

Instytut Informatyki, Automatyki i Robotyki

**Seria: PREPRINTY nr 38/2007**

**Metody planowania ruchu manipulatorów mobilnych w środowisku kolizyjnym**

(rozprawa doktorska)

Paweł Ludwików

Promotor: dr hab. inż. Ignacy Dulęba, prof. PWr.

Słowa kluczowe:

- planowanie ruchu
- środowisko kolizyjne
- manipulator mobilny
- układ nieholonomiczny
- metody Lie-algebraiczne

Wrocław, 5.9.2007

# Spis treści

<b>1. Wstęp</b>	3
<b>2. Preliminaria</b>	6
2.1. Notacja i operatory	6
2.2. Oznaczenia	6
2.3. Lokalne kierunki ruchu a sterowania	7
2.3.1. Realizacja sterowań	10
<b>3. Postawienie zadanie planowania i idea jego rozwiązania</b>	12
3.1. Idea rozwiązania zadania planowania ruchu manipulatora mobilnego	14
<b>4. Przegląd metod planowania ruchu</b>	16
4.1. Metoda planowania ruchu nilpotentnych układów nieholonomicznych	18
4.2. Metoda sterowań sinusoidalnych	20
4.3. Metoda kontynuacji	23
4.4. Metoda typu pontriaginowskiego	24
4.5. Metoda typu Lapunowa	25
4.6. Metody newtonowskie	26
4.7. Metody przestrzeni endogenicznej	28
4.8. Manipulowalność a planowanie ruchu	31
4.9. Metoda optymalizacyjna	32
4.10. Planowanie trajektorii robotów mobilnych w środowisku kolizyjnym	35
4.11. Metody pomocnicze w przestrzeniach kolizyjnych	41
4.12. Ocena przydatności ścieżek	43
<b>5. Algorytmy</b>	46
5.1. Konfiguracja transportowa manipulatora	47
5.1.1. Przenoszenie sił nacisku na podłoże	47
5.1.2. Optymalna lokalizacja siły nacisku	51
5.1.3. Wpływ pochylenia platformy na optymalne położenie siły	52
5.1.4. Wpływ tarcia i zjawisko poślizgu na optymalne położenie siły	53
5.1.5. Położenie środka masy manipulatora	54
5.2. Zbiór pól docelowych platformy	55
5.2.1. Wybór bieżącej pozycji docelowej platformy	57
5.3. Wyznaczenie lokalnie optymalnej ścieżki platformy	58
5.3.1. Planowanie ścieżki holonomicznej	58
5.3.2. Poprawienie ścieżki holonomicznej	59
5.3.3. Metoda lokalnych wariacji	61
5.3.4. Metoda lokalnego programowania dynamicznego	62
5.4. Dobór sterowań zapewniający bezpieczeństwo manewru platformy	63
5.5. Rozwiązywalność zadania planowania ruchu platformy	66
<b>6. Symulacje</b>	67
6.1. Modele manipulatorów	67
6.2. Modele platform	68
6.2.1. Monocykl	69

---

6.2.2.	Dwukołowiec . . . . .	70
6.3.	Przeszkody . . . . .	71
6.4.	Sterowania . . . . .	71
6.5.	Parametry algorytmu planowania ruchu . . . . .	72
6.6.	Badania symulacyjne . . . . .	73
6.6.1.	Zbiór konfiguracji docelowych . . . . .	73
6.6.2.	Zmiana dokładności realizacji ścieżki . . . . .	73
6.6.3.	Uproszczony sposób wyznaczania dokładności realizacji . . . . .	78
6.6.4.	Liczba punktów ścieżki testowej . . . . .	78
6.6.5.	Współczynniki wykorzystywane w algorytmie elastycznej wstęgi . . . . .	78
6.6.6.	Współczynniki optymalizatora nieholonomicznego . . . . .	83
6.6.7.	Wielokrotne planowanie ścieżki . . . . .	86
6.6.8.	Ograniczenie dopuszczalnej rodziny sterowań . . . . .	86
6.6.9.	Algorytm optymalizacji odległości od przeszkód . . . . .	90
<b>7.</b>	<b>Zakończenie . . . . .</b>	<b>92</b>
<b>A.</b>	<b>Dodatki . . . . .</b>	<b>93</b>
A.1.	Algorytm Newtona . . . . .	93
A.2.	Sprawdzanie kolizyjności robota z przeszkodami . . . . .	94
A.3.	Obszerność manewru . . . . .	94
A.3.1.	Wskaźniki obszerności manewru . . . . .	95
<b>Bibliografia</b>	<b>. . . . .</b>	<b>99</b>

# 1. Wstęp

Postęp technologiczny oraz wyzwania współczesnego przemysłu i usług stymulują rozwój robotyki w kierunku zwiększonej autonomii robotów i ich mobilności. Manipulatory przemysłowe, jakkolwiek użyteczne, oferowały usługi jedynie w ograniczonej przestrzeni hal fabrycznych. W ostatnich latach coraz popularniejsze jest wykorzystanie robotów mobilnych. Najczęściej są to platformy poruszające się na kołach, bogato wyposażone w układy sensoryczne (sonary ultradźwiękowe, dalmierze laserowe, kamery, czujniki zbliżeniowe). Podstawową cechą tych platform jest umiejętność osiągnięcia wyznaczonego celu z wykorzystaniem technik modelowania środowiska i metod planowania ruchu. Zdolności sprawcze platform mobilnych są ograniczone i polegają co najwyżej na biernym przewożeniu ładunków jak w przypadku zastosowania platformy dostarczającej pacjentom leki w szpitalach. Naturalnym więc jest połączenie zdolności mobilnych platform ze sprawczymi manipulatorów w jeden układ zwany manipulatorem mobilnym. Techniki planowania ruchu wypracowane dla elementarnych podukładów manipulatora mobilnego są już ugruntowane. Jednak interesującym jest zagadnienie zbadania istnienia synergii dwóch podukładów na gruncie planowania ruchu i ewentualnego jej wykorzystania w praktyce. Wiadomo, że zdecydowana większość manipulatorów to układy holonomiczne, czyli (w uproszczeniu), z odrębnym silnikiem napędzającym każdy ze stopni swobody. Natomiast kołowe platformy mobilne są przedstawicielkami układów nieholonomicznych, w których liczba sterowań jest istotnie mniejsza od wymiaru przestrzeni konfiguracyjnej. Zatem, w typowym przypadku, planowanie ruchu manipulatora jest istotnie łatwiejsze od planowania ruchu platformy. Istnieją w literaturze robotycznej podejścia unifikujące, których istotą jest potraktowanie obu podukładów manipulatora mobilnego jednolicie, sformułowanie modelu o wektorze stanu będącym konkatenacją wektorów stanów podukładów oraz rozwiązanie klasycznymi metodami problemu w pełnej przestrzeni stanu. W niniejszej dysertacji, której celem jest zaproponowanie metod planowania ruchu manipulatora mobilnego w środowisku kolizyjnym, przedstawiono nieco odmienne podejście. Skoro trudniejszym jest zadanie planowania ruchu dla platformy niż dla manipulatora na niej umieszczonego, zatem celowym wydaje się położenie nacisku na efektywne planowanie ruchu platformy, a służebne potraktowanie planowania ruchu manipulatora, którego zadaniem będzie generowanie "łatwych" do osiągnięcia konfiguracji platformy. Obrazowo ideę tę można przedstawić następująco: zamiast planować ruch manipulatora mobilnego hipotetycznie rozsprzęgamy manipulator od platformy. Manipulator o uwięzionym efektorze w docelowym punkcie przestrzeni zadaniowej ma możliwość przemieszczania swych ogniw. Tak więc baza (podstawa) manipulatora, dla tradycyjnego manipulatora nieruchoma, uzyskuje możliwości ruchowe i kreśli zbiór potencjalnych docelowych konfiguracji platformy mobilnej, czyli takich dla których manipulator mobilny osiąga punkt docelowy w przestrzeni zadaniowej. W ten sposób uzyskuje się więc naturalne połączenie klasycznych metod planowania ruchu manipulatorów i platform mobilnych traktowanych odrębnie, a jednocześnie efekt synergetyczny, polegający na tym, że zbiór konfiguracji docelowych platformy, inaczej niż w klasycznych zadaniach, jest nie pojedynczym punktem, lecz zbiorem konfiguracji. Zatem planowanie ruchu platformy mobilnej może być łatwiejsze, tym bardziej, że podczas swego ruchu może ona wybierać jako konfiguracje docelowe różne konfiguracje oferowane przez manipulator.

