

dr inż. Wojciech Szynkiewicz\*

dr inż. Jacek Błaszczuk†

prof. dr hab. Krzysztof Malinowski\*†

\*Instytut Automatyki i Informatyki Stosowanej Politechniki Warszawskiej

†Naukowa i Akademicka Sieć Komputerowa NASK

## OPTIMALIZACJA W ZASTOSOWANIU DO PLANOWANIA RUCHU ŚCIŚLE WSPÓLPRACUJĄCYCH ROBOTÓW

*W artykule<sup>1</sup> omówiono wykorzystanie zaawansowanych technik optymalizacji do rozwiązywania zadań planowania ścieżek ruchu dla ściśle współpracujących robotów. Zadanie planowania ścieżki jest sformułowane jako problem minimalizacji warunkowej funkcjonalu, a następnie jest sprowadzane do zadania programowania nieliniowego (NLP). Do numerycznego rozwiązania zadania NLP wykorzystuje się solver IPOPT oparty na prymalno-dualnej metodzie punktu wewnętrznego dla zadań nieliniowych, będącej obecnie jedną z wiodących technik optymalizacji nieliniowej dla zadań wielkiej skali.*

### OPTIMIZATION IN MOTION PLANNING FOR TIGHTLY COOPERATING ROBOTS

*Application of advanced optimization techniques to solve the path planning problem for tightly cooperating robots is discussed in this paper. The approach to path planning is formulated as a "quasi-dynamic" nonlinear optimization (NLP) problem with equality and inequality constraints in terms of the joint variables. The essence of the method is to find joint paths which satisfy the given constraints and minimize the proposed performance index. For numerical solution of the NLP problem the IPOPT solver is used, which implements a nonlinear primal-dual interior-point method one of the leading techniques for large-scale nonlinear optimization.*

## 1. WPROWADZENIE

Planowanie działań, w szczególności planowanie ruchu, jest jednym z istotnych problemów w robotyce. Sformułowanie problemu planowania ruchu jako zadania optymalizacyjnego, a następnie jego rozwiązanie z wykorzystaniem odpowiednich metod numerycznych jest jednym z stosowanych podejść [15]. W niniejszej pracy jest rozważane zagadnienie planowania ruchu i koordynacji działań zespołu robotów wykonujących czynności manipulacyjne wymagające ścisłej współpracy robotów. Dla ściśle współpracujących robotów, gdy wymagana jest ścisła koordynacja działań/ruchów robotów, już znalezienie trajektorii dopuszczalnych jest złożonym problemem [28, 40]. Wspólne przeniesienie przez kilka robotów jednego obiektu jest typowym przykładem takiego zadania.

W zadaniach manipulacji obiekt jest centralnym elementem systemu robotycznego, którego ruch i wywierane nań siły są przedmiotem analizy przy planowaniu manipulacji. Po uchwyceniu obiektu przez palce/chwytyki tworzy się mechanizm wielo-przegubowy zawierający jeden lub więcej zamkniętych łańcuchów (pętli) kinematycznych. W obszarze

<sup>1</sup>Praca została wykonana w ramach Projektów Badawczych MNiSzW N N514 1287 33 oraz N N514 4169 34.

robotyki można znaleźć więcej przykładów systemów, w których występują pętle kinematyczne. Roboty równoległe są same w sobie strukturami zawierającymi zamknięte łańcuchy [24]. W maszynach kroczących zamknięty łańcuch tworzy się w chwili kontaktu nóg z podłożem [41]. Poza obszarem robotyki zamknięte łańcuchy są przedmiotem badań m.in. w grafice komputerowej, wirtualnym prototypowaniu, a także chemii i biologii molekularnej [13, 18, 39].

Planowanie ruchu dla zamkniętego łańcucha jest wykonywane w przestrzeni konfiguracyjnej o dużym wymiarze i przy skomplikowanych ograniczeniach fizycznych [31, 40, 28]. Więzy wynikające z warunku zamknięcia pętli kinematycznej istotnie ograniczają możliwości ruchowe układu. W pracach dotyczących planowania ruchu dla zamkniętych łańcuchów kinematycznych zazwyczaj przyjmuje się, że obiekt, stanowiący jedno z ogniw łańcucha, jest sztywno uchwycony i konfiguracje chwytów są zadane i stałe [17, 20, 28].

Należy podkreślić, iż zagadnienie planowania ruchu dla mechanizmów z pętlami kinematycznymi jest ciągle otwartym problemem badawczym. Opisane w literaturze nieliczne przykłady algorytmów zupełnych odnoszą się zazwyczaj do konkretnych klas systemów i wykorzystują ich charakterystyczne cechy oraz dają zadowalające wyniki przy wielu założeniach upraszczających. Trinkle i Milgram zaproponowali algorytm zupełny dla płaskich łańcuchów z kilkoma przegubami typu obrotowego [31]. Jednakże w tym algorytmie nie uwzględnia się istnienia przeszkód i unikania wzajemnych kolizji między ogniwami łańcucha.

Przekleństwo wymiarowości występujące w zadaniach planowania ruchu robotów skłoniło wielu badaczy do rozwijania technik losowych, które choć nie są algorytmami zupełnymi, umożliwiają znalezienie rozwiązania w przestrzeniach wielowymiarowych. Szczególnie intensywnie są rozwijane metody planowania ścieżek ogólnie nazywane metodami probabilistycznych map dróg (ang. *Probabilistic Roadmap Methods*) [9, 17, 19] oraz algorytmy RRT (ang. *Rapidly-exploring Randomized Trees*) [21]. W obu podejściach planowanie składa się z dwóch faz. Faza pierwsza polega na utworzeniu nieskierowanego grafu (mapy) dróg w swobodnej przestrzeni konfiguracyjnej. Węzłami grafu są losowo wygenerowane konfiguracje, a krawędziami lokalne ścieżki łączące węzły. Następnie w drugiej fazie, zwanej fazą zapytań, utworzony graf jest przeszukiwany w celu znalezienia ścieżki łączącej zadaną konfigurację początkową z konfiguracją docelową [19, 23]. Kluczowymi czynnikami decydującymi o efektywności realizacji konkretnego algorytmu są sposoby generowania punktów (konfiguracji) w przestrzeni swobodnej oraz sprawdzania kolizji. Początkowo opracowano algorytmy PRM i RRT tylko dla manipulatorów o otwartych łańcuchach kinematycznych, wkrótce jednak podjęto próby ich adaptacji dla manipulatorów z pętlami kinematycznymi (manipulatorów równoległych [9, 27, 40], kilku manipulatorów przenoszących wspólnie obiekt [17]). Szczególnie trudnym problemem jest losowe generowanie konfiguracji spełniających więzy wynikające z warunku zamknięcia pętli.

Do planowania skoordynowanych torów ruchu robotów można zaproponować, przedstawione w niniejszym referacie, podejście polegające na spełnieniu ograniczeń i bazujące na kinematyce systemu robotycznego o strukturze pętli kinematycznej. Problem poszukiwania dopuszczalnego toru jest formułowany w postaci zadania wariacyjnego z ustalonymi warunkami końcowymi, polegającego na poszukiwaniu minimum warunkowego pewnego funkcjonału, przy czym na funkcje ze zbioru na którym określony jest funkcjonał są nałożone warunki zwane więzami. Przy formułowaniu tego zadania istotnym założeniem jest osiągalność konfiguracji początkowej i końcowej (ustalone końce krzywej dopuszczalnej). Do rozwiązania tego zadania wykorzystano zmodyfikowaną metodę Ritza, jedną z tzw. metod bezpośrednich rachunku wariacyjnego [10, 16]. Podstawową ideą metod bezpośrednich jest

to, że zadanie wariacyjne rozpatruje się jako graniczny przypadek zadania na ekstremum funkcji o skończonej liczbie zmiennych.

Istotną zaletą tego podejścia jest stosowalność zarówno dla układów z niedomiarem stopni swobody, a także dla układów redundantnych. Podstawowym problemem przy planowaniu ruchu dla systemów z niedomiarem stopni swobody jest znalezienie dopuszczalnego toru ruchu, łączącego konfigurację początkową z zadaną konfiguracją końcową. W przypadku układów redundantnych, tzn. z nadmiarem stopni swobody, mamy do czynienia niejako z problemem odwrotnym do poprzedniego, polegającym mianowicie na wyborze, spośród wielu możliwych, toru spełniającego ograniczenia i minimalizującego dany wskaźnik jakości. Pewnym ograniczeniem stosowalności podejścia wariacyjnego jest duża złożoność obliczeniowa, stąd konieczność stosowania efektywnych algorytmów rozwiązywania zadań optymalizacji nieliniowej.

Wśród metod numerycznych służących do efektywnego rozwiązywania zadań optymalizacji nieliniowej dużej skali najbardziej rozpowszechnione są obecnie algorytmy sekwencyjnego programowania kwadratowego (SQP) oraz metody punktu wewnętrznego dla zadań nieliniowych (NLP). Pod względem efektywności obliczeniowej i odporności metody punktu wewnętrznego lepiej niż metody SQP nadają się do optymalizacji dużych zadań NLP. W publikacjach z dziedziny optymalizacji nieliniowej można spotkać wiele artykułów porównujących wydajność numeryczną solwerów opartych na metodach SQP i metodach punktu wewnętrznego dla zadań nieliniowych przy użyciu metodologii profili efektywności [11]. Warto tu wymienić prace [2, 25] porównujące solwery SNOPT, filterSQP, LOQO i KNITRO, oraz [34, 37], w których porównano solwer IPOPT z solwerami LOQO i KNITRO. Natomiast w [8, 7] porównano efektywność specjalizowanych solwerów SQP korzystających ze struktury zadania i solwera IPOPT dla zadań optymalizacji dynamicznej, dla których rozwiązanie otrzymujemy w wyniku ich transformacji do zadań NLP. W niniejszej pracy do rozwiązania zadania planowania ruchu ściśle współpracujących robotów będzie używany solwer IPOPT<sup>2</sup>, oparty na prymalno-dualnej metodzie punktu wewnętrznego dla zadań nieliniowych [37].

