematyki, można dokonać z poziomu interfejsu użytkownika, poprzez zmianę wartości parametru *kinematic_model*.

Rozdział 6

Podsumowanie

Celem zaprezentowanej pracy było przedstawienie podstaw teoretycznych, wyprowadzenie zależności odpowiadających manipulatorowi IRp-6 o sześciu stopniach swobody, oraz wdrożenie w strukturę MRROC++ struktur sterowania opartych na kinematyce pozycyjnej i kinematyce różniczkowej. Pod koniec pracy chciałabym skupić się na zaletach i wadach obu przedstawionych metod.

Kinematyka pozycyjna jest najczęściej stosowanym modelem sterowania manipulatorów. Umożliwia ona dokładne przełożenie pozycji pomiędzy końcówką roboczą manipulatora, a współrzędnymi wewnętrznymi robota. Dzięki temu, iż zadania kinematyki pozycyjnej wyrażone są za pomocą rozwikłanych, z reguły mało złożonych równań, to ich rozwiązanie nie wymaga dużego nakładu obliczeniowego, a otrzymany wynik obarczony jest znikomym błędem numerycznym. Kolejną zaletą metody jest możliwość wyboru, na podstawie ustalonego kryterium¹, jednego z dostępnych rozwiązań w przypadku osiągnięcia przez robota konfiguracji osobliwych.

Z drugiej strony samo wyprowadzenie równań kinematyki pozycyjnej jest dla wielu struktur manipulatorów zadaniem bardzo trudnym², do którego w większości przypadków niemożliwe jest podanie konkretnego schematu rozwiązania. Aby wyznaczyć te równania konieczna jest zręczna i pomysłowa manipulacja poszczególnymi zależnościami analitycznymi. Poza tym z samych równań kinematyki pozycyjnej trudno jest określić konfiguracje osobliwe struktury robota.

W odróżnieniu od kinematyki pozycyjnej, kinematyka różniczkowa nie jest metodą dokładną, a popełniany przez nią błąd, jak zostało wykazane podczas symulacji, rośnie wykładniczo wraz z wielkością zadanego przesunięcia. Struktura sterowania oparta na kinematyce różniczkowej daje jednak dużo więcej możliwości niż kinematyka pozycyjna. Przy pomocy sterowania opartego na jakobianie możliwe jest przełożenie zarówno pozycji jak i prędkości, a nawet przyspieszeń z końcówki manipulatora na poszczególne przeguby, dzięki czemu

¹Przyjętym w pracy kryterium była minimalna energia konieczna do ruchu złącza.

²W przypadku manipulatorów o strukturze szeregowej mowa tu o odwrotnym zadaniu kinematyki, z kolei dla manipulatorów równoległych - zadaniu prostym.

możliwe jest wykonanie przez manipulator skomplikowanych trajektorii gładkich o niezerowych pochodnych pierwszego i drugiego stopnia. Mało tego, sama wyprowadzona w pracy struktura jacobianu jest jedną z najważniejszych wielkości stosowanych w robotyce. Jego znajomość jest konieczna przy wyprowadzaniu dynamicznych równań ruchu oraz przy przekształcaniu sił i momentów sił z końcówki do przegubów manipulatora. Znając samą strukturę jacobianu dla danego łańcucha kinematycznego możliwe jest, poprzez wyznaczenie wyznacznika jacobianu, wyznaczenie wszystkich konfiguracji osobliwych struktury. Sam proces wyznaczenia macierzy jacobianu można przedstawić w zwartej analitycznej postaci³, dzięki czemu jego wyznaczenie może przebiegać automatycznie.

Znaczącym defektem metody, a zarazem szeroko badanym i rozwijanym problemem [6, 12, 27, 24, 43, 29], jest niemożliwość wyznaczenia rozwiązania odwrotnego zadania kinematyki różniczkowej dla konfiguracji osobliwych. Kolejną wadą metody jest duży nakład obliczeń, wiążący się koniecznością wyznaczenia odwrotności jacobianu, oraz duży, zwłaszcza przy złych uwarunkowaniach macierzy, błąd numeryczny.

Trudno jest powiedzieć, która z przedstawionych struktur sterowania jest metodą dominująca. Oczywiście w przypadku gdy podstawą zadania jest samo odwzorowanie pomiędzy pozycją końcówki manipulatora, a położeniem poszczególnych przegubów, korzystniejsza jest struktura oparta na kinematyce pozycyjnej, wnosząca znikomo mały uchyb, niezależny od wartości przesunięcia. Jednakże dla ogólniejszych postaci zadania, uwzględniających zależności prędkościowe lub siłowe, musi być zastosowana struktura sterowania opierająca się na jacobianie.