W pracy modelujemy obydwa podukłady manipulatora mobilnego na poziomie kinematycznym, to znaczy manipulator w pełni zadany jest przez jego kinematykę prostą, natomiast platforma mobilna opisywana jest bezdryfowym układem nieholonomicznym wynikającym z ograniczeń w postaci Pfaffa.

Dla większości metod planowania ruchu podejście takie jest powszechne w literaturze i wystarczające w praktyce (dynamika komponent modelowana jest zwykle dopiero na poziomie sterowania). Zdecydowana większość manipulatorów i platform mobilnych należy do rozważanej klasy.

Zadanie planowania ruchu dodatkowo utrudniamy przez ograniczenia przestrzeni konfiguracyjnej wynikające z istnienia przeszkód w otoczeniu manipulatora mobilnego. Wynikiem planowania ma być bezkolizyjna trajektoria (manipulatora i platformy) zapewniająca osiągnięcie zadanego punktu przestrzeni zadaniowej. Ważnym elementem rozwiązania zadania planowania jest dobór metody planowania ruchu platformy. Istnienie przeszkód praktycznie uniemożliwia efektywne stosowanie metod globalnych, które są bardzo czułe na ograniczenia w przestrzeni konfiguracyjnej. Co więcej, w zadaniach praktycznych dość często środowisko nie jest w pełni znane (modelowane łącznie z przemieszczaniem się robota), a pełna znajomość środowiska jest warunkiem *sine qua non* stosowania metod globalnych. Jako metodę bazową wybieramy lokalną metodę Lie-algebraiczną planowania ruchu układów nieholonomicznych. Modele przez nią akceptowane są w pełni zgodne z założonymi dla manipulatora mobilnego, jest elastyczna w uwzględnianiu przeszkód i była stosowana z powodzeniem przez członków zespołu badawczego Zakładu Podstaw Cybernetyki i Robotyki do zadania planowania ruchu platform nieholonomicznych z jednym punktem docelowym.

Dość często zadanie planowania ruchu rozpatrywane jest jedynie z uwzględnieniem modelu obiektu sterowanego, lecz bez uwzględnienia jego potencjalnych interakcji ze środowiskiem. Metoda proponowana w dysertacji, jakkolwiek nie uwzględnia wszystkich możliwych interakcji to jednak bierze pod uwagę fizyczne uwarunkowania ruchu. Dlatego też dla zadania planowania ruchu wykorzystywana jest oryginalna metafora robota-dźwigu, w której zadanie planowania jest dekomponowane na następujące podzadania częściowe: przyjęcie przez manipulator konfiguracji transportowej, przemieszczenie platformy do jej dynamicznie wygenerowanej konfiguracji docelowej, przemieszczenie manipulatora tak, by realizował wraz z platformą punkt docelowy w przestrzeni zadaniowej.

Techniki zaproponowane w dysertacji są dedykowane manipulatorom mobilnym o kołowej platformie jezdnej, jednak możliwe jest wykorzystanie ich elementów składowych także dla innych manipulatorów mobilnych o modelu wynikającym z ograniczeń w postaci Pfaffa, czyli robotów szybujących w przestrzeni kosmicznej i pojazdów podwodnych.

Dysertacja zorganizowana jest w siedem rozdziałów, uzupełnionych dodatkami. We wprowadzającym rozdziale 2 opisano najczęściej używane oznaczenia, wprowadzono niezbędny aparat metod Lie-algebraicznych intensywnie stosowanych w planowaniu ruchu platformy. Odwracając najczęstszą kompozycję rozpraw doktorskich (aby nie powtarzać terminów i modeli) najpierw, w rozdziale 3, zdefiniowano zadanie planowania ruchu manipulatora mobilnego w środowisku kolizyjnym i przedstawiono główne idee jego rozwiązania. Dopiero następnie, w kolejnym rozdziale, opisano rozwiązania literaturowe zadania planowania ruchu dla platform mobilnych i manipulatorów traktowanych jako układy niezależne oraz metody łączące dwa podukłady manipulatora mobilnego w jedną całość. Rozdział 5 jest uszczegółowieniem idei rozwiązania zadania planowania i wprowadza algorytmy określające z detalami sposób planowania ruchu manipulatora mobilnego i dyskutuje znaczenie parametrów warunkujących działanie metody. W rozdziale 6 przedstawiono wyniki symulacji działania metody Lie-algebraicznej planowania ruchu manipulatora mobilnego. Jako testowane platformy wybrano układ jednośladowy i dwuśladowy, natomiast część manipulacyjną tworzą dwuwahadło i trójwahadło. Zarówno platformy jak i manipulatory oraz przeszkody wybrano tak, by zilustrować idee na jak najprostszych przykładach pokazujących specyfikę zastosowanych narzędzi. Zastosowanie proponowanych metod do bardziej skomplikowanych modeli i środowisk nie nastrocza zasadniczych kłopotów, zwiększając jedynie nakłady obliczeniowe. Część symulacyjna zawiera wnioski z przeprowadzonych testów wskazujących na wpływ poszczególnych faz planowania i ich parametrów na jakość wynikowego toru i trajektorii ruchu. W podsumowującym rozprawę rozdziale 7 uogólniono wyniki szcze-

główne i wskazano cechy zaproponowanego rozwiązania zadania planowania ruchu. W dodatkach zebrano materiał pomocniczy.

**Skład:**

Do składu pracy wykorzystano system przygotowania dokumentów  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  [45], będący nakładką systemu  $\text{T}_{\text{E}}\text{X}$  [40]. Rysunki wykonano programem XFig (<http://www.xfig.org/>). Wykresy wygenerowano programem gnuplot (<http://www.gnuplot.info/>). Obliczenia symboliczne wykonano ze wsparciem programu Mathematica firmy Wolfram Research, Inc., [93]. Pomocnicze skrypty łączące poszczególne części w całość zostały napisane w językach Perl [92], Python [63] i Bash. Główny program na użytek symulacji komputerowych napisano w języku C [38] z wykorzystaniem bibliotek Gnu Scientific Library (<http://www.gnu.org/software/gsl/>).

## 2. Preliminaria

W rozdziale przedstawiono notacje oraz podstawowe pojęcia występujące w dalszej części pracy.

### 2.1. Notacja i operatory

Wielkości skalarne pisane są czcionką pochyłą ( $a$ ), wielkości wielowymiarowe — wektory, macierze i zbiory są oznaczono symbolami pogrubionymi ( $\mathbf{a}$ ,  $\mathbf{A}$ ). Przestrzenie zapisywane są z użyciem stylizowanych dużych liter ( $\mathbb{A}$ ). Domyślnie, wszystkie wektory są kolumnowe.

W opisach algorytmów wykorzystano symbol  $\leftarrow$  do operacji przypisania (by uniknąć niepoprawnego matematycznie zapisu  $i = i + 1$ ). Zatem przypisanie  $i \leftarrow i + 1$  należy odczytać jako powiększenie zmiennej  $i$  o jeden.

Wykorzystywane w pracy operatory oznaczane są następująco:

- $\mathbf{x} \circ \mathbf{y}$  — iloczyn skalarny,
- $\mathbf{a} \times \mathbf{b}$  — iloczyn wektorowy,
- $\mathbf{A}^T$  — transpozycja wektora, macierzy,
- $\|\mathbf{x}\|$  — norma wektora (o ile nie założono inaczej – norma euklidesowa),
- $\mathbf{A}^{-1}$  — odwrotność macierzy kwadratowej,
- $\mathbf{A}^\#$  — uogólniona odwrotność macierzy – pseudoodwrotność Moore’a-Penrose’a:  
$$\mathbf{A}^\# = \mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1},$$
- $\text{diag}(\mathbf{A})$  — macierz diagonalna,
- $a \times b$  – rozmiar macierzy (w przykładzie macierz o  $a$  wierszach i  $b$  kolumnach),
- $\dim(\mathbf{a})$  — wymiar (przestrzeni, wektora),
- $\text{mod}$  — modulo
- $\#\mathbf{A}$  — liczba elementów (moc) zbioru,
- $\det(\mathbf{A})$  — wyznacznik macierzy,
- $[\mathbf{A}, \mathbf{B}]$  — nawias Liego pól wektorowych  $\mathbf{A}$ ,  $\mathbf{B}$ ,
- $\text{stopień}(\mathbf{A})$  — stopień pola wektorowego (jednomianu Liego)  $\mathbf{A}$ ,
- $\text{rank}(LA(\mathbf{G}(\mathbf{q})))$  — rząd algebry Liego rozpiętej przez generatory (kolumny macierzy  $\mathbf{G}$ ) w punkcie  $\mathbf{q}$ ,
- $\nabla_z$  — pochodna względem  $z$ .