## 2. OPIS KINEMATYCZNY UKŁADU WSPÓŁPRACUJĄCYCH ROBOTÓW

Sformułujmy ogólny opis mechanizmu  $M$  składającego się z sztywnych członów  $L_i$  połączonych nieelastycznymi przegubami  $J_k$ , zawierającego zamknięty łańcuch kinematyczny. Przestrzeń robocza  $W$  mechanizmu  $M$  będzie pewnym podzbiorem przestrzeni fizycznej, którą można utożsamić z przestrzenią euklidesową  $\mathbb{R}^3$ , czyli  $W \subset \mathbb{R}^3$ . Domknięty i ograniczony podzbiór  $O \subset \mathbb{R}^3$  oznacza obiekt. Obiekt może być modelowany na wiele sposobów, np. w postaci zbioru wielościanów wypukłych, zbiorów semi-algebraicznych, zbiorów trójkątów [22]. Używając standardowych metod parametryzacji dla  $M$  [40], możemy wyrazić jego konfigurację jako wektor  $\mathbf{q} = [q_1, \dots, q_n]^T$ . Przy czym, w ogólnym przypadku, poszczególne zmienne konfiguracyjne mogą odpowiadać zarówno przegubom aktywnym (napędzanym), jak też biernym (nie napędzanym, np. gdy przedmiot manipulacji nie jest pojedynczym ciałem sztywnym, lecz składa się dwóch lub więcej elementów połączonych przegubami). Zbiór wszystkich punktów  $\mathbf{q}$  tworzy przestrzeń konfiguracyjną  $\mathcal{C}$ , zatem  $M(\mathbf{q})$  oznacza transformację  $M$  do konfiguracji  $\mathbf{q}$ . Przestrzeń

<sup>2</sup>Kod źródłowy solwera IPOPT w języku C++ dostępny jest pod adresem <http://projects.coin-or.org/Ipopt>

konfiguracyjna mechanizmu zawierającego pętle kinematyczne nie jest gładką rozmaiłością, tak jak to ma miejsce w przypadku swobodnego ciała sztywnego. Więzy ruchu wynikające z warunku zamknięcia pętli powodują, że przestrzeń ta ma skomplikowaną strukturę, która może być reprezentowana przez obiekt matematyczny zwany *rozmaiłością semi-algebraiczną* [31]. Rozmaiłość ta jest definiowana jako zbiór zer wielomianów. Niezależne ograniczenia konfiguracyjne wynikające ze spełnienia warunku zamknięcia pętli mają postać więzów holonomicznych i wyznaczają rozmaiłość konfiguracyjną

$$\mathbb{C}_{cl} = \{\mathbf{q} \mid \mathbf{q} \in \mathbb{C} \wedge \mathbf{f}(\mathbf{q}) = \{f_1(\mathbf{q}) = 0, \dots, f_m(\mathbf{q}) = 0\}\} \quad (1)$$

Łatwo pokazać [22], że jeśli więzy holonomiczne można przekształcić do postaci wielomianowej, wówczas przestrzeń  $\mathbb{C}_{cl}$  może być przedstawiona jako rozmaiłość semi-algebraiczna. Dopuszczalne konfiguracje  $M(\mathbf{q})$  muszą spełniać także warunki bezkolizyjności z przeszkodami w przestrzeni roboczej i unikania kolizji między poszczególnymi członami mechanizmu  $M$ . Jeśli jako  $B_i \subset W, i=1, \dots, n_b$  oznaczymy zbiór przeszkód w przestrzeni roboczej, to *przestrzeń bezkolizyjną* (wolną) można zdefiniować jako

$$\mathbb{C}_f = \left( \bigcup_{i \in [1, n_b]} \{\mathbf{q} \in \mathbb{C} \mid (M(\mathbf{q}) \cap B_i = \emptyset)\} \right) \cup \left( \bigcup_{i, j \in PK} \{\mathbf{q} \in \mathbb{C} \mid (L_i(\mathbf{q}) \cap L_j(\mathbf{q}) = \emptyset)\} \right), \quad (2)$$

gdzie  $PK$  oznacza potencjalne pary kolizyjne  $[i, j]$  ogniw  $L_i(\mathbf{q})$  i  $L_j(\mathbf{q})$  w danej konfiguracji  $(\mathbf{q})$  takie, że  $i \in [1, n_l - 1]$  oraz  $j \in [i + 1, n_l]$ .

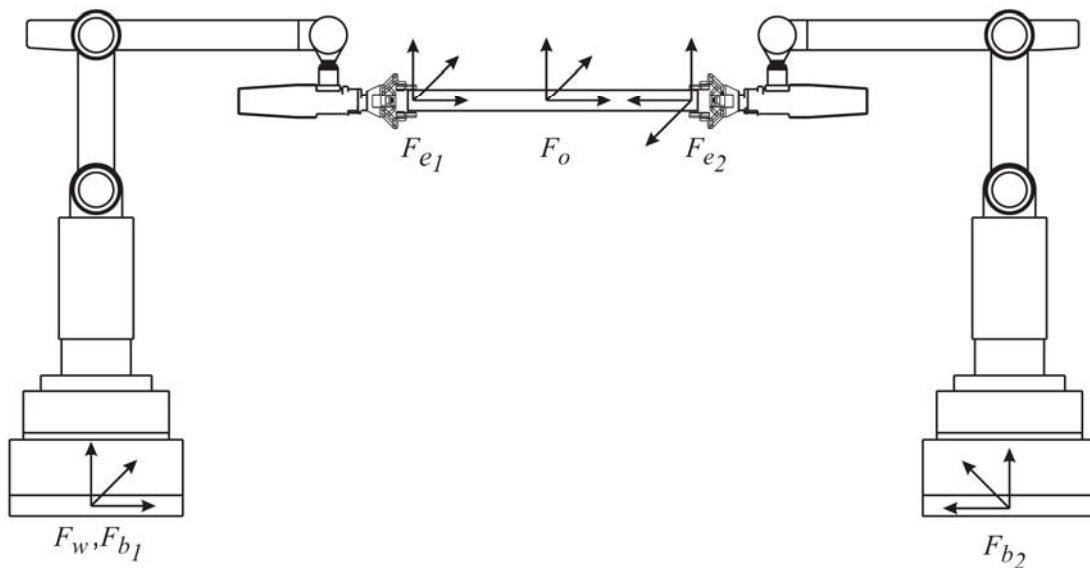
Zbiór konfiguracji spełniających wszystkie ograniczenia geometryczne można przedstawić jako

$$\mathbb{C}_v = \mathbb{C}_{cl} \cap \mathbb{C}_f \quad (3)$$

W ogólnym przypadku,  $\mathbb{C}_v$ , może składać się z rozłącznych podzbiorów (składowych) o skomplikowanej strukturze.

Rozważamy zatem mechanizm  $M$  składający się z dwóch manipulatorów z uchwyconym sztywno obiektem będącym ciałem sztywnym (rys. 1). Niech  $\mathbf{q} = [\mathbf{q}_1^T, \mathbf{q}_2^T]^T$  oznacza wektor współrzędnych konfiguracyjnych tego mechanizmu. Do opisu systemu, z jego wyróżnionymi elementami związane układy współrzędnych kartezjańskich, tak jak pokazano na rys. 1. Są to następujące układy współrzędnych:  $F_w$  – globalny układ odniesienia dla całego systemu,  $F_{b_i}$  – układ bazowy robota  $R^i, i=1, 2$ ,  $F_o$  – układ związany z obiektem oraz  $F_{e_i}$  – układ związany z chwytakiem robota  $R^i$ .

Do opisu przejścia pomiędzy poszczególnymi układami współrzędnych wykorzystamy macierze przekształcenia jednorodnego  $T \in \mathbb{SE}(3)$  o wymiarze  $(4 \times 4)$ , gdzie  $\mathbb{SE}(3)$  oznacza specjalną grupę euklidesową ruchów ciała sztywnego [26]. Pozycje układów bazowych robotów  $F_{b_i}$  względem globalnego układu odniesienia  $F_w$  są określone za pomocą macierzy  $T_{wb_i}, i=1, 2$ . Macierze  $T_{b_i e_i}(\mathbf{q}_i), i=1, 2$ , reprezentują kinematykę prostą  $\mathbf{K}_i$   $i$ -tego łańcucha i opisują pozycję układu  $F_{e_i}$  jako funkcję zmiennych konfiguracyjnych  $\mathbf{q}_i$ , zaś macierz  $T_{e_i o}, i=1, 2$  określa pozycję układu obiektu  $F_o$  względem układu  $F_{e_i}$ .



Rys. 1. Struktura geometryczna układu ściśle współpracujących robotów

Warunki zamknięcia pętli kinematycznej można, w rozważanym przypadku, zapisać w postaci równania macierzowego

$$T_{b_1 e_1}(\mathbf{q}_1) T_{e_1 o} - T_{b_1 b_2} T_{b_2 e_2}(\mathbf{q}_2) T_{e_2 o} = \mathbf{0}, \quad (4)$$

gdzie  $T_{b_1 b_2}$  jest macierze przekształcenia jednorodnego opisującą pozycję układu bazowego  $F_{b_2}$  względem układu bazowego łańcucha  $F_{b_1}$ .