³Przy użyciu macierzy przekształcenia jednorodnego lub operatorów algebry przestrzennej

Bibliografia

- [1] prime.jsc.nasa.gov/rov. ROVer Ranch.
- [2] www.asimo.pl. Polski wortal robotyki.
- [3] www.bme.uconn.edu/bme/davinci. Uconn biomedical engineering "Da Vinci" robot.
- [4] Jain A. and Rodriguez G. An analysis of the kinematics and dynamics of underactuated manipulators. *IEEE Transactions on Robotics and Automation*, 9(4):411 – 422, August 1993. ISSN: 1042-296X. CODEN: IRAUEZ. INSPEC Accession Number: 4588454. Digital Object Identifier: 10.1109/70.246052. Posted online: 2002-08-06 18:40:19.0.
- [5] Gianluca Antonelli, Stefano Chiaverini, and Giuseppe Fusco. A new on-line algorithm for inverse kinematics of robot manipulators ensuring path tracking capability under joint limits. *IEEE Transactions on Robotics and Automation*, 19(1):162–167, February 2003. ISSN: 1042-296X. INSPEC Accession Number: 7545983. Digital Object Identifier: 10.1109/TRA.2002.807543. Posted online: 2003-02-19 12:48:57.0.
- [6] Fabrizio Caccavale, Stefano Chiaverini, and Bruno Siciliano. Second-order kinematic control of robot manipulators with jacobian damped least-squares inverse: theory and experiments. *IEEE /ASME Transactions on Mechatronics*, 2(3):188–194, September 1997. ISSN: 1083-4435. CODEN: IATEFW. INSPEC Accession Number: 5699895. Digital Object Identifier: 10.1109/3516.622971. Posted online: 2002-08-06 20:57:33.0.
- [7] Chien Chern Cheah, Masanori Hirano, Sadao Kawamura, and Suguru Arimoto. Approximate jacobian control for robots with uncertain kinematics and dynamics. *IEEE Transactions on Robotics and Automation*, 19(4):692–702, August 2003. ISSN: 1042-296X. INSPEC Accession Number: 7719816. Digital Object Identifier: 10.1109/TRA.2003.814517. Posted online: 2003-08-11 14:23:10.0.
- [8] Chien Chern Cheah, Sadao Kawamura, Suguru Arimoto, and Kiju Lee. Pid control of robotic manipulator with uncertain jacobian matrix. In

- International Conference on Robotics and Automation*, volume 1, pages 494–499, Detroit, Michigan, 10-15 May 1999. INSPEC Accession Number: 6345742. Digital Object Identifier: 10.1109/ROBOT.1999.770025. Posted online: 2002-08-06 22:31:00.0.
- [9] Chien Chern Cheah, Sadao Kawamura, Suguru Arimoto, and Kai Li. Approximate jacobian feedback control of robots with kinematic uncertainty and its application to visual servoing. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2535–2540, Seoul, Korea, 21-26 May 2001. ISSN: 1050-4729. INSPEC Accession Number: 7005852. Digital Object Identifier: 10.1109/ROBOT.2001.933004. Posted online: 2003-07-09 09:50:46.0.
- [10] Chien Chern Cheah, Chen Chung. Liu, and Hwee Choo Liaw. Stability of inverse jacobian control for robot manipulator. In *IEEE International Conference on Control Applications*, volume 1, pages 321–326, Taipei, Taiwan, 2-4 September 2004. ISSN: 1085-1992. INSPEC Accession Number: 8281716. Posted online: 2005-01-31 08:28:10.0.
- [11] Chien Chern Cheah, Chen Chung Liu, and Jean-Jacques E. Slotine. Adaptive jacobian tracking control of robots with uncertainties in kinematic, dynamic and actuator models. *IEEE Transactions on Robotics and Automation*, 51(6):1024–1029, June 2006. ISSN: 0018-9286. INSPEC Accession Number: 8990519. Digital Object Identifier: 10.1109/TAC.2006.876943. Posted online: 2006-06-19 10:21:07.0.
- [12] Stefano Chiaverini, Bruno Siciliano, and Olav Egeland. Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator. *IEEE Transactions on Control Systems Technology*, 2(2):123–134, June 1994. ISSN: 1063-6536. CODEN: IETTE2. INSPEC Accession Number: 4729858. Digital Object Identifier: 10.1109/87.294335. Posted online: 2002-08-06 19:25:43.0.
- [13] John J. Craig. *Wprowadzenie do robotyki. Mechanika i sterowanie*. Wydawnictwo Naukowo-Techniczne, 1993.
- [14] Adam Currie. www.faculty.ucr.edu/currie/roboadam.htm. The history of robotics.
- [15] Hari Das, Jean-Jacques E. Slotine, and Thomas B. Sheridan. Inverse kinematic algorithms for redundant systems. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 43–48, Philadelphia, PA, 24-29 April 1988. INSPEC Accession Number: 3247360. Digital Object Identifier: 10.1109/ROBOT.1988.12021. Posted online: 2002-08-06 15:56:49.0.