### 2.2. Oznaczenia

Najczęściej wykorzystywane oznaczenia zebrano poniżej.

Przestrzenie:

- $\mathbb{R}^k$  —  $k$ -wymiarowa przestrzeń liczb rzeczywistych,
- $\mathbb{X}$  — przestrzeń konfiguracyjna manipulatora,

- $\mathbb{Q}$  — przestrzeń pól platformy,
- $\mathbb{Y}$  — przestrzeń zadaniowa manipulatora mobilnego,
- $\mathbb{U}$  — przestrzeń sterowań,
- $\mathbb{SO}(3)$  — specjalna grupa ortogonalna wymiaru 3 do której należą macierze rotacji (obrotu)  $\mathbf{R}$  ( $3 \times 3$ ) spełniające własności  $\mathbf{R}^{-1} = \mathbf{R}^T$ ,  $\det(\mathbf{R}) = 1$ ,
- $\mathbb{SE}(3)$  — specjalna grupa euklidesowa, zawierająca elementy będące parami (macierz obrotu, wektora przesunięcia–translacji), jest sumą półprostą  $\mathbb{SE}(3) \simeq \mathbb{SO}(3) \times \mathbb{R}^3$ .

Wektory, funkcje:

- $\mathbf{x} \in \mathbb{X}$ ,  $\dim \mathbf{x} = r$  — wektor konfiguracji manipulatora,
- $\mathbf{q} \in \mathbb{Q}$ ,  $\dim \mathbf{q} = n$  — wektor stanu platformy,
- $\mathbf{u}(t) \in \mathbb{U}$ ,  $\dim \mathbf{u} = m$  — sterowanie platformy w chwili  $t$ ,
- $\mathbf{u}(\cdot)$  — sterowanie platformy na horyzoncie czasowym,
- $\mathbf{y} \in \mathbb{Y} \subset \mathbb{SE}(3)$  — wektor współrzędnych efektora,
- $\mathbf{g}_i(\mathbf{q})$  —  $i$ -ty generator, pole wektorowe,
- $\mathbf{G}(\mathbf{q})_{n \times m}$  — macierz generatorów,
- $\mathbf{0}$  — wektor (macierz) o wszystkich elementach będących zerami,
- $\mathbf{I}$  — kwadratowa macierz jednostkowa, (na diagonalu wartości 1, poza – 0).

Przekształcenia:

- $\mathbf{k}(\mathbf{q}, \mathbf{x}) \in \mathbb{Y}$  — kinematyka prosta manipulatora mobilnego,

Dane specyficzne dla algorytmów planowania ruchu:

- $t$  zmienna opisująca czas,
- $\mathbf{q}_c$  — stan platformy w bieżącym punkcie (podczas omawiania algorytmów planowania),
- $\Delta T$  — długość elementarnego kroku planowania,
- $\mathbf{Q}_0$  — macierz wag dla algorytmu pseudoodwrotności ważonej.

Opis otoczenia robota:

- układ globalny  $\mathbf{G}$ ,
- układ docelowy  $\mathbf{D}$ , położony „nad początkiem” układu  $\mathbf{G}$ :  $\mathbf{T}_G^D = Tr(Z, z) \cdot Rot$ ,  $Rot \in \mathbb{SO}(3)$ , gdzie  $Tr(Z, z)$  oznacza przesunięcie wzdłuż osi  $Z$  o  $z \in \mathbb{R}$ ,
- układ początkowy  $\mathbf{P}$ .

### 2.3. Lokalne kierunki ruchu a sterowania

<sup>1</sup> Pole wektorowe jest wektorem o elementach będących funkcjami (z założenia analitycznymi), które przypisuje każdemu punktowi przestrzeni odpowiadający mu wektor mający interpretację kierunku ruchu. Regularność pola wektorowego wymaga odpowiednio ciągłej zmiany wektorów od punktu do punktu leżącego w jego małym otoczeniu. Nawias Liego jest operatorem generującym z dwóch pól–argumentów  $\mathbf{A}, \mathbf{B}$  kolejne pole wektorowe  $[\mathbf{A}, \mathbf{B}]$ . We współrzędnych nawias Liego zdefiniowany jest następująco:

$$[\mathbf{A}, \mathbf{B}] = \frac{\partial \mathbf{B}(\mathbf{q})}{\partial \mathbf{q}} \mathbf{A} - \frac{\partial \mathbf{A}(\mathbf{q})}{\partial \mathbf{q}} \mathbf{B}. \quad (2.1)$$

<sup>1</sup> Podrozdział opracowano na podstawie prac [12, 18]



Łatwo pokazać, korzystając z definicji (2.1), następujące własności nawiasu Liego:

- $[\mathbf{A}, \mathbf{B}] = -[\mathbf{B}, \mathbf{A}]$  zwana antysymetrią,
- $[\mathbf{A}, [\mathbf{B}, \mathbf{C}]] + [\mathbf{B}, [\mathbf{C}, \mathbf{A}]] + [\mathbf{C}, [\mathbf{A}, \mathbf{B}]] = \mathbf{0}$  znana jako tożsamość Jacobięgo.

Niech będzie zadany układ pól wektorowych zwanych generatorami:  $\mathbf{g}_i, i = 1, \dots, m$ . Rekurencyjnie definiujemy stopień pola wektorowego uzyskanego, przy pomocy operacji (2.1), z generatorów i ich potomków:

$$\begin{cases} \text{stopień}(\mathbf{g}_i) = 1, \\ \text{stopień}([\mathbf{V}, \mathbf{Z}]) = \text{stopień}(\mathbf{V}) + \text{stopień}(\mathbf{Z}). \end{cases} \quad (2.2)$$

Wszystkie pola wektorowe o tym samym stopniu należą do warstwy:

$$W_i = \{\mathbf{A} \mid \text{stopień}(\mathbf{A}) = i\}, \quad i = 1, \dots \quad (2.3)$$

Własności pól wektorowych wskazują, że nie wszystkie pola utworzone rekurencyjnie z generatorów są potencjalnie od siebie niezależne. Ponieważ zależne pola wektorowe nie wnoszą nowego kierunku ruchu, zatem warto je pominąć. Z tego powodu w literaturze robotycznej przyjmuje się najczęściej bazę Ph. Halla [89] tworzącą niezależny (nieskończony) zbiór jednomianów Liego. Ich kombinacja liniowa nad przestrzenią liczb rzeczywistych tworzy wolną algebrę Liego. Przykład początkowych elementów bazy Ph. Halla zawiera tabela 2.1.

Tablica 2.1. Elementy bazy Ph. Halla, do warstwy czwartej włącznie, generowanej przez  $\mathbf{X}, \mathbf{Y}$  i ich przypisanie do warstw.

warstwa	jednomian Liego	warstwa	jednomian Liego
1	$\mathbf{X}$	4	$[\mathbf{X}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]$
1	$\mathbf{Y}$	4	$[\mathbf{Y}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]$
2	$[\mathbf{X}, \mathbf{Y}]$	4	$[\mathbf{Y}, [\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]]$
3	$[\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]$		
3	$[\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]$		

Jako model matematyczny platformy mobilnej rozważanej w pracy służy bezdryfowy układ nieholonomiczny opisany równaniem:

$$\dot{\mathbf{q}} = \mathbf{G}(\mathbf{q})\mathbf{u} = \sum_{i=1}^m \mathbf{g}_i(\mathbf{q})u_i, \quad \dim \mathbf{q} = n, \dim(\mathbf{u}) = m, \quad (2.4)$$

gdzie  $\mathbf{g}_i, i = 1, \dots, m$  są generatorami. Po podstawieniu za nieokreślone generatory wolnej algebry Liego konkretnych pól wektorowych  $\mathbf{g}_i$  i ich zwartościowaniu, czyli wyliczeniu pól i ewentualnym ich uproszczeniu, otrzymujemy algebrę Liego pól wektorowych stowarzyszoną z (2.4),  $LA(\mathbf{G}(\mathbf{q}))$ . Rozważając algebrę Liego pól wektorowych w ustalonym punkcie przestrzeni stanu uzyskujemy kombinacje liniowe wektorów wskazujących możliwe kierunki ruchu, bez określenia jak te kierunki wygenerować przy pomocy sterowań.