Rozmaitość zadaniową można utożsamiać z zwartym podzbiorem grupy  $\mathbb{SE}(3)$ . Punkt w przestrzeni zadaniowej można wyrazić za pomocą minimalnej liczby zmiennych wprowadzając reprezentacje przestrzeni w postaci układów współrzędnych odpowiednio dla grupy przesunięć  $\mathbb{R}^3$  i grupy obrotów  $\mathbb{SO}(3)$  [29]. Dla opisu przesunięć zazwyczaj stosuje się współrzędne kartezjańskie, natomiast reprezentacji grupy obrotów istnieje wiele, m.in. kąty Eulera, oś-kąt, kwaterniony, itd., [26].

### 3. PLANOWANIE TORU RUCHU

Krzywe geometryczne w przestrzeni konfiguracyjnej będziemy nazywali *torami*, zaś krzywe w przestrzeni zadaniowej *ścieżkami*.

Określmy zatem podstawowe zadanie planowania toru w przestrzeni konfiguracyjnej  $C_v$ , polegającej na wyznaczeniu krzywej  $\mathbf{q}(\cdot)$ , która realizuje odpowiednio gładką ścieżkę (krzywą) w przestrzeni zadaniowej. Dla danej konfiguracji początkowej  $\mathbf{q}_p \in C_v$  i końcowej  $\mathbf{q}_f \in C_v$  należy znaleźć ciągły i dostatecznie gładki tor  $\mathbf{q}(s): [0,1] \rightarrow C_v$  taki, że  $\mathbf{q}(0) = \mathbf{q}_p$  i  $\mathbf{q}(1) = \mathbf{q}_f$ . Parametr  $s$  może być interpretowany jako znormalizowana długość krzywej zakreślonej przez wybrany punkt, np. początek układu  $F_o$ . Warunkiem koniecznym istnienia rozwiązania tego zadania, jest przynależność obu konfiguracji  $\mathbf{q}_p$  i  $\mathbf{q}_f$  do jednej składowej spójnej przestrzeni  $C_v$  [31].



Zadanie poszukiwania torów ruchu w przestrzeni  $C_v$  jest, w ogólnym przypadku, bardzo złożonym problemem obliczeniowym. Struktura tej przestrzeni już dla zamkniętych łańcuchów o kilku stopniach swobody jest bardzo skomplikowana. Chociaż, w teorii, istnieją ogólne algorytmy zupełne planowania torów na rozmaitościach semi-algebraicznych [31], to ze względu na stopień ich złożoności nie zostały one w pełni zaimplementowane dla rzeczywistych systemów robotycznych. Dlatego też w celu znalezienia praktycznego rozwiązania wykorzystano jedną z metod rachunku wariacyjnego.

### 3.1. Sformułowanie zadania planowania toru jako problemu wariacyjnego

W podejściu wariacyjnym można, w ogólnym przypadku, uwzględnić przy planowaniu ruchu dynamikę systemu, jednakże takie zadanie sprowadza się w istocie do zadania sterowania optymalnego [15]. Rozwiązanie kompletnego zadania sterowania optymalnego w rozważanym przypadku jest bardzo trudne i czasochłonne. W wielu przypadkach wystarcza rozwiązanie uproszczonego zadania, w którym uwzględnia się tylko kinematykę systemu robotycznego [28]. W takim przypadku problem planowania można sprowadzić do zadania na poszukiwanie dopuszczalnych funkcji (krzywych geometrycznych), które minimalizują dany funkcjonal kosztu. Dalej będziemy rozpatrywali funkcjonały dające się przedstawić w ogólnej postaci

$$I[s, \mathbf{q}] = \int_0^1 F(s, \mathbf{q}(s)) ds \quad (5)$$

przy ograniczeniach

$$\mathbf{q} \in Y \triangleq \{ \mathbf{q} \in C^2([0,1]) : \mathbf{h}(s, \mathbf{q}) = \mathbf{0} \wedge \mathbf{g}(s, \mathbf{q}) \leq \mathbf{0} \wedge \mathbf{q}(0) = \mathbf{q}_p, \mathbf{q}(1) = \mathbf{q}_f \}, \quad (6)$$

Funkcjonał (5) rozpatrujemy nie na wszystkich krzywych spełniających warunki graniczne  $\mathbf{q}_p, \mathbf{q}_f$ , ale tylko na tych z nich, które leżą na pewnej  $(n - m_h)$ -wymiarowej rozmaitości [16], gdzie  $m_h$  jest liczbą ograniczeń równościowych. Więzy jednostronne  $\mathbf{g}(s, \mathbf{q}) \leq \mathbf{0}$  odpowiadają ograniczeniom zakresów ruchów przegubów i unikania kolizji między obiektem i ramionami manipulatorów oraz przeszkodami. Jak łatwo zauważyć, podane ograniczenia zbioru rozwiązań dopuszczalnych w istotny sposób wpływają na charakter rozwiązania optymalnego.

W celu sformułowania zadania planowania dopuszczalnego toru ruchu w postaci zadania wariacyjnego należy określić konkretną postać funkcjonału (5). Niech  $\mathbf{p}^d(\cdot)$  będzie zadaną ścieżką obiektu w przestrzeni zadaniowej łączącą pozycję początkową  $\mathbf{p}_p$  i zadaną pozycję końcową  $\mathbf{p}_f$ . Załóżmy, że ścieżka  $\mathbf{p}^d(\cdot)$  została narzucona arbitralnie bez uwzględniania czy jest ona dopuszczalna (poza warunkami granicznymi, które musi spełniać). Dla systemu przedstawionego na rys. 1 problem planowania można sprowadzić do zadania minimalizacji funkcjonału  $I[\mathbf{q}(\cdot)]$  postaci:

$$I[\mathbf{q}(\cdot)] = \int_0^1 (\mathbf{p}_i(\mathbf{q}_i(s)) - \mathbf{p}^d(s))^T W (\mathbf{p}_i(\mathbf{q}_i(s)) - \mathbf{p}^d(s)) ds, \quad (7)$$

gdzie  $\mathbf{p}_i(\cdot), i = 1 \vee i = 2$  określają aktualną ścieżkę dla układu związanego z obiektem jako funkcji współrzędnych konfiguracyjnych jednego z manipulatorów (odpowiednio manipulatora  $i = 1 \vee i = 2$ ). Wektory  $\mathbf{p}_i$  opisują pozycję obiektu w układzie odniesienia  $F_w$  przy odpowiedniej parametryzacji współrzędnych położenia i orientacji. Macierz wagowa  $W$

jest dobierana stosowanie do postawionego zadania i określa, które współrzędne obiektu będą lepiej „nadażały” za ścieżką zadaną. Ograniczenia równościowe (4) przy przyjętej parametryzacji można zapisać jako

$$\forall s \in [0,1] \quad h_k(s, \mathbf{q}) = p_{1k}(\mathbf{q}_1(s)) - p_{2k}(\mathbf{q}_2(s)) = 0, \quad k = 1, \dots, 6 \quad (8)$$

Klasyczne rozwiązanie zadania wariacyjnego (5)–(6) polega na wprowadzeniu wewnętrznych funkcjonałów kary i rozwiązaniu (o ile takie rozwiązanie istnieje) układu równań Eulera-Poissona opisującym warunki konieczne istnienia ekstremum takiego funkcjonału [16, 15]. W ogólnym przypadku równania Eulera-Poissona są równaniami różniczkowymi i znajdowanie rozwiązania zadania wariacyjnego sprowadza się do zagadnienia istnienia i poszukiwania rozwiązania tych równań. Zauważmy jednak, że w naszym przypadku rozważamy pewną szczególną klasę funkcjonałów, dla których funkcja podcałkowa nie zależy od pochodnych  $q'_k = \partial q_k / \partial s$  (równania Eulera-Poissona nie są tutaj równaniami różniczkowymi). Rozwiązanie takiego układu równań nie zawiera stałych dowolnych i dlatego, w ogólnym przypadku, może nie spełniać warunków granicznych  $\mathbf{q}(0) = \mathbf{q}_p$  i  $\mathbf{q}(1) = \mathbf{q}_f$ .

Do najbardziej ogólnych, a jednocześnie efektywnych obliczeniowo, metod przybliżonego rozwiązywania zadań wariacyjnych z ograniczeniami zalicza się tzw. metody bezpośrednie. Podstawową ideą metod bezpośrednich jest to, że zadanie wariacyjne rozpatruje się jako graniczny przypadek zadania na ekstremum funkcji o skończonej liczbie zmiennych. W niniejszej pracy wykorzystano jedną z takich metod zwaną metodą Ritza [16].

### 3.2 Aproksymacja skończenie-wymiarowa zadania wariacyjnego

Idea proponowanego podejścia polega na poszukiwaniu rozwiązań zadania wariacyjnego nie na wszystkich dopuszczalnych<sup>3</sup> krzywych  $\mathbf{q}$  lecz tylko wśród  $N+1$  pierwszych funkcji z ciągu  $\{\varphi_j\}_{j=0}^{\infty}$

$$\mathbf{q}(\cdot) = \sum_{j=0}^N \mathbf{c}_j \varphi_j(\cdot), \quad \dim \mathbf{q} = \dim \mathbf{c}_j, \quad (9)$$

gdzie  $\mathbf{c}_j$  jest wektorem stałych współczynników, zaś funkcje  $\mathbf{q}(\cdot)$  leżą w zbiorze określoności danego funkcjonału. Ciąg funkcji bazowych  $\varphi_j$  nie może być wybrany w sposób dowolny. Wybór konkretnej bazy jest uwarunkowany wymaganym charakterem rozwiązania. Dokładność aproksymacji zależy od wymiaru  $N$  podprzestrzeni  $Y_N \subset Y$  i od postaci funkcji bazowych  $\varphi_j$ .