- [16] Bentley W. Drake and T. C. Steve Hsia. Implementation of a unified robot kinematics and inverse dynamics algorithm on a dsp chip. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1193–1200, Raleigh, NC, 7-10 July 1992. Posted online: 2002-08-06 18:12:46.0.
- [17] Rodriguez G., Kreutz-Delgado K., and Milman M. A spatial operator algebra for manipulator modelling and control. volume 10, pages 418 – 423, Arlington, VA, 24-26 August 1988. INSPEC Accession Number: 3685643. Digital Object Identifier: 10.1109/ISIC.1988.65468. Posted online: 2002-08-06 16:03:31.0.
- [18] Luis G. Herrera-Bendezu, Eduardo Mu, and James T. Cain. Symbolic computation of robot manipulator kinematics. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 993–998, Philadelphia, PA, 24-29 April 1988. INSPEC Accession Number: 3228088. Digital Object Identifier: 10.1109/ROBOT.1988.12190. Posted online: 2002-08-06 15:57:08.0.
- [19] Koh Hosoda, Katsuji Igarashi, and Minoru Asada. Adaptive hybrid visual servoing / force control in unknown environment. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1097–1103, Osaka, 4-8 November 1996. INSPEC Accession Number: 5514166. Digital Object Identifier: 10.1109/IROS.1996.568956. Posted online: 2002-08-06 20:48:04.0.
- [20] Abhinandan Jain and Guillermo Rodriguez. An analysis of the kinematics and dynamics of underactuated manipulators. *IEEE Transactions on Robotics and Automation*, 9(4):411–422, August 1993. ISSN: 1042-296X. CODEN: IRAUEZ. INSPEC Accession Number: 4588454. Digital Object Identifier: 10.1109/70.246052. Posted online: 2002-08-06 18:40:19.0.
- [21] Caenen Jean-Luck and Angue Jean-Claude. Identification of geometric and nongeometric parameters of robots. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1032–1037, Cincinnati, OH, 13-18 May 1990. INSPEC Accession Number: 3863751. Digital Object Identifier: 10.1109/ROBOT.1990.126128. Posted online: 2002-08-06 17:13:06.0.
- [22] Wisama Khalil and Jean-Francois Kleinfinger. A new geometric notation for open and closed-loop robots. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 1174–1179, April 1986. Posted online: 2003-01-06 17:04:34.0.
- [23] Tatsuya Kishimoto and Yasutaka Fujimoto. Numerically stable inverse kinematics calculation of robot manipulators even in singular configura-