Podstawową własnością badaną dla układów sterowania jest ich sterowalność, czyli możliwość osiągnięcia przy pomocy sterowań każdego punktu, gdy ruch zainicjowany jest w dowolnym punkcie przestrzeni. Trajektoria w przestrzeni stanu układu sterowalnego może być teoretycznie bardzo obszerna, nawet gdy stany brzegowe nie są od siebie odległe. Dlatego szczególnie dla planowania ruchu w środowiskach kolizyjnych wymagana jest własność mocniejsza, znana jako sterowalność lokalna w krótkim czasie (*small time locally controllable, STLC*). Własność STLC dla układu bezdryfowego (2.4) opisuje twierdzenie Chow [4]:

**Twierdzenie 2.3.1.** *Układ (2.4) jest lokalnie sterowalny w krótkim czasie, jeśli algebra Liego z nim stowarzyszona rozpiną  $\mathbb{R}^n$  w każdym punkcie przestrzeni stanu, czyli*

$$\forall \mathbf{q} \text{ rank } (LA(\mathbf{G}(\mathbf{q}))) = n. \quad (2.5)$$

Bez straty ogólności, algebrę Liego można zastąpić przez kombinację liniowe pól z bazy Halla (warunek (2.5) jest wtedy łatwiejszy do sprawdzenia, gdyż zawiera mniej pól). Dość często warunek (2.5) nosi nazwę rzędu algebry Liego, LARC - (*ang. Lie algebra rank condition*). Geometrycznie twierdzenie Chow mówi o lokalnej sterowalności w krótkim czasie, gdy możliwy jest ruch w dowolnym stanie w dowolnym kierunku. Ponadto, istnieje trajektoria łącząca każde punkty bliskie sobie zawarta w małym otoczeniu zawierające te punkty. Generowanie bazy Ph. Halla jest operacją czysto algorytmiczną (efektywny algorytm zawiera praca [11]) wykonywaną w trybie pre-planowania (*ang. off-line*). Zatem warunek rzędu (2.5) także jest sprawdzany w trybie pre-planowania, co w szczególności oznacza, że dla układów sterowalnych wiadomo, w których obszarach przestrzeni stanu jakie pola wektorowe rozpinają przestrzeń  $\mathbb{R}^n$ . W praktyce najczęściej można wybrać jeden minimalny (złożony z minimalnej liczby) zbiór pól wektorowych rozpinający przestrzeń w każdym punkcie.

Dysponując polami wektorowymi z bazy Ph. Halla i wykorzystywanymi do sprawdzenia warunku sterowalności (2.5) (czyli lokalnymi kierunkami ruchu w konkretnym punkcie przestrzeni) należy je uzyskać przy pomocy dopuszczalnych sterowań. Istnieją dwie drogi osiągnięcia tego celu. Jedna wykorzystująca formułę Campbella-Bakera-Hausdorffa-Dynkina, CBHD, [78] i oferująca sterowania kawałkami stale. Druga, stosująca uogólnienie formuły, gCBHD, której zaletą jest generowanie sterowań ciągłych (lokalnie, wokół bieżącego punktu).

Zapis formuły gCBHD jest dość skomplikowany, dlatego skoncentrujemy się na ważnych dla planowania ruchu aspektach praktycznych opuszczając dokładne wyprowadzenia zależności, które można znaleźć w pracach [13, 14]. Formuła (g)CBHD pełni w teorii równań różniczkowych podobną rolę jak wzór Taylora dla analizy funkcji statycznych, pozwala bowiem przedstawić (aproksymować) zmianę stanu w dowolnym punkcie przestrzeni stanu  $\mathbf{q}_c$  jako liniową kombinację pól wektorowych pomnożonych przez funkcje zależne od sterowań

$$\Delta \mathbf{q}(\mathbf{q}_c)(t) \simeq \sum_{i=1}^k \psi_i(\mathbf{u}(\cdot), t) \cdot \mathbf{Z}_i(\mathbf{q}_c), \quad (2.6)$$

gdzie  $\mathbf{Z}_i(\mathbf{q}_c)$ ,  $i = 1, \dots$ , są polami wektorowymi zwartościowanymi w punkcie  $\mathbf{q}_c$  i gwarantującymi spełnienie warunku (2.5). Jeśli założymy, że układ (2.4) spełnia warunek rzędu, zatem wartość górnego indeksu  $k$  jest skończona. Oczywiście  $k \geq n$ . Warto zauważyć, że funkcje  $\psi_i$  nie zależą od bieżącego stanu, można więc je wyliczyć w trybie pre-planowania. Jak lokalnie wpływają sterowania na przemieszczenie stanu  $\Delta \mathbf{q}$  w punkcie bieżącym  $\mathbf{q}_c$ , zależy w głównej mierze od układu pól wektorowych w tym punkcie. Interesująca jest także interpretacja zależności (2.6) od  $t$ . Czas  $t$  należy rozumieć jako parametr odpowiadający efektywnemu przemieszczaniu się stanu w kierunku  $\Delta \mathbf{q}(\mathbf{q}_c)$ . Sterowania są odmierzane rzeczywistym czasem ruchu  $s \in [0, t]$ . Istotnym faktem ukrytym w zależności (2.6) jest zauważenie, że lokalny ruch nie odbywa się wzdłuż odcinka  $\mathbf{q}(\tau) = \mathbf{q}_c + \Delta \mathbf{q}(\mathbf{q}_c)(t) \cdot \tau/t$ ,  $\tau = [0, t]$ . Trajektoria osiągając punkty tego odcinka przy zmiennym  $\tau$  może wykonywać dość obszerne przemieszczenia, co jest szczególnie ważne w planowaniu ruchu w środowisku kolizyjnym, stanowi bowiem istotne utrudnienie planowania. Formuła (2.6) jest zależnością aproksymacyjną (pominięto pola wyższych rzędów), zatem jej stosowalność jest ograniczona dla małych czasów  $t$ , gdyż wraz ze wzrostem  $t$  maleje dokładność aproksymacji przemieszczenia stanu.

Ponieważ w pracy ilustrowanie metod planowania ruchu manipulatora mobilnego będzie dla platform o dwóch wejściach,  $m = 2$ , przytoczymy uszczegółowioną postać obliczeniową zależności (2.6) w postaci

$$\Delta \mathbf{q} \simeq \alpha_X \mathbf{X} + \alpha_Y \mathbf{Y} + \alpha_{[X,Y]} [\mathbf{X}, \mathbf{Y}] + \alpha_{[X,[X,Y]]} [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]] + \alpha_{[Y,[X,Y]]} [\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]. \quad (2.7)$$

o współczynnikach  $\alpha_X$ ,  $\alpha_Y$ ,  $\alpha_{[X,Y]}$ ,  $\alpha_{[X,[X,Y]]}$ ,  $\alpha_{[Y,[X,Y]]}$  zależnych od sterowań  $\mathbf{u} = (u, v)^T$  danych zależnościami [18]

$$\begin{aligned}\alpha_X &= \int_{s=0}^t u(s)ds, & \alpha_Y &= \int_{s=0}^t v(s)ds, \\ \alpha_{[X,Y]} &= \frac{1}{2!} \int_{s_2=0}^t \int_{s_1=0}^{s_2} (u(s_1)v(s_2) - v(s_1)u(s_2)) ds_1 ds_2, \\ \alpha_{[X,[X,Y]]} &= \frac{1}{3!} \int_{s_3=0}^t \int_{s_2=0}^{s_3} \int_{s_1=0}^{s_2} (v(s_1)u(s_2)u(s_3) + u(s_1)u(s_2)v(s_3) - 2u(s_1)v(s_2)u(s_3)) ds_1 ds_2 ds_3, \\ \alpha_{[X,[X,Y]]} &= \frac{1}{3!} \int_{s_3=0}^t \int_{s_2=0}^{s_3} \int_{s_1=0}^{s_2} (-u(s_1)v(s_2)v(s_3) - v(s_1)v(s_2)u(s_3) + 2v(s_1)u(s_2)v(s_3)) ds_1 ds_2 ds_3.\end{aligned}\tag{2.8}$$

W pracy [18] wykazano, że przemieszczenie wzdłuż pola wektorowego  $\mathbf{Z}_i$  związany jest z jego stopniem wzorem:

$$\psi_i(\mathbf{u}(\cdot), t) \sim t^{\text{stopień}(\mathbf{Z}_i)/2}.\tag{2.9}$$

gdzie  $t$  jest małym czasem ruchu (by formuła (2.6)) była ważna. Współczynniki  $\psi_i(\mathbf{u}(\cdot), t)$  zależą również od parametrów zastosowanych sterowań  $\mathbf{u}(\cdot)$ , np. amplitud i przesunięć fazowych dla sterowań harmoniczych.