Zakładamy, że funkcje  $\varphi_j$  są klasy  $C^2$  i ciąg jest zupełny w sensie normy przestrzeni  $Y_N$ .

Niech  $\Delta_N$  oznacza podział przedziału  $[0,1] \subset \mathbb{R}^1$  na  $N$  równych podprzedziałów

$\Delta s = s_{j+1} - s_j = \frac{1}{N}$  dla  $s_j, j = 0, 1, \dots, N-1$ , gdzie  $s_j$  są takimi wartościami, że

$0 = s_0 < s_1 < \dots < s_N = 1$ . W rozważanym przypadku jako funkcje bazowe  $\varphi_j$  wybrano B-funkcje sklepane, które można zdefiniować rekurencyjnie jako [4]:

<sup>3</sup>Dopuszczalność oznacza tutaj tylko, że funkcje należą do dziedziny funkcjonału, ale nie muszą spełniać ograniczeń.

Stała B-funkcja sklejana w  $j$ -tym podprzedziale jest dana w postaci:

$$b_{j,0}(s) = \begin{cases} 1 & s \in [s_j, s_{j+1}] \\ 0 & \text{pozostało } s \end{cases} \quad (10)$$

B-funkcja sklejana stopnia  $m$  w przedziale  $[s_j, s_{j+m+1}]$  jest zdefiniowana jako

$$b_{j,m}(s) = \frac{s - s_j}{s_{j+m} - s_j} b_{j,m-1}(s) + \frac{s_{j+m+1} - s}{s_{j+m+1} - s_{j+1}} b_{j+1,m-1}(s) \quad (11)$$

Oznaczmy jako  $B^2$  przestrzeń B-funkcji sklejanych składających się z wielomianów trzeciego stopnia określonych zgodnie z powyższą definicją. Istotną cechą B-funkcji sklejanych jest możliwość *lokalnej modyfikacji* krzywej.

Problem poszukiwania dopuszczalnego toru ruchu dla zespołu robotów może być sprowadzony do zadania znalezienia minimum funkcji  $(N+3) \times n$  zmiennych. Bez utraty ogólności można przyjąć macierz wagową w postaci  $W = \text{diag}(w_1, \dots, w_6)$ . Funkcjonał (7)

można wyrazić jako funkcję  $\mathbf{c} = [\mathbf{c}_1^T, \mathbf{c}_2^T]^T$  ( $\mathbf{c}_i \in \mathbb{R}^{n_i \times (N+3)}$ ,  $i = 1, 2$ ), współczynników rozwinięcia  $\mathbf{q}(\cdot)$  w bazie funkcyjnej  $\{\varphi_j\}_{j=-1}^{N+1}$

$$\min_{\mathbf{c}} \sum_{j=0}^{N-1} \int_{s_j}^{s_{j+1}} \sum_{k=1}^6 w_k (p_{1k} (\sum_{i=j-1}^{j+2} \mathbf{c}_i \varphi_i(s)) - p_k^d(s))^2 ds \quad (12)$$

Poszukujemy zatem minimum funkcji (12) względem  $\mathbf{c}$ , przy ograniczeniach

$$\mathbf{c} \in Y_N = \left\{ \mathbf{c} \in \mathbb{R}^{n \times (N+3)} : \mathbf{h}(s_j, \mathbf{c}) = 0 \wedge \mathbf{g}(s_j, \mathbf{c}) \leq 0, j = 0, \dots, N \right\} \quad (13)$$

Tak otrzymane zadanie programowania nieliniowego może być rozwiązywane numerycznie za pomocą odpowiednich algorytmów optymalizacji nieliniowej z ograniczeniami, w szczególności za pomocą solverów bazujących na metodzie punktu wewnętrznego dla zadań nieliniowych.

#### 4. METODY PUNKTU WEWNĘTRZNEGO DLA ZADAŃ PROGRAMOWANIA NIELINIOWEGO

Metody punktu wewnętrznego dla zadań programowania nieliniowego (NLP), nazywane też metodami barierowymi, powstały z potrzeby efektywnego rozwiązywania dużych zadań optymalizacji. W szczególności dla zadań NLP z dużą liczbą ograniczeń nierównościowych, metody te stanowią poważną alternatywę wobec strategii zbiorów aktywnych. W ciągu ostatnich 15 lat doprowadzono do lepszego zrozumienia zbieżności metod punktu wewnętrznego oraz rozwinięto dla nich efektywne algorytmy obliczeniowe charakteryzujące się pożądaną zbieżnością globalną i lokalną.

Termin *metoda punktu wewnętrznego* został po raz pierwszy użyty przez Fiacco i McCormicka w 1968 roku w książce [12], do nazwania dowolnego algorytmu, który służy do wyznaczenia minimum lokalnego zadania NLP za pomocą rozwiązania określonej sekwencji zadań minimalizacji *bez ograniczeń*. Definicja taka ewoluowała do postaci, w której za metodę IP (ang. *Interior-Point*) uważamy dowolny algorytm rozwiązujący zestaw zadań optymalizacji stowarzyszony ze zmniejszaniem wartości mnożnika  $\mu$ , którego celem jest znalezienie lokalnych rozwiązań leżących we wnętrzu obszaru dopuszczalnego wyznaczonego przez ograniczenia nierównościowe zadania NLP.



Celem zapewniania zbieżności z „niedobrych” punktów początkowych, dla metod punktu wewnętrznego, zarówno w wersji z użyciem obszaru zaufania, jak i minimalizacją kierunkową, opracowano technikę funkcji oceny, bazującą na dokładnej funkcji kary i gwarantującą zbieżność do rozwiązania [5, 30]. Z drugiej strony, Fletcher i Leyffer [14] zaproponowali ostatnio metody z filtrem, jako alternatywę wobec funkcji oceny, gwarantującą globalną zbieżność dla algorytmów programowania nieliniowego. Opierają się one na pomysłach akceptacji punktów generowanych przez algorytm optymalizacji w przypadku, gdy poprawiają one wartość funkcji celu *lub* poprawiają wartość przekroczenia ograniczeń, zamiast kombinacji obu tych miar, zdefiniowanej przez funkcję oceny.

Ostatnio, koncepcję filtra zaadaptowano do metod barierowych. W artykule [32] autorzy rozważają metodę obszaru zaufania z filtrem, w której akceptuje się kolejne iteracje rozwiązania na podstawie normy warunków optymalności. Z kolei w pracy [3] zaproponowano szereg heurystyk korzystających z koncepcji metod z filtrem, dla których uzyskano poprawę efektywności w porównaniu do metody z funkcją oceny. Ostatecznie, w artykule [36] przedstawiono analizę zbieżności globalnej dla algorytmu punktu wewnętrznego z metodą minimalizacji kierunkowej opartej na technice filtra.

Metody punktu wewnętrznego dla zadań NLP doczekały się implementacji w wielu solverach optymalizacyjnych, takich jak LOQO [33], KNITRO [6, 38], czy IPOPT [34, 37]. W testach numerycznych solwery te okazały się efektywne i odporne dla wielu dużych zadań NLP.

#### 4.1. Solver IPOPT

W rozdziale tym zostanie opisany prymalno-dualny algorytm punktu wewnętrznego z metodą minimalizacji w kierunku opartą na technice filtra, użyty do implementacji solvera IPOPT. Jego twórcy przyjęli następujące sformułowanie wyjściowego zadania NLP:

$$(P) \quad \min_{y \in \mathbb{R}^{m_y}} \{J(y) \mid h(y) = 0, y \geq 0\} \quad (14)$$

Założono, że funkcja celu  $J: \mathbb{R}^{m_y} \mapsto \mathbb{R}$  i ograniczenie równościowe  $h: \mathbb{R}^{m_y} \mapsto \mathbb{R}^{m_h}$  ( $m_h < m_y$ ) mają ciągłe drugie pochodne. Zadania z ograniczeniami nierównościami typu  $g(y) \geq 0$  mogą zostać przeformułowane do powyższej postaci przez wprowadzenie zmiennych pomocniczych, tj.  $g(y) - s = 0, s \geq 0$ .

Algorytm barierowy w solverze IPOPT bazuje na zastąpieniu ograniczeń kostkowych na zmienne  $y \geq 0$  dodatkowym składnikiem funkcji celu barierą logarytmiczną:

$$(P_\mu) \quad \min_{y \in \mathbb{R}^{m_y}} \left\{ J_\mu(y) = J(y) - \mu \sum_{j=1}^{m_y} \ln(y^j) \mid h(y) = 0 \right\}, \quad (15)$$

gdzie  $\mu > 0$  parametr barierowy, a  $y^j$  oznacza  $j$ -ty element wektora  $y$ . Ponieważ funkcja celu dla zadania  $(P_\mu)$  staje się dowolnie duża, gdy  $y^j$  osiąga jedno ze swoich ograniczeń, zatem rozwiązanie lokalne  $y_*(\mu)$  tego zadania znajduje się we wnętrzu zbioru wyznaczonego przez ograniczenia  $y_*(\mu) > 0$ . Stopień wpływu bariery jest określony przez wielkość parametru  $\mu$ , i przy pewnych założeniach, gdy  $\mu \rightarrow 0$ , rozwiązanie  $y_*(\mu)$  jest zbieżne do rozwiązania lokalnego  $y_*$  zadania wyjściowego (P). W rezultacie algorytm poszukiwania rozwiązania wyjściowego zadania (P) opiera się na rozwiązaniu sekwencji zadań barierowych

$(P_\mu)$ , dla zmniejszających się wartości parametru  $\{\mu_l\} \rightarrow 0$ , gdzie  $l$  jest licznikiem kolejnych podproblemów barierowych.