- tion. In *30th Annual Conference of IEEE Industrial Electronics Society*, volume 2, pages 1036–1040, Busan, Korea, 2-6 November 2004. INSPEC Accession Number: 8402711. Digital Object Identifier: 10.1109/IECON.2004.1431717. Posted online: 2005-05-23 09:06:47.0.
- [24] Koichi Kogenezawa. Redundancy resolution for robot manipulators - comparison of computational efficiency between the svds, the fast similarity factorization and recursive formulation. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 89–95, Alberta, Canada, July 29-August 1 2001. INSPEC Accession Number: 7348246. Digital Object Identifier: 10.1109/CIRA.2001.1013178. Posted online: 2002-08-07 00:04:45.0.
- [25] Krzysztof Kozłowski, Piotr Dutkiewicz, and Waldemar Wróblewski. *Planowanie zadań i programowanie robotów*. Wydawnictwo Politechniki Poznańskiej, 1999.
- [26] Serdar Kucuk and Zafer Bingul. The inverse kinematics solutions of fundamental robot manipulators with offset wrist. In *IEEE International Conference on Mechatronics*, pages 197–202, Taipei, Taiwan, 10-12 July 2005. INSPEC Accession Number: 8624432. Posted online: 2005-11-14 10:11:02.0.
- [27] Jadran Lenarcic. Alternative computational scheme of manipulator inverse kinematics. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 3235–3240, Leuven, Belgium, 16-20 May 1998. INSPEC Accession Number: 5966363. Digital Object Identifier: 10.1109/ROBOT.1998.680926. Posted online: 2002-08-06 21:43:20.0.
- [28] Chao Liu and Chien Chern Cheah. Task-space adaptive setpoint control for robots with uncertain kinematics and actuator model. *IEEE Transactions on Automatic Control*, 50(11):1854–1860, November 2005. ISSN: 0018-9286. INSPEC Accession Number: 8646610. Digital Object Identifier: 10.1109/TAC.2005.858664. Posted online: 2005-11-14 10:09:52.0.
- [29] Ren V. Mayorga, Andrew K.C. Wong, and N. Milano. A fast procedure for manipulator inverse kinematics evaluation and pseudoinverse robustness. *IEEE Transactions on Systems, Man and Cybernetics*, 22(4):790–798, July-August 1992. ISSN: 0018-9472. CODEN: ISYMAW. INSPEC Accession Number: 4277475. Digital Object Identifier: 10.1109/21.156590. Posted online: 2002-08-06 18:10:42.0.
- [30] Claudio Melchiorri and J. Kenneth Salisbury. An algorithm for the control of a hand-arm robotic system. In *Fifth International Conference on Advanced Robotics*, volume 1, pages 471–476, Pisa, 19-22 June

1991. INSPEC Accession Number: 4462987. Digital Object Identifier: 10.1109/ICAR.1991.240609. Posted online: 2002-08-06 17:57:31.0.
- [31] Simon Moncleton and Dale Cherkas. Jacobian transpose control: a foundation for multiagent manipulator control. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 3391–3396, Vancouver, BC, 22-25 October 1995. INSPEC Accession Number: 5138644. Digital Object Identifier: 10.1109/ICSMC.1995.538310. Posted online: 2002-08-06 20:10:47.0.
- [32] Sudath R. Munasinghe, Masatoshi Nakamura, Satoru Goto, and Nobuhiro Kyura. Precise control of industrial robot arms considering trajectory allowance under torque and speed constraints. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 3949–3954, Seoul, Korea, 21-26 May 2001. ISSN: 1050-4729. INSPEC Accession Number: 7012183. Digital Object Identifier: 10.1109/ROBOT.2001.933233. Posted online: 2003-07-09 09:51:13.0.
- [33] Ataru Nakamura, Yasuhiro Ohyama, Kazuhiko Ito, and Koji Saito. Controller for industrial robots. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 254–259, April 1986. Posted online: 2003-01-06 17:04:26.0.
- [34] Zoran R. Novakovic and Bojan Nemec. A solution of the inverse kinematics problem using the sliding mode. *IEEE Transactions on Robotics and Automation*, 6(2):247–252, April 1990. ISSN: 1042-296X. CODEN: IRAUEZ. INSPEC Accession Number: 3703593. Digital Object Identifier: 10.1109/70.54740. Posted online: 2002-08-06 17:22:45.0.
- [35] Pablo Sanchez-Sanchez and Fernando Reyes-Cortes. Cartesian controllers evaluation in joint space. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2059–2064, Beijing, China, 9-15 October 2006. INSPEC Accession Number: 9407719. Digital Object Identifier: 10.1109/IROS.2006.282419. Posted online: 2007-01-15 13:04:07.0.
- [36] Pablo Sanchez-Sanchez and Fernando Reyes-Cortes. New control structure in cartesian space. In *IEEE International Conference on Mechatronics and Automation*, pages 1980–1985, Luoyang, China, 25-28 June 2006. Posted online: 2006-12-11 09:21:22.0.
- [37] Lorenzo Sciavicco and Bruno Siciliano. A solution algorithm to the inverse kinematic problem for redundant manipulators. *IEEE Journal of Robotics and Automation*, 4(4):403–410, August 1988. ISSN: 0882-4967. CODEN: IJRAE4. INSPEC Accession Number: 3222952. Digital Object Identifier: 10.1109/56.804. Posted online: 2002-08-06 15:49:17.0.