### 2.3.1. Realizacja sterowań

Przyjmijmy że dana jest trajektoria  $\mathbf{q}(t)$ ,  $t \in [0, 1]$ . Z założenia o sterowalności układu wynika, że w każdym punkcie stanu istnieje zbiór pól wektorowych

$$\mathbf{G}(\mathbf{q}) = \{\mathbf{g}_1(\mathbf{q}), \dots, \mathbf{g}_m(\mathbf{q}), \mathbf{g}_{m+1}(\mathbf{q}), \dots, \mathbf{g}_s(\mathbf{q})\}\tag{2.10}$$

gdzie  $\mathbf{g}_{m+1}(\mathbf{q}), \dots, \mathbf{g}_s(\mathbf{q})$  są polami uzyskanymi przy użyciu nawiasu Liego zastosowanego rekurencyjnie do generatorów  $\mathbf{g}_1(\mathbf{q}), \dots, \mathbf{g}_m(\mathbf{q})$ , spełniający warunek rzędu algebry Liego (*Lie algebra rank condition*, *LARC*)

$$\text{rank } \mathbf{G}(\mathbf{q}) = n.\tag{2.11}$$

Warto zauważyć, że liczba pól wektorowych może być większa niż wymiar przestrzeni stanu, ponieważ wraz z wyborem jednego wektora (pola) o pewnym stopniu, do  $\mathbf{G}(\mathbf{q})$  automatycznie wszystkie pozostałe wektory (pola) danego stopnia są dołączane jako mające porównywalny wpływ na ruch. Z tego powodu często występuje sytuacja, że liczba wektorów wchodzących w skład  $\mathbf{G}$  przewyższa wymiarowość przestrzeni stanu. Na podstawie zależności (2.9) należy zauważyć, iż do  $\mathbf{G}$  powinny być dobierane wektory o najniższym możliwym stopniu, ponieważ są bardziej efektywne (dają dłuższe przesunięcia) w porównaniu do wektorów stopni wyższych. W każdym punkcie stanu może być inny skład pól wchodzących w skład  $\mathbf{G}(\mathbf{q})$ . Lokalne prędkości  $\dot{\mathbf{q}}$  można wyrazić w formie rozszerzonych (wirtualnych) sterowań  $\mathbf{v}$

$$\dot{\mathbf{q}} = \mathbf{G}(\mathbf{q})\mathbf{v}\tag{2.12}$$

otrzymywanych z zależności

$$\mathbf{v} = \mathbf{G}^\#(\mathbf{q})\dot{\mathbf{q}}.\tag{2.13}$$

W przypadku, gdy sterowania mają różny wkład w całkowity ruch stosuje się ważoną pseudoodwrotność [60]

$$\mathbf{v} = \mathbf{Q}_0(\mathbf{G}^*(\mathbf{q}))^\# \dot{\mathbf{q}},\tag{2.14}$$

gdzie  $\mathbf{Q}_0$  jest dodatniookreśloną macierzą wag, oraz

$$\mathbf{G}^*(\mathbf{q}) = \mathbf{G}(\mathbf{q})\mathbf{Q}_0^{-1}. \quad (2.15)$$

Użycie bardziej wymagającej obliczeniowo metody pseudoodwrotności ważonej zamiast standardowego pseudoinwersu Moore-Penrose’a podyktowane jest różną efektywnością pól wektorowych wchodzących w skład macierzy  $\mathbf{G}$ . Ruch wzdłuż wektorów o wyższym stopniu powinien być tłumiony na rzecz pól niższego stopnia.

### 3. Postawienie zadanie planowania i idea jego rozwiązania

Niech będzie zadany globalny układ współrzędnych manipulatora mobilnego i związana z nim przestrzeń zadaniowa rozumiana jako zbiór punktów w układzie globalnym osiągalnych przez manipulator mobilny. Układ współrzędnych holonomicznego manipulatora jest sztywno związany z nieholonomiczną platformą mobilną. Pozę platformy oznaczamy przez  $\mathbf{q} \in \mathbb{Q}$ , a konfigurację manipulatora przez  $\mathbf{x} \in \mathbb{X}$ . Na platformę oddziałujemy przy pomocy sterowań  $\mathbf{u}(\cdot) \in \mathbb{U}$  zdefiniowanych na ograniczonym, choć nieokreślonym horyzoncie czasu  $T$ .

Danymi do zadania planowania ruchu manipulatora mobilnego są:

- model matematyczny platformy wyrażony jako bezdryfowy układ sterowania postaci

$$\dot{\mathbf{q}}(t) = \mathbf{G}(\mathbf{q}(t))\mathbf{u}(t) = \sum_{i=1}^m \mathbf{g}_i(\mathbf{q})u_i, \quad \dim \mathbf{q} = n > \dim \mathbf{u} = m, \quad (3.1)$$

gdzie  $\mathbf{g}_i$  są analitycznymi polami wektorowymi, Zakładamy, że układ (3.1) jest sterowalny w krótkim czasie, czyli spełnia warunek rzędu algebry Liego (2.5).

- sterowania  $\forall t \mathbf{u}(t)$  platformy są nieograniczone,
- kinematyka prosta manipulatora zadana jest przez

$$\mathbf{k}_m : \mathbb{X} \rightarrow \mathbb{Y}, \quad (3.2)$$

z dodatkowym założeniem, że układ lokalny manipulatora pokrywa się z układem globalnym.

- odwzorowanie wyjścia

$$\mathbf{k} : \mathbb{Q} \times \mathbb{X} \mapsto \mathbb{Y} \subset \mathbb{SE}(3), \quad \mathbf{k}(\mathbf{q}, \mathbf{x}) = \mathbf{y}, \quad (3.3)$$

wiążące pozę platformy i konfigurację manipulatora z położeniem efektora manipulatora (tożsamego z efekтором manipulatora mobilnego),

- istnienia przeszkód nakłada ograniczenia występujące w przestrzeni zadaniowej w postaci obszarów zabronionych  $\mathbf{O} \subset \mathbb{Y}$ ,
- zadana jest poza początkowa platformy  $\mathbf{q}_0$  i konfiguracja manipulatora  $\mathbf{x}_0$ , a także punkt docelowy manipulatora mobilnego w przestrzeni zadaniowej  $\mathbf{y}_d \in \mathbb{Y}$ .

Zadanie planowania ruchu manipulatora mobilnego polega na wyznaczeniu sterowań platformy  $\mathbf{u}(\cdot)$  i takiego przebiegu trajektorii manipulatora  $\mathbf{x}(\cdot)$  na przedziale czasu  $[0, T]$ , zainicjowanych w punkcie  $(\mathbf{q}_0, \mathbf{x}_0)$ , by trajektoria była bezkolizyjna:

$$\forall t \in [0, T] \quad \mathbf{B}(\mathbf{q}(t), \mathbf{x}(t)) \cap \mathbf{O} = \emptyset, \quad (3.4)$$

gdzie  $\mathbf{B} \subset \mathbb{R}^3$  oznacza bryłę w układzie globalnym odpowiadającej wektorowi  $\mathbf{q}(t), \mathbf{x}(t)$  utworzoną przez manipulator mobilny w chwili  $t$ ,

oraz punkt docelowy został osiągnięty przez efektor:

$$\mathbf{k}(\mathbf{q}(T), \mathbf{x}(T)) = \mathbf{y}_d. \quad (3.5)$$

Opcjonalnie, etapy ruchu manipulatora mobilnego mogą podlegać optymalizacji odpowiednio dobranymi funkcjami kryterialnymi.