Algorytm punktu wewnętrznego solwera IPOPT znajduje rozwiązanie prymalno-dualnych warunków stacjonarności dla zadania  $(P_\mu)$ , sformułowanych w postaci następującego układu równań nieliniowych:

$$\nabla_y J(y) + \nabla_y h(y)\lambda - z = 0, \tag{16}$$

$$h(y) = 0, \tag{17}$$

$$YZ - \mu e_y = 0, \tag{18}$$

gdzie  $Y$  i  $Z$  to macierze diagonalne o elementach odpowiednio  $y$  i  $z$ ,  $e_y$  jest wektorem jednostkowym o długości  $m_y$ ,  $\lambda \in \mathbb{R}^{m_h}$  to wektor mnożników Lagrange'a dla ograniczeń równościowych, a  $z \in \mathbb{R}^{m_y}$  to wektor mnożników Lagrange'a dla ograniczeń kostkowych w granicy przy  $\mu \rightarrow 0$ . Zwróćmy uwagę, że układ równań (16)–(18) dla  $\mu = 0$  z dodatkowym warunkiem  $y, z \geq 0$  daje warunki optymalności KKT dla pierwotnego zadania (P).

Do rozwiązania układu równań (16)–(18) dla ustalonej wartości parametru  $\mu$  używamy iteracyjnej metody Newtona, polegającej na rozwiązaniu liniowego układu równań:

$$\begin{bmatrix} W_k & \nabla_y h(y_k) & -I \\ \nabla_y h(y_k)^T & 0 & 0 \\ Z_k & 0 & Y_k \end{bmatrix} \begin{pmatrix} d_k^y \\ d_k^\lambda \\ d_k^z \end{pmatrix} = - \begin{pmatrix} \nabla_y J(y_k) + \nabla_y h(y_k)\lambda_k - z_k \\ h(y_k) \\ Y_k Z_k - \mu e_y \end{pmatrix}, \tag{19}$$

gdzie  $W_k$  oznacza dokładny hesjan funkcji Lagrange'a dla zadania wyjściowego (P):

$$W_k = \nabla_{yy} J(y_k) + \sum_{i=1}^{m_h} \lambda_k^i \nabla_{yy} h^i(y_k) \tag{20}$$

lub też jego aproksymację; funkcja Lagrange'a ma postać:

$$L(y, \lambda, z) := J(y) + h(y)^T \lambda - z. \tag{21}$$

Indeks  $k$  oznacza tu licznik wewnętrznych iteracji dla metody Newtona, wektor  $(y_k, \lambda_k, z_k)$  jest aktualną iteracją rozwiązania i mnożników Lagrange'a spełniającą warunek  $y_k, z_k > 0$ , zaś  $(d_k^y, d_k^\lambda, d_k^z)$  to szukane rozwiązanie nowy kierunek poszukiwań.

W solwerze IPOPT zamiast rozwiązywać bezpośrednio niesymetryczny układ równań (19) szuka się równoważnego rozwiązania najpierw rozwiązując symetryczny układ równań o mniejszym rozmiarze:

$$\begin{bmatrix} W_k + \Sigma_k & \nabla_y h(y_k) \\ \nabla_y h(y_k)^T & 0 \end{bmatrix} \begin{pmatrix} d_k^y \\ \lambda_k^+ \end{pmatrix} = - \begin{pmatrix} \nabla_y J_\mu(y_k) \\ h(y_k) \end{pmatrix}, \tag{22}$$

gdzie  $\Sigma_k := Y_k^{-1} Z_k$ . Równania dla układu (22) wyprowadza się z równań układu (19) przez eliminację ostatniego wiersza blokowego. Następnie wyznacza się kierunek  $d_k^\lambda$  z równania:

$$d_k^\lambda = \lambda_k^+ - \lambda_k \tag{23}$$

oraz kierunek  $d_k^z$  z równania:

$$d_k^z = \mu Y_k^{-1} e_y - z_k - \Sigma_k d_k^y. \quad (24)$$

Po obliczeniu kierunków poszukiwań z zależności (22)–(24), nową iterację wektora zmiennych wyznaczamy następująco:

$$(y_{k+1}, \lambda_{k+1}, z_{k+1}) := (y_k, \lambda_k, z_k) + (\alpha_k d_k^y, \alpha_k d_k^\lambda, \alpha_k d_k^z), \quad (25)$$

gdzie  $\alpha, \alpha_k^z \in (0,1]$  to długości kroków, przy czym jak widać dla zmiennych  $z$  przewidziano osobną długość kroku.

Ponieważ wiadomo, że w punkcie optymalnym problemu barierowego ( $P_\mu$ ) zmienne  $y$  i  $z$  muszą być dodatnie, solver IPOPT stara się zachować tę właściwość dla wszystkich iteracji w rezultacie stosowana jest następująca reguła doboru długości kroku:

$$\bar{\alpha}_k := \max\{\alpha \in (0,1]: y_k + \alpha d_k^y \geq (1-\tau)y_k\}, \quad (26)$$

$$\bar{\alpha}_k^z := \max\{\alpha \in (0,1]: z_k + \alpha d_k^z \geq (1-\tau)z_k\}, \quad (27)$$

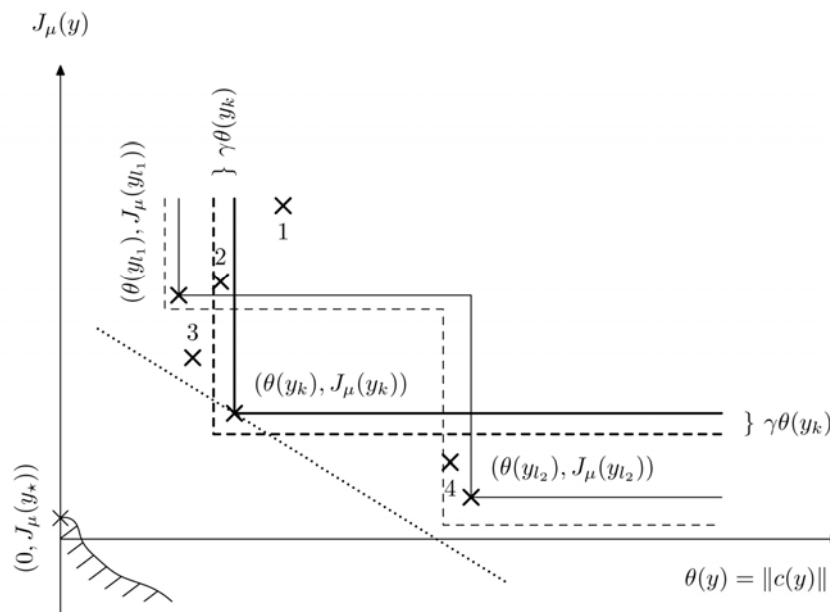
dla parametru  $\tau \in (0,1)$ , zwykle bliskiemu 1 (np.  $\tau = 0.995$ ). Dla zmiennych  $z$  długość kroku wybieramy jako  $\alpha_k^z := \bar{\alpha}_k^z$ , zaś długość kroku  $\alpha_k \in (0, \bar{\alpha}_k]$  dla pozostałych zmiennych jest wyznaczana w wyniku zastosowania procedury minimalizacji kierunkowej z powrotami, wykorzystującej malejący ciąg próbnych długości kroków,  $\alpha_{k,l} = 2^{-l} \bar{\alpha}_k$ ,  $l = 0,1,2,\dots$  używany jest tu wariant metody z filtrem typu Fletchera i Leyffer'a [14], co zapewnia globalną zbieżność algorytmu punktu wewnętrznego w kierunku rozwiązania ( $P_\mu$ ).

Metody minimalizacji z filtrem opierają się na pomysłach optymalizacji dwukryterialnej, w której oprócz minimalizacji funkcji barierowej  $J_\mu(y)$ , chcemy minimalizować funkcję przekroczenia ograniczeń  $\theta(y) := \|h(y)\|$ , tak aby zapewnić zbieżność do punktu dopuszczalnego. Na rysunku 2 przedstawiono rzut przestrzeni  $\mathbb{R}^m$  na płaszczyznę  $(\theta(y), J_\mu(y))$ . Każdy punkt z pierwotnej przestrzeni zmiennych decyzyjnych, taki jak rozwiązanie optymalne  $y_*$  lub aktualna iteracja  $y_k$ , ma swój odpowiednik na tym rysunku, np.  $(\theta(y_k), J_\mu(y_k))$ . Decyzja o przyjęciu punktu testowego  $y_k + \alpha_{k,l} d_k^y$  jako następnej iteracji rozwiązania  $y_{k+1}$  zależy od tego, czy gwarantuje on wystarczającą poprawę wartości jednej z miar:  $\theta$  lub  $J_\mu$ , w porównaniu do ich wartości w punkcie  $y_k$ . Na przykładzie z rysunku 2, punkt testowy "1" nie został zaakceptowany, gdyż pogarsza on wartości obydwu miar. Także punkt "2" powinien zostać odrzucony, gdyż nie zmniejsza on wystarczającym stopniu wartości miary przekroczenia ograniczeń (na rysunku wystarczające wartości poprawy wartości miar wyznaczone są przez linie przerywane, których punkt przecięcia leży blisko punktu  $(\theta(y_k), J_\mu(y_k))$ ). Punkt testowy "3" powinien zostać przyjęty.