- [38] Jon M. Selig. Active versus passive transformations in robotics. *IEEE Robotics and Automation Magazine*, 13(1):79–84, March 2006. ISSN: 1070-9932. INSPEC Accession Number: 8805313. Digital Object Identifier: 10.1109/MRA.2006.1598057. Posted online: 2006-02-27 09:48:43.0.
- [39] Mark W. Spong and Mathukumalli Vidyasagar. *Dynamika i sterowanie robotów*. Wydawnictwo Naukowo-Techniczne, 1997.
- [40] Henry Stone, Arthur Sanderson, and Charles Neuman. Arm signature identification. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 41–48, April 1986. Posted online: 2003-01-06 17:04:38.0.
- [41] Taro Takahashi and Atsuo Kawamura. High speed numerical calculation method for on-line inverse kinematics of robot manipulators. In *IECON '99. The 25th Annual Conference of the IEEE Industrial Electronics Society*, volume 1, pages 436–441, San Jose, CA, 29 November–3 December 1999. INSPEC Accession Number: 6514708. Digital Object Identifier: 10.1109/IECON.1999.822237. Posted online: 2002-08-06 22:52:42.0.
- [42] Ulrike Thomas, I. Maciuszek, and Friedrich M. Wahl. A unified notation for serial, parallel, and hybrid kinematic structures. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2868–2873, Washington, 11–15 May 2002. INSPEC Accession Number: 7379469. Digital Object Identifier: 10.1109/ROBOT.2002.1013667. Posted online: 2002-08-07 00:46:41.0.
- [43] Han-Lei Wang, Jia-Wei Li, and Hong Liu. Practical limitations of an algorithm for the singular value decomposition as applied to redundant manipulators. In *IEEE Conference on Robotics, Automation and Mechatronics*, pages 1–6, Bangkok, December 2006. INSPEC Accession Number: 9273746. Digital Object Identifier: 10.1109/RAMECH.2006.252609. Posted online: 2006-12-04 08:25:14.0.
- [44] Chi-Haur Wu and Kuu-Young Young. An efficient solution of a differential inverse kinematics problem for wrist-partitioned robots. *IEEE Transactions on Robotics and Automation*, 6(1):117–123, February 1990. ISSN: 1042-296X. CODEN: IRAUEZ. INSPEC Accession Number: 3621643. Digital Object Identifier: 10.1109/70.88126. Posted online: 2002-08-06 17:28:20.0.
- [45] Fengfeng Xi and Robert G. Fenton. Modeling and analysis of flexible link manipulators using the algebra of rotations. *IEEE Transactions on Systems, Man and Cybernetics*, 24(7):1022–1034, July 1994. ISSN: 0018-9472. CODEN: ISYMAW. INSPEC Accession Number: 4742111. Digital Object Identifier: 10.1109/21.297790. Posted online: 2002-08-06 19:29:53.0.

-
- [46] Jian Xie, Shizuo Yan, and Wenyi Qiang. A method for solving the inverse kinematics problem of 6-dof space manipulator. In *ISSCAA 2006. 1st International Symposium on Systems and Control in Aerospace and Astronautics*, page 4 pp., 19-21 January 2006. INSPEC Accession Number: 9005105. Digital Object Identifier: 10.1109/ISSCAA.2006.1627648. Posted online: 2006-05-08 09:30:03.0.
- [47] Hanqi Zhuang, Zvi S. Roth, and Fumio Hamano. A complete and parametrically continuous kinematic model for robot manipulators. *IEEE Transactions on Robotics and Automation*, 8(4):451–463, August 1992. ISSN: 1042-296X. CODEN: IRAUEZ. INSPEC Accession Number: 4242477. Digital Object Identifier: 10.1109/70.149944. Posted online: 2002-08-06 18:09:48.0.
- [48] Cezary Zieliński and Wojciech Szyrkiewicz. Podręcznik programowania systemu mrroc++ dla robota irp-6 na torze jezdny. Technical report, Instytut Automatyki i Informatyki Stosowanej Politechnika Warszawska, April 1999.
- [49] Cezary Zieliński and Wojciech Szyrkiewicz. System mrroc++ dla robota irp-6. Technical report, Instytut Automatyki i Informatyki Stosowanej Politechnika Warszawska, May 1999.