Dyskusja danych i założeń:

**układ globalny** Jakkolwiek umiejscowienie układu globalnego można dobierać dowolnie, warto jednak wybrać jego lokalizację tak, by łatwo interpretować położenia i orientacje efektora manipulatora. Zakładając, że platforma mobilna porusza się po płaszczyźnie, proponujemy umieścić początek układu globalnego w punkcie rzutu prostopadłego punktu docelowego efektora na płaszczyznę ruchu platformy. Wybrany początek układu globalnego jest określony jednoznacznie, a ruch w najbardziej interesującym obszarze manipulacji jest wykonywany przy relatywnie małych wartościach współrzędnych wektora położenia uogólnionego (położenie i orientacja) w przestrzeni zadaniowej;

**przestrzeń zadaniowa** W przypadku manipulatora o pełnych możliwościach ruchowych przestrzeń zadaniową jest pełnowymiarowym podzbiorem specjalnej grupy euklidesowej  $SE(3)$ . Gdy możliwości ruchowe celowo ograniczamy - stanowi podzbiór jej właściwej podgrupy (np.  $\mathbb{R}^3$ );

**model platformy** model platformy (3.1) wynika ze spełnienia ograniczeń nieholonomicznych w formie Pfaffa:

$$\mathbf{A}(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0}, \quad (3.6)$$

gdzie macierz  $\mathbf{A}$ , rozmiaru  $r \times n$ ,  $r < n$ , jest pełnego rzędu. Ewentualnie renumerując współrzędne wektora  $\mathbf{q}$  można przekształcić zależność (3.6) do postaci

$$\mathbf{A}_1(\mathbf{q})\dot{\mathbf{q}}^1 + \mathbf{A}_2(\mathbf{q})\dot{\mathbf{q}}^2 = \mathbf{0} \quad (3.7)$$

z macierzą kwadratową  $\mathbf{A}_1$  ( $r \times r$ ) pełnego rzędu i współrzędnymi wektora  $\mathbf{q}$  zgrupowanymi w dwa wektory  $\mathbf{q}^1$  i  $\mathbf{q}^2$ . Przekształcając (3.7), użyjemy modelu (3.1), z

$$\mathbf{G} = \begin{bmatrix} -\mathbf{A}_1^{-1}(\mathbf{q}) \\ \mathbf{I}_{n-r} \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} \mathbf{q}^1 \\ \mathbf{q}^2 \end{bmatrix}, \quad \mathbf{u} = \dot{\mathbf{q}}^2. \quad (3.8)$$

**model manipulatora** manipulator z założenia nie może mieć elastyczności w przegubach, gdyż wtedy zależność pomiędzy położeniem przegubów a elementem napędzanym (wał silnika) jest dynamiczna. Sztywność ogniw manipulatora jest także wymagana, gdyż w przypadku ogniw elastycznych uzyskanie statycznej zależności wyjściowej (3.3) jest w typowym przypadku niemożliwe;

**bezpieczeństwo ruchu** Warunek (3.4) w praktyce może okazać się zbyt tolerancyjny, dopuszcza bowiem dowolnie bliskie zbliżenie manipulatora mobilnego do przeszkody. W bardziej restrykcyjnej wersji warunku (3.4) żądamy, by zachowany był warunek bezpiecznej odległości:

$$\forall t \in [0, T] \quad \|\mathbf{B}(\mathbf{q}(t), \mathbf{x}(t)) - \mathbf{O}\| \geq d_0 > 0, \quad (3.9)$$

gdzie  $d_0$  jest założonym marginesem bezpieczeństwa. Odległość brył w zależności (3.9) rozumiemy klasycznie

$$\|\mathbf{B}, \mathbf{O}\| = \min_{\mathbf{a} \in \mathbf{B}, \mathbf{b} \in \mathbf{O}} \|\mathbf{a} - \mathbf{b}\|.$$

**ograniczenia na sterowanie platformy** Przyjęcie braku ograniczeń na sterowanie łatwo osłabić, jeśli tylko wynikowe sterowania są ograniczone (a takie być muszą ze względu na ich fizyczną realizowalność). Przez wydłużenie horyzontu czasowego i liniowe przeskalowanie sterowań platformy można zrealizować ten sam tor ruchu, który odpowiada sterowaniom platformy bez skalowania. Podobna uwaga dotyczy części manipulacyjnej, choć explicite nie występują w niej sterowania;

**warunek rzędu algebry Liego** jest warunkiem kluczowym. Dla platform spotykanych w praktyce jest on spełniony przez co najwyżej kilka pierwszych warstw pól wektorowych utworzonych z generatorów układu  $\mathbf{g}_i(\mathbf{q})$ ,  $i = 1, \dots, m$ . Znaczenie tego warunku polega na fakcie możliwości dowolnej reorientacji platformy w małym (otwartym) otoczeniu jej konfiguracji bieżącej. Własność maksymalnego rzędu algebry Liego ma istotne znaczenie w dowodzie zbieżności algorytmu planowania ruchu.

### 3.1. Idea rozwiązania zadania planowania ruchu manipulatora mobilnego

Zakładamy, że ruch manipulatora mobilnego będzie przebiegał, i będzie planowany, w trzech fazach:

1. ustawienie manipulatora w konfiguracji transportowej. Etap charakteryzujący się bezruchem platformy i sprowadzeniem manipulatora do konfiguracji optymalnej w sensie odpowiednio skonstruowanej funkcji kryterialnej, która ma uwzględnić minimalizację zajmowanej objętości przez manipulator mobilny i/lub odpowiednie usytuowanie środka masy manipulatora względem środka masy platformy;
2. etap transportowy, w którym manipulator pozostaje nieruchomy, a platforma, w iteracyjnym procesie, dąży do bieżącego celu wyznaczonego przez hipotetyczny manipulator o końcu efektora umieszczonym w punkcie docelowym ruchu;
3. faza manipulacyjna. Platforma zostaje unieruchomiona przez zastosowanie podpór odciażających koła platformy, a manipulator osiąga docelowe położenie w przestrzeni zadaniowej.

Argumenty, natury fizycznej i obliczeniowej, uzasadniające trójpodział zadania planowania są następujące (w nawiasie numer fazy ruchu):

- (1,2,3) jednoczesne stosowanie napędu manipulatora i platformy może nie być wskazane (zasilanie z jednego źródła, napęd hybrydowy, spalinowo-elektryczny);
- (1) przemieszczanie nieruchomego manipulatora powinno odbywać się w takiej konfiguracji, która zapewni symetryczne obciążenie kół platformy, a także minimalizację prawdopodobieństwa kolizji z przeszkodami. Ze względu na bezpieczeństwa manewru (bezkolizyjność) celowe wydaje się minimalizowanie uwypuklenia manipulatora mobilnego;
- (2) wykonywanie jednoczesnego ruchu przez manipulator i platformę może prowadzić do uszkodzenia łożysk manipulatora, przez wibracje i udary nieuniknione podczas ruchu platformy. Unieruchomienie manipulatora w fazie drugiej umożliwia jego zaparkowanie przez zastosowanie mechanicznego unieruchomienia (hamulce);
- (2) w przypadku ruchu manipulatora w fazie drugiej, możliwe jest zmiana położenia środka masy manipulatora, a tym samym niesymetryczne obciążenie kół platformy, którego efektem (nawet przy jednorodnym podłożu) może być niedokładność realizacji zaplanowanej trajektorii;
- (3) dla zachowania energetycznej efektywności ruchu konieczność jest stosowanie lekkich i kompaktowych (relatywnie mała platforma w stosunku do manipulatora) konstrukcji nośnych. Konieczność napędzania ciężkich konstrukcji wymaga stosowania dużych akumulatorów, które dodatkowo obciążają platformę. Unieruchomienie platformy przed początkiem fazy trzeciej ma następujące cele,
  - a) przez rozstaw podpór zwiększa się zasięg działania manipulatora (ruch środka masy manipulatora może odbywać się w szerszym zakresie nie powodując utraty przyczepności platformy do podłoża),
  - b) podczas ruchu manipulatora bez podpór i z miękkimi kołami możliwe jest pogorszenie dokładności wykonywanych operacji wynikających z ugięcia kół (amortyzatory, ogumienie)
  - c) możliwe jest przemieszczanie cięższych ładunków.
- (1,2,3) jakkolwiek planowanie ruchu odbywa się na poziomie kinematycznym, należy wziąć pod uwagę, że realizacja zadania planowania będzie wymagała także uwzględnienie dynamiki manipulatora mobilnego. Podział na fazy ruchu stanowi pewien rodzaj odsprężenia podukładów manipulatora mobilnego, rozumianego jako możliwość potraktowania nieruchomego w danej fazie obiektu jako obciążenia statycznego obiektu znajdującego się w ruchu. Może się zdarzyć, że należy rozważać sprzężenie podukładów, nawet wtedy, gdy teoretycznie jeden z nich powinien znajdować się w spoczynku. Dzieje się tak, na przykład, w przypadku gdy szybki ruch manipulatora może spowodować zerwanie kontaktu kół platformy z podłożem. Przypadek taki należy uwzględnić po etapie

planowania ruchu podczas generacji sterowań fizycznych uwzględniających dynamiczne efekty ruchu;