Do tej stosunkowo prostej procedury wyboru następnej iteracji rozwiązania w solverze IPOPT dodano następujące zabezpieczenia:

- W przypadku, gdy bieżąca iteracja rozwiązania spełnia (prawie) ograniczenia, lecz nie jest w wystarczającym stopniu „optymalna”, opisany powyżej warunek odpowiedniego zmniejszenia się jednej z miar dla  $y_k$ , jest zastępowany warunkiem wystarczającego zmniejszenia wyłącznie wartości funkcji barierowej  $J_\mu$ .

- Aby zapobiegać tworzeniu się cykli, pary  $(\theta, J_\mu)$  odpowiadające poprzednim iteracją i tworzące swego rodzaju kopertę (w naszym przykładzie są to iteracje  $y_{l_1}$  i  $y_{l_2}$ ) są dodawane do *filtra*; punkt testowy jest przyjmowany tylko w przypadku jeśli gwarantuje on wystarczającą poprawę wartości jednej z miar w stosunku do wszystkich tych punktów. W naszym przykładzie, punkt testowy "4" zostanie odrzucony, gdyż nie daje on wystarczającej poprawy wartości żadnej obydwu z miar  $\theta$  i  $J_\mu$  w stosunku do punktu  $y_{l_2}$ .
- Może się zdarzyć, że nie ma takiej wartości długości kroku  $\alpha_{k,l}$ , która dawałaby akceptowalny punkt testowy. Po wykryciu takiej sytuacji, algorytm przełącza się na fazę przywrócenia dopuszczalności, której celem jest wyłącznie minimalizacja miary niespełnienia ograniczeń (przy pominięciu minimalizacji pierwotnej funkcji celu), tak długo dopóki nie zostanie znaleziona nowa dopuszczalna iteracja rozwiązania, lub nie można już dłużej redukować miary niespełnienia ograniczeń, np. w przypadku, gdy zadanie ( $P$ ) jest (lokalnie) sprzeczne.



Rys. 2. Metoda minimalizacji w kierunku z filtrem

Formalny opis i analizę procedury minimalizacji kierunkowej z filtrem zaimplementowanej w solwerze IPOPT można znaleźć w [36]. Metoda z filtrem w porównaniu do tradycyjnych algorytmów minimalizacji kierunkowej, takich jak metoda pojedynczej funkcji oceny, jest zazwyczaj mniej konserwatywna i pozwalana na większe długości kroków (na rysunku 2 dla metody dokładnej funkcji kary iteracje musiałyby leżeć pod kropkowaną prostą). Ponadto zabezpieczenie w postaci fazy przywracania dopuszczalności czyni algorytm z filtrem odpornym na błędy, takie jak opisane w [35].

Najbardziej pracochłonną obliczeniowo część solwera IPOPT (oprócz obliczania wartości funkcji celu i ograniczeń oraz ich pochodnych) stanowi rozwiązanie układu równań liniowych (22) najczęściej dużego rozmiaru, a dla zadań optymalizacji dynamicznej bardzo rzadkiego. Do jego faktoryzacji i rozwiązania w solwerze IPOPT używa się zewnętrznych solwerów dla rzadkich układów równań liniowych, takich jak MA27, MA57, WSMP, PARDISO i MUMPS.

## 5. PRZYKŁAD NUMERYCZNY

W celu weryfikacji przydatności proponowanego algorytmu planowania dopuszczalnych ścieżek rozpatrzmy zadanie polegające na przeniesieniu sztywnej belki wspólnie przez dwa manipulatory robotów IRp-6, każdy o pięciu stopniach swobody. Parametry manipulatora IRp-6 w notacji Denavita-Hartenberga oraz zakresy dla poszczególnych współrzędnych zawiera tabela 1. Kinematykę prostą  $i$ -tego manipulatora przedstawia równanie (28).

Przegub $i$	$\alpha_{i-1}$ [rad]	$a_{i-1}$ [m]	$d_i$ [m]	$[\theta_i^{min}, \theta_i^{max}]$ [rad]
1	0	0	0.7	$[-\frac{17\pi}{18}; \frac{17\pi}{18}]$
2	$-\pi/2$	0	0	$[-\frac{13\pi}{18}; -\frac{5\pi}{18}]$
3	0	0.45	0	$[-\frac{25\pi}{180}; \frac{4\pi}{18}]$
4	0	0.67	0	$[-\pi/2; +\pi/2]$
5	$-\pi/2$	0	0	$[-\pi; +\pi]$
6			0.1	

Tabela 1. Parametry kinematyczne i zakresy ruchu ogniów manipulatora robota IRp-6

$$\mathbf{K}_i = \begin{bmatrix} c_{i1}s_{i4}c_{i5} + s_{i1}s_{i5} & -c_{i1}s_{i4}s_{i5} + s_{i1}c_{i5} & c_{i1}c_{i4} & c_{i1}(d_{i6}c_{i4} + a_{i3}c_{i3} + a_{i2}c_{i2}) \\ s_{i1}s_{i4}c_{i5} - c_{i1}s_{i5} & -s_{i1}s_{i4}s_{i5} - c_{i1}c_{i5} & s_{i1}c_{i4} & s_{i1}(d_{i6}c_{i4} + a_{i3}c_{i3} + a_{i2}c_{i2}) \\ c_{i4}c_{i5} & -c_{i4}s_{i5} & -s_{i4} & d_{i1} - d_{i6}s_{i4} - a_{i3}s_{i3} - a_{i2}s_{i2} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (28)$$

gdzie  $s_{ij} = \sin(\theta_{ij})$  oraz  $c_{ij} = \cos(\theta_{ij})$ ,  $i=1,2$ ,  $j=1, \dots, 5$ . Kąty  $\theta_{ij}$  to zmienne wewnętrzne opisujące położenie poszczególnych ogniów dwóch manipulatorów robotów IRp-6.

Przyjęto, że globalny układ odniesienia  $F_w$  pokrywa się z układem bazowym  $F_{b_1}$ , zaś pozycja układu bazowego drugiego robota jest wyrażona macierzą

$$T_{wb_2} = \begin{bmatrix} -1 & 0 & 0 & l_0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

gdzie  $l_0 = 1.8[m]$ . Pozycje układu związanego z obiektem  $F_o$  względem układów związanych z chwytakami robotów  $F_{e_i}$  są określone za pomocą macierzy

$$T_{e_1^o} = \begin{bmatrix} 1 & 0 & 0 & l_1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_{e_2^o} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & l_2 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$



gdzie  $l_1 = l_2 = 0.2[m]$  odległości między początkiem układu  $F_o$  a początkami układów odpowiednio  $F_{e_1}$  oraz  $F_{e_2}$ . Ścieżka zadana  $\mathbf{p}^d(\cdot)$  dla obiektu (układ  $F_o$ ) opisana w układzie  $F_o$  ma postać linii prostej (liniowa zmiana współrzędnych położenia i orientacji przy wybranej parametryzacji)

$$\mathbf{p}^d(s) = \mathbf{p}^d(0) + s(\mathbf{p}^d(1) - \mathbf{p}^d(0)), \quad s \in [0,1]$$

Zakładamy, że zadana pozycja początkowa obiektu  $\mathbf{p}^d(0) = [0.9, 0, 0.9, 0, 0, 0]^T$  i odpowiadająca jej konfiguracja początkowa obu manipulatorów

$$\mathbf{q}^p = [0.0, -1.4674, 0.2221, \pi/2, 0.0, 0.0, -1.6193, 0.3815, 0.0, 0.0]^T$$

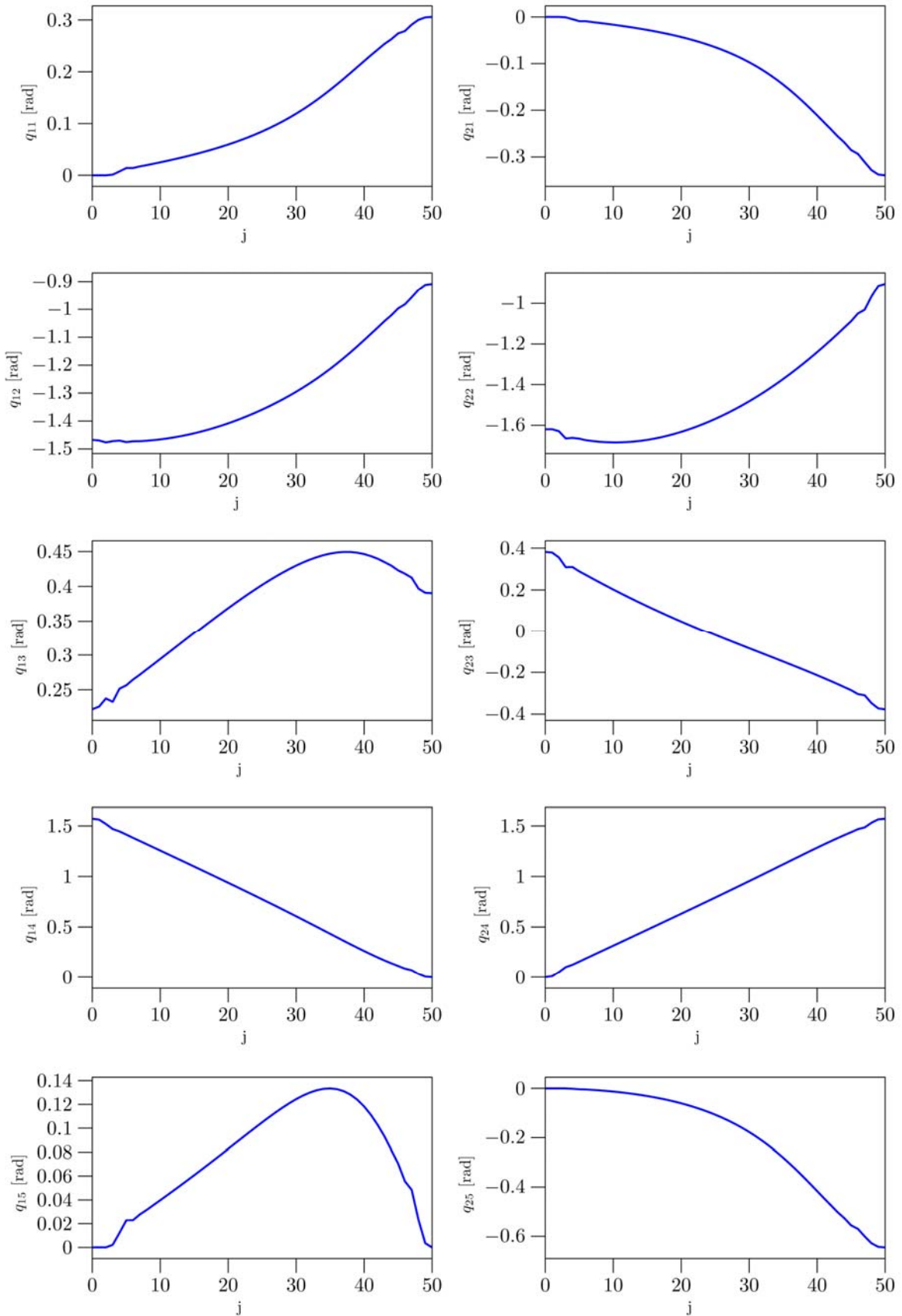
jest dopuszczalna. Podobnie pozycja końcowa  $\mathbf{p}^d(1) = [0.95, 0.3, 1.0, 0.3059, \pi/2, 0]^T$  oraz odpowiadająca jej konfiguracja końcowa manipulatorów