(1,2,3) dekompozycja zadania planowania ruchu manipulatora mobilnego zwiększa liczbę zadań, jednak są to zadania o mniejszej liczbie wymiarów, zatem łatwiejsze do rozwiązania. Praktycznie złożoność wszystkich zadań technicznych nie jest liniowa ze względu na liczbę wymiarów (najczęściej wielomianowa lub NP-trudna). Zatem korzystne, z obliczeniowego punktu widzenia, jest rozwiązywanie nawet wielu zadań niskowymiarowych niż jednego w wyższym wymiarze.

Zadanie pierwszej fazy jest typowym zadaniem optymalizacji statycznej polegającej na wyznaczeniu optymalnej konfiguracji manipulatora na platformie połączonej z planowaniem ruchu manipulatora do tej konfiguracji. W fazie drugiej rozwiązujemy zadanie planowania trajektorii platformy nieholonomicznej o dynamicznie zmieniającym się celu ruchu, natomiast faza trzeciej jest planowaniem ruchu manipulatora do zadanego punktu docelowego platformy mobilnej.

Zaproponowana dekompozycja zadania planowania ma także wady, które wymieniamy poniżej:

- optymalizacja każdej z faz ruchu niekoniecznie prowadzi do ruchu globalnie optymalnego,
- w przypadku ruchu krótkich niecelowe może okazać się składanie manipulatora do pozycji transportowej, a następnie jego rekonfiguracja w pobliżu docelowego punktu pracy,
- w przypadku bardzo trudnych środowisk (wiele przeszkód o skomplikowanych kształtach zajmujących objętość porównywalną z przestrzenią wolną (gdy planujemy ruch w ograniczonym podzbiore  $\mathbb{R}^3$ )) może się okazać, że rozwiązanie zadania planowania może wymagać jednoczesnego ruchu zarówno manipulatora jak i platformy mobilnej. Zadania dla takich środowisk stanowią inspirację dla projektowania dedykowanych algorytmów planowania ruchu. Przykładami takich środowisk są techniki planowania ruchu w labiryntach [47] czy klasyczny problem transportu fortepianu (*piano-mover problem* [30]).



## 4. Przegląd metod planowania ruchu

Zadanie planowania ruchu robotów rozważane było przez wielu badaczy. Ponieważ zagadnienie jest klasyczne ma swoją już kilkudziesięcioletnią historię i doczekało się opracowań podręcznikowych [28, 58, 77, 87]. W rozdziale niniejszym przedstawiono przegląd różnorodnych metod zaproponowanych w literaturze robotycznej z uwzględnieniem ich wad i zalet. Zanim przytoczone zostaną ich detale, warto pokusić się o klasyfikację zadań planowania ruchu, gdyż rozważana klasa zadań silnie koreluje z techniką ich rozwiązywania.

By przystąpić do rozwiązania zadania planowania ruchu należy określić model obiektu dla którego projektowana jest metoda planowania ruchu. Główna linia podziału wyznacza holonomiczność bądź nieholonomiczność obiektu, którym może być zarówno platforma mobilna jak i manipulator. Dla manipulatora mobilnego rozważanego w niniejszej rozprawie, możliwych są cztery warianty: nieholonomiczna/holonomiczna platforma z nieholonomicznym/holonomicznym manipulatorem na jej pokładzie. We wczesnych pracach podstawowym obiektem była holonomiczna platforma (tzw. *omni-directional robot*) i holonomiczny manipulator. Literatura dotycząca planowania ruchu robotów holonomicznych nawet w środowisku kolizyjnym jest bogata [5, 39, 75, 79] i w większości przypadków polega na wykorzystaniu odpowiednio skonstruowanych funkcji nawigacyjnych. Obecnie najczęściej spotykane w rozważaniach teoretycznych (i zarazem najbardziej praktyczne) jest połączenie nieholonomicznej platformy z holonomicznym manipulatorem. Nieholonomiczne platformy rozpatruje się bądź jako konkretne egzemplifikacje układów jezdnych (najczęściej modelowane jako jednokołowy robot mobilny, bądź samochód kinematyczny) bądź zapisuje się ograniczenia ruchu w postaci Pfaffa, z których łatwymi przekształceniami dochodzi się do bezdryfowego układu sterowania. Istnieją układy robotyczne wiodące do tego typu układu, a istotnie różne od platform mobilnych. Są to roboty szybujące, w których ograniczenia nieholonomiczne wynikają z zasady zachowania momentu pędu, a nie z braku poślizgu jak to ma miejsce w układach jezdnych. Do tej klasy układów należy także model spadającego, w polu grawitacyjnym kota. Układami bezdryfowymi są także układy o pewnych dodatkowych własnościach (np. nilpotentność w przypadku obiektu znanego z literatury jako integrator Brocketta [3]). Bezdryfowy układ sterowania jest uzyskiwany, gdy rozważania są przeprowadzane na poziomie kinematycznym. Gdy, dodatkowo, wzięta jest także pod uwagę dynamika obiektu, sterowany układ zawiera także dryf. Panuje jednak wśród badaczy przekonanie [27], że najbardziej istotne trudności ze sterowaniem układów nieholonomicznych leżą już na poziomie kinematycznym (układ bezdryfowy), a wynikają z istotnie mniejszej liczby sterowań niż wynosi wymiar przestrzeni stanu. Pojawiają się także ostatnio prace poszerzające (osłabiające) założenia leżące u podstaw wyprowadzenia modelu bezdryfowego. W tej klasie układów warto wyróżnić układy z poślizgiem. W naturalny sposób są także osłabiane założenia o znajomości modelu (parametryczna, strukturalna), które, w skrajnej postaci, prowadzą do negacji potrzeby modelowania robota i sterowania behawioralnego. Wydaje się, że ciągle niedocenionym aspektem na gruncie robotyki (acz badanym przez mechaników) jest oddziaływanie koła z podłożem na różnych rodzajach gruntów, ponieważ większość współczesnych platform jezdnych pracuje w środowisku zestrukturalizowanym (szpitale, hale fabryczne, laboratoria uniwersyteckie). Jednak należy oczekiwać wzrostu zainteresowania zastosowaniami robotów (manipulatorów) mobilnych w środowiskach otwartych. W ten sposób naturalnie dochodzimy do opisu środowiska robota, które w starszych pracach było bezkolizyjne, gdyż sterowanie obiektem nieholonomicznym obiektem stanowiło nie lada wyzwanie. Środowisko kolizyjne może być znane, częściowo lub całkowicie nieznane.

Przeszkody mogą być statyczne (jedynym obiektem ruchomym jest robot) lub dynamiczne. W środowiskach dynamicznych wyróżniamy obiekty o znanej charakterystyce (przykładem obiekt poruszający się po znanej trasie według z góry określonego scenariusza), lub częściowo znanej (gdy znamy jedynie stan poruszającego się obiektu i ewentualnie jego charakterystyki, typu maksymalna prędkość, przyspieszenie, zwrotność mierzona promieniem minimalnego promienia skrętu). Klasycznym i zarazem najtrudniejszym zadaniem związanym z uwzględnieniem przeszkód w otoczeniu robota jest sprawdzanie kolizyjności [56]. Ponieważ robot się porusza, test ten musi być wykonywany często. Więc najważniejsza jest optymalizacja złożoności obliczeniowej sprawdzania bezkolizyjności robota, gdyż obliczenia wykonywane są zwykle w trybie czasu rzeczywistego. Dla przyspieszenia obliczeń wymagany jest właściwy dobór reprezentacji przeszkód. Zwykle dokonuje się aproksymacji rzeczywistych obiektów przy pomocy obiektów elementarnych, charakteryzujących się łatwym opisem analitycznym i efektywnością w określaniu odległości między nimi. Jako obiekty elementarne wykorzystuje się wielościanny wypukłe, elipsoidy, kule, stożki. W wielu przypadkach dąży się do rozpatrywania kolizyjności w przestrzeniach niskowymiarowych, najczęściej dwuwymiarowych. Wiele interesujących zagadnień powstaje już na etapie określania środowiska robota. W zakres tych zagadnień wchodzi dobór środków i metod zbierania informacji o otoczeniu robota, efektywna akwizycja danych i budowa mapy otoczenia robota.

Przejdźmy teraz do klasyfikacji metod planowania ruchu robota. Głównym podziałem różnicującym zadania planowania jest zakres działania metody. Wyróżniamy tu planowanie globalne, w którym projektowana jest cała trajektoria (ścieżka) robota, oraz lokalne – gdzie określany jest jedynie najbliższy fragment trajektorii wokół bieżącego stanu robota (a finalna trajektoria powstaje jako sklejenie lokalnych fragmentów). Cechą metod globalnych jest uwzględnianie globalnych informacji o otoczeniu robota, co umożliwia stawianie zadań optymalizacji globalnej. Jednak metody globalne mają także wiele wad, gdyż są bardziej złożone obliczeniowo od lokalnych oraz ich wyniki częściowe są całkowicie bezużyteczne w przypadku zmian w środowisku robota. Ponadto, metody globalne wymagają pełnej znajomości środowiska, co wyklucza planowanie ruchu w środowiskach otwartych i ubogo osensorowanych (aspekt ekonomiczny). Następny podział metod wynika z przyjętego modelu. Mamy więc metody ogólne, działające dla szerokiej klasy modeli i metody specjalizowane, które istotnie wykorzystując własności modelu – planują ruch niewielu (aczkolwiek ważnych praktycznie) typów robotów. Ze względu na technikę rozwiązania zadania planowania, wyróżniamy metody analityczne, w których obliczenia prowadzone są przez przekształcanie wzorów, numeryczne – gdy intensywnie stosowane są procedury numeryczne, oraz hybrydowe, w których obliczenia prowadzi się tak długo analitycznie jak to jest możliwe, a w końcowym etapie – numerycznie. Klasycznie zadanie planowania ruchu polega na zaplanowaniu trajektorii (ścieżki) robota, która następnie staje się daną wejściową dla układu sterowania, którego celem jest najbardziej wierne odtworzenie zadanej trajektorii. Dlatego zwykło się uważać, że zadanie planowania można przeprowadzić w trybie off-line. Istnieją także podejścia, w których zachodzi różnica między planowaniem ruchu a sterowaniem. W tych metodach projektowane jest prawo sterowania w sposób jawny uzależniające sygnały sterujące od środowiska robota i jego modelu. Tradycyjnie można podzielić metody ze względu na postawione zadanie na: optymalne (gwarantujące optymalizację przyjętego wskaźnika jakości), sub-optymalne – generujące statystycznie lepsze rozwiązania od typowych (lub gdy optymalizacja występuje, lecz trudno udowodnić optymalność otrzymanego rezultatu), oraz bez optymalizacji. Optymalizacja często występuje w problemach nisko-wymiarowych. Dla zadań o bardzo dużej liczbie wymiarów, trudno znaleźć efektywnie rozwiązanie przy pomocy algorytmów deterministycznych. Dlatego chętnie stosowane są algorytmy stochastyczne (losowe). W układach złożonych, składających się z dwóch (więcej) podukładów, często stosowane są techniki dekompozycji zadania planowania na dwa (kilka) podzadania. Uzasadnieniem teoretycznym dla techniki dekompozycji jest fakt znacznego zmniejszenia złożoności obliczeniowej, gdy problem wielowymiarowy jest zastępowany kilkoma problemami o mniejszej wymiarowości. W przypadku manipulatora

mobilnego, w którym podukłady charakteryzują się istotnie różnymi możliwościami ruchowymi, planowanie ruchu manipulatora (jako łatwiejszego w sterowaniu) jest podporządkowane projektowaniu trajektorii platformy mobilnej.

Dalej przedstawiono metody planowania ruchu zarówno platform mobilnych jak i manipulatorów, korzystające z różnorodnych technik poszukiwania rozwiązania zadania.

#### 4.1. Metoda planowania ruchu nilpotentnych układów nieholonomicznych

W artykule [43] Lafferriere i Sussmann zaproponowali algorytm planowania ruchu dla układów bezdryfowych (2.4), które są dodatkowo nilpotentne. Sformułowanie zadania planowania jest klasyczne: znaleźć trajektorię przeprowadzającą układ z zadanego stanu początkowego do końcowego w przestrzeni bezkolizyjnej. Najpierw system sterowany podlega rozszerzeniu o pola wektorowe wyższych rzędów pomnożone przez sterowania wirtualne, tak by wszystkie pola (oryginalne i dodatkowe) spełniały warunek rzędu LARC (2.5), czyli rozpięły przestrzeń stanu  $\mathbb{R}^n$ . Zapis układu rozszerzonego jest następujący:

$$\dot{\mathbf{q}} = v_1 \mathbf{g}_1(\mathbf{q}) + \dots + v_m \mathbf{g}_m(\mathbf{q}) + v_{m+1} \mathbf{g}_{m+1}(\mathbf{q}) + \dots + v_r \mathbf{g}_r(\mathbf{q}), \quad (4.1)$$

gdzie  $\mathbf{g}_1(\mathbf{q}), \dots, \mathbf{g}_m(\mathbf{q})$  są oryginalnymi polami wektorowymi generującymi układ bezdryfowy, natomiast  $\mathbf{g}_{m+1}(\mathbf{q}), \dots, \mathbf{g}_r(\mathbf{q})$  są polami wektorowymi rozszerzenia uzyskiwanymi przez rekurencyjne zastosowanie operatora nawiasu Liego do generatorów. Pola wektorowe rozszerzenia należy wybrać z bazy jednomianów Liego (np. bazy Ph. Halla). Oczywiście spełniony jest warunek  $r \geq n > m$ . Działanie opisywanej metody jest dwuetapowe: najpierw wyznaczenie sterowania wirtualne  $v_i, i = 1, \dots, r$  dla systemu rozszerzonego (4.1) zapewniające przemieszczenie stanu między zadanymi konfiguracjami brzegowymi, a następnie wyliczane są pierwotne sterowania  $u_1, \dots, u_m$  zachowujące tę własność. W pierwszym etapie, w zależności od wyboru współrzędnych Ph. Halla “w przód” (ang. forward) lub “do tyłu” (ang. backward), wprowadzany jest operator

$$\mathbf{S}(t) = \exp(h_1(t)\mathbf{g}_1) \exp(h_2(t)\mathbf{g}_2) \dots \exp(h_r(t)\mathbf{g}_r), \quad (4.2)$$

który definiuje przemieszczenie stanu po sekwencji działań pól wektorowych, a  $\exp(h(t)\mathbf{Z})$  oznacza strumień pola  $\mathbf{Z}$  działający przez czas  $h(t)$ . Warunek brzegowy  $\mathbf{S}(0) = \mathbf{I}$  oznacza, że przemieszczenie stanu układu w chwili  $t = 0$  jest zerowe. Równanie (4.1) przybiera postać

$$\dot{\mathbf{S}}(t) = \mathbf{S}(t)(v_1 \mathbf{g}_1(\mathbf{q}) + \dots + v_m \mathbf{g}_m(\mathbf{q}) + v_{m+1} \mathbf{g}_{m+1}(\mathbf{q}) + \dots + v_r \mathbf{g}_r(\mathbf{q})). \quad (4.3)$$

Po zróżniczkowaniu (4.2), podstawieniu do (4.3) oraz przemieszczeniu  