$$\mathbf{q}^f = [0.3059, -0.9087, 0.3903, 0.0, 0.0, -0.3393, -0.9039, -0.3766, \pi/2, -0.6451]^T$$

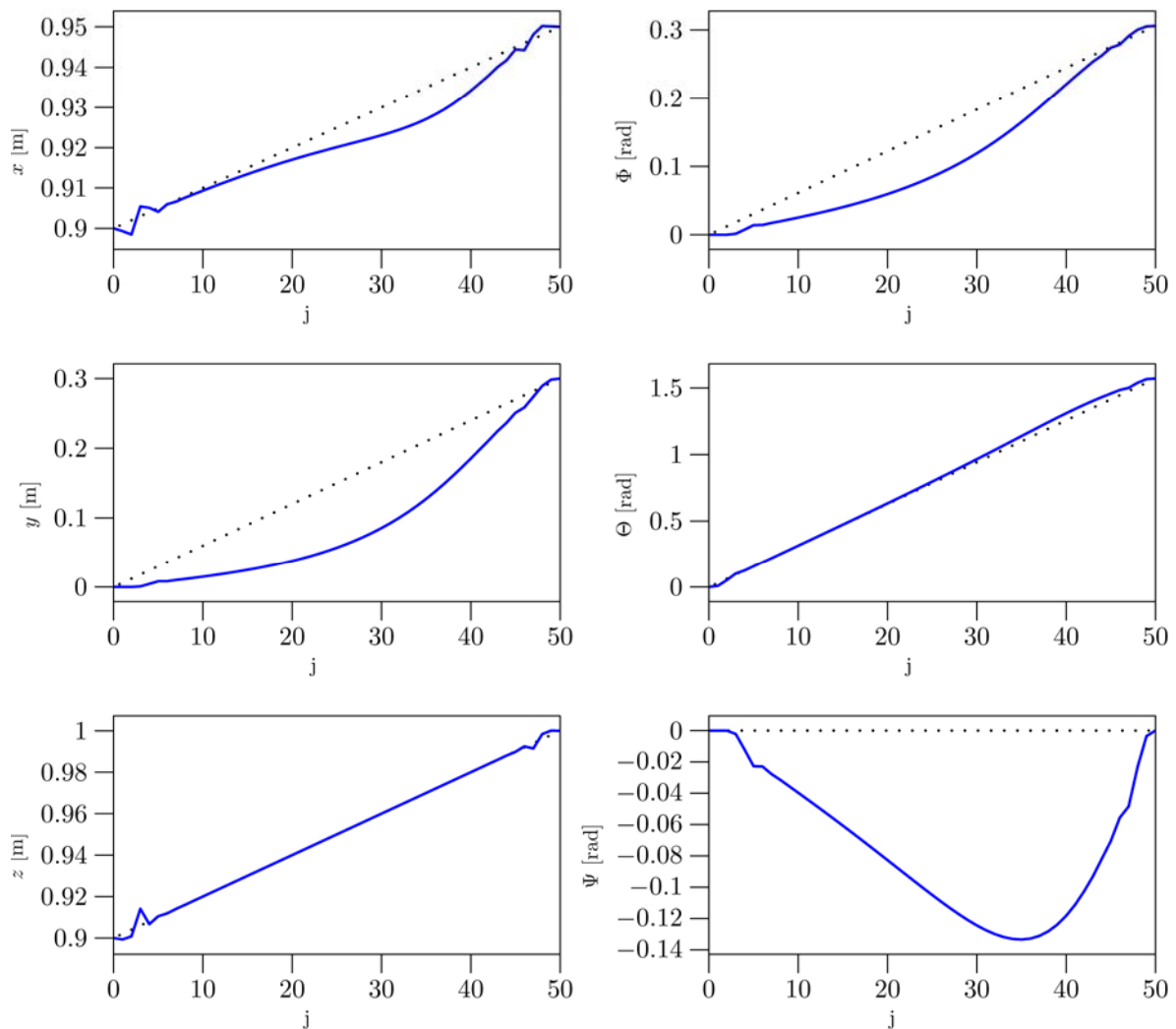
są dopuszczalne.

Rozpatrywane zadanie optymalizacji nieliniowej dla systemu dwóch współpracujących robotów zaimplementowano w języku C++ z użyciem mechanizmu klas i ich dziedziczenia, przy czym do obsługi operacji wektorowo-macierzowych użyto biblioteki uBLAS w szczególności zaimplementowano klasy do obsługi wektorów i macierzy o stałym rozmiarze oraz klasę do obsługi macierzy przekształcenia jednorodnego opartą na implementacji dostępnej w systemie MRROC++ [42] (zawiera ona m.in. metody do konwersji macierzy jednorodnych na współrzędne kartezjańskie XYZ oraz kąty Eulera Z-Y-Z lub Z-Y-X). Do rozwiązania wynikowego zadania NLP użyto solwera IPOPT korzystającego z drugich pochodnych funkcji celu i ograniczeń, co znacząco wpływa na szybkość zbieżności do rozwiązania lokalnego. Solwer IPOPT skonfigurowano do używania solwera macierzowego MUMPS, ponadto zmieniono standardowe ustawienia kilku parametrów solwera mające decydujący wpływ na szybkość zbieżności. Jako punkt startowy dla solwera IPOPT użyto wartości współczynników  $\mathbf{c}$  otrzymanych w wyniku rozwiązania pomocniczego układu równań liniowych dla uprzednio wyznaczonej dopuszczalnej ścieżki początkowej. Jej wyznaczenie polega na iteracyjnym poszukiwaniu kolejnych jej punktów rozwiązując zadanie optymalizacji dla pojedynczego punktu, a nie dla całej ścieżki. Oczywiście nie ma gwarancji, że otrzymany ciąg punktów da ciągłą ścieżkę początkową, ale jest to zbiór punktów dopuszczalnych. Aby uniknąć wyznaczania i wprowadzania skomplikowanych wzorów analitycznych na pierwsze i drugie pochodne funkcji, skorzystano z biblioteki automatycznego różniczkowania CppAD<sup>4</sup> [1], która w najnowszych wersjach posiada możliwość współpracy z solwerem IPOPT. W rozważanym przykładzie obliczeniowym wykorzystano jako funkcje bazowe B-funkcje sklepane trzeciego stopnia. Przedział  $[0,;1]$  został podzielony na  $N = 50$  równych podprzedziałów o długości  $\Delta s = 0.02$ . Ponieważ punkty początkowe i końcowe trajektorii robotów są ustalone należy wyznaczyć dla nich wartości współczynników  $c_{i,-1}, c_{i,0}, c_{i,1}, c_{i,N-1}, c_{i,N}, c_{i,N+1}$ , które to z kolei należy traktować podczas optymalizacji jako zmienne ustalone (ang. *fixed*).

<sup>4</sup>Kod źródłowy biblioteki CppAD w języku C++ dostępny jest pod adresem <http://projects.coin-or.org/CppAD>



Rys. 3. Trajektorie we współrzędnych wewnętrznych dla obu robotów



Rys. 4. Trajektorie położenia i orientacji obiektu we współrzędnych zewnętrznych

Uzyskane wyniki optymalizacji przedstawiono na rys. (3)–(4). Jak widać, obliczone trajektorie ruchu we współrzędnych wewnętrznych obu robotów są ze sobą ściśle skoordynowane i jeśli zostaną dokładnie zrealizowane to w rezultacie belka będzie przeniesiona wspólnie przez dwa roboty. Dzięki aproksymacji funkcjami sklejanymi z wielomianów trzeciego stopnia obliczone trajektorie dla ogniów są dostatecznie gładkie. Trajektorie dla obiektu pokazane na rys. (4) otrzymano rozwiązując proste zadanie kinematyki dla obliczonych wartości  $\mathbf{q}_1$ . Na wykresach przedstawiono przebiegi współrzędnych kartezjańskich XYZ oraz kątów Eulera Z-Y-Z (odpowiednio  $\Phi$ ,  $\Theta$  i  $\Psi$ ) dla punktu będącego początkiem układu  $F_o$  związanego z obiektem. Linią kropkowaną przedstawiono trajektorię zadaną (niekoniecznie realizowalną) obiektu, zaś linią ciągłą przebiegi obliczone. Dokładność spełnienia więzów ruchu, wynikająca ze spełnienia ograniczeń równościowych w zadaniu optymalizacji nieliniowej, dla rozpatrywanego przypadku jest rzędu  $1e-12$ , co w zestawieniu z powtarzalnością pozycjonowania dla robota IRp-6 wynoszącą  $\pm 0.2$  mm, czyni obliczone trajektorie ruchu praktycznie stosownymi w układzie sterowania systemem dwurobotowym.

## 6. PODSUMOWANIE

W pracy omówiono algorytm obliczania ściśle dopuszczalnych torów ruchu dla ściśle współpracujących robotów. Proponowane podejście może być stosowane dla mechanizmów o dowolnej liczbie stopni swobody i dowolnej liczbie pętli kinematycznych. Nie jest wymagane rozwiązywanie odwrotnego zadania kinematyki dla złożonych mechanizmów zawierających zamknięte łańcuchy w algorytmie jest wykorzystywane tylko proste zadanie kinematyki manipulatora. Obliczone ścieżki są dopuszczalne i dostatecznie gładkie. Klasyczne algorytmy programowania nieliniowego umożliwiają uzyskanie rozwiązania suboptymalnego. Dla wyznaczenia rozwiązania optymalnego konieczne jest zastosowanie metod optymalizacji globalnej.

## LITERATURA

- [1] B. M. Bell, J. V. Burke. Algorithmic Differentiation of Implicit Functions and Optimal Values. In Ch. H. Bischof, H. M. Bücker, P. D. Hovland, U. Naumann, J. Utke, editors, *Advances in Automatic Differentiation*, str. 67–77. Springer, 2008.
- [2] H. Y. Benson, D. F. Shanno, R. J. Vanderbei. A Comparative Study of Large-Scale Nonlinear Optimization Algorithms. Raport instytutowy ORFE-01-04, Operations Research and Financial Engineering, Princeton University, 2002.
- [3] H. Y. Benson, D. F. Shanno, R. J. Vanderbei. Interior-Point Methods for Nonconvex Nonlinear Programming: Filter Methods and Merit Functions. Raport instytutowy ORFE-00-06, Operations Research and Financial Engineering, Princeton University, 2001.
- [4] C. de Boor. *Practical guide to splines*. Springer, New York, Heidelberg, 1978.
- [5] R. H. Byrd, J. Ch. Gilbert, J. Nocedal. A Trust Region Method Based on Interior Point Techniques for Nonlinear Programming. *Mathematical Programming*, 89:149–185, 2000.
- [6] R. H. Byrd, M. E. Hribar, J. Nocedal. An Interior Point Algorithm for Large Scale Nonlinear Programming. *SIAM Journal on Optimization*, 9(4):877–900, 1999.
- [7] J. Błaszczyk, A. Karbowski, K. Malinowski. Object Library of Algorithms for Dynamic Optimization Problems; Benchmarking SQP and Nonlinear Interior Point Methods. *International Journal of Applied Mathematics and Computer Science*, 17(4):515–537, 2007.
- [8] J. P. Błaszczyk. *Obiektowa biblioteka algorytmów optymalizacji dynamicznej; badanie efektywności metod sekwencyjnego programowania kwadratowego i punktu wewnętrznego dla zadań nieliniowych*. Praca doktorska, Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych, Instytut Automatyki i Informatyki Stosowanej, Warszawa, 2007. Rozprawa doktorska w dyscyplinie Automatyka i Robotyka.
- [9] J. Cortés, T. Siméon, J.-P. Laumond. A Random Loop Generator for Planning the Motions of Closed Kinematic Chains using PRM Methods. *IEEE International Conference on Robotics and Automation ICRA*, str. 2141–2146, 2002. IEEE.
- [10] J.W. Daniel. *Approximate minimisation of functionals*. Prentice Hall, Englewood Cliffs, N.J., 1971.

- [11] E. D. Dolan, J. J. Moré. Benchmarking Optimization Software with Performance Profiles. *Mathematical Programming*, 91(2):201–213, 2002.
- [12] A. V. Fiacco, G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. John Wiley and Sons, New York/London, 1968.
- [13] A. Fiser, R.K. Do, A. Sali. Modeling of loops in protein structure. *Protein Science*, 9(9):1753–1773, 2000.
- [14] R. Fletcher, S. Leyffer. Nonlinear Programming without a Penalty Function. *Mathematical Programming*, 91(2):239–269, 2002.
- [15] M. Galicki. *Wybrane metody planowania optymalnych trajektorii robotów manipulacyjnych*. WNT, 2000.
- [16] I. M. Gelfand, S.W. Fomin. *Rachunek wariacyjny*. PWN, Warszawa, 1979.
- [17] L. Han, N. Amato. A kinematics-based probabilistic roadmap method for closed chain systems. *Workshop on Algorithmic Foundations of Robotics*, str. 233–245, 2000.
- [18] M. Kallmann, A. Aubel, T. Abaci, D. Thalmann. Planning collision-free reaching motions for interactive object manipulation and grasping. *Eurographics*, 22:313–322, 2003.
- [19] L. E. Kavraki, P. Svestka, J.-C. Latombe, M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.
- [20] K. Kozłowski, P. Dutkiewicz, W. Wróblewski. *Modelowanie i sterowanie robotów*. Wydawnictwo Naukowe PWN, Warszawa, 2003.
- [21] J. J. Kuffner, S. M. LaValle. RRT-Connect: An efficient approach to single-query path planning. *IEEE International Conference on Robotics and Automation*, str. 995–1001, 2000.
- [22] J.-C. Latombe. *Robot motion planning*. Kluwer, Boston, MA, 1991.
- [23] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006.
- [24] J.P. Merlet. *Parallel Robots*. Kluwer, Dordrecht, 2000.
- [25] J. L. Morales, J. Nocedal, R. A. Waltz, G. Liu, J.-P. Goux. Assessing the Potential of Interior Methods for Nonlinear Optimization. Raport instyтуowy OTC 2001/4, Optimization Technology Center of Northwestern University, 2001.
- [26] R. M. Murray, Z. Li, S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.
- [27] T. Siméon, J. -P. Laumond, J. Cortés, A. Sahbani. Manipulation planning with probabilistic roadmaps. *The International Journal of Robotics Research*, 23(7-8):729–746, 2004.
- [28] W. Szynekiewicz. Motion Planning for Multi-Robot Systems with Closed Kinematic Chains. *9th IEEE Int. Conf. on Methods and Models in Automation and Robotics*, str. 779–786, Międzyzdroje, 2003.
- [29] K. Tchoń, I. Dulęba, A. Mazur, R. Hossa, R. Muszyński. *Manipulatory i roboty mobilne. Modele, planowanie ruchu, sterowanie*. Akademicka Oficyna Wydawnicza PLJ, Warszawa, 2000.



- [30] A. L. Tits, A. Wächter, S. Bakhtiari, T. J. Urban, C. T. Lawrence. A Primal-Dual Interior-Point Method for Nonlinear Programming with Strong Global and Local Convergence Properties. Raport instytutowy TR 2002-29, Institute for Systems Research, University of Maryland, 2002.
- [31] J. C. Trinkle, R. J. Milgram. Complete path planning for closed kinematic chains with spherical joints. *International Journal of Robotics Research*, 21(9):773–789, 2002.
- [32] M. Ulbrich, S. Ulbrich, L. N. Vicente. A Globally Convergent Primal-Dual Interior-Point Filter Method for Nonlinear Programming. *Mathematical Programming*, 100(2):379–410, 2004.
- [33] R. J. Vanderbei, D. F. Shanno. An Interior-Point Algorithm for Non-convex Nonlinear Programming. Raport instytutowy SOR-97-21, Statistics and Operations Research, Princeton University, 1997.
- [34] A. Wächter. *An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering*. Ph. D. Dissertation, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA, USA, 2002.
- [35] A. Wächter, L. T. Biegler. Failure of Global Convergence for a Class of Interior Point Methods for Nonlinear Programming. *Mathematical Programming*, 88(3): 565–574, 2000.
- [36] A. Wächter, L. T. Biegler. Line Search Filter Methods for Nonlinear Programming: Motivation and Global Convergence. *SIAM Journal on Optimization*, 16(1): 1–31, 2005.
- [37] A. Wächter, L. T. Biegler. On the Implementation of a Primal-Dual Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming. *Mathematical Programming*, 106(1):25–57, 2006.
- [38] R. A. Waltz, T. Plantenga. *KNITRO 5.0 User's Manual*. 2006.
- [39] W. Wedemeyer, H. Scheraga. Exact analytical loop closure in proteins using polynomial equations. *Journal of Computational Chemistry*, 20(8):819–844, 1999.
- [40] J. H. Yakey, S. M. LaValle, L. E. Kavraki. Randomized path planning for linkages with closed kinematic chains. *IEEE Transactions on Robotics and Automation*, 17(6):951–958, 2001.
- [41] T. Zielińska. *Maszyny kroczące*. PWN, 2003.
- [42] C. Zieliński, W. Szykiewicz, T. Winiarski, T. Kornuta. MRROC++ Based System Description. Raport instytutowy 06-9, IAIS, Warsaw, 2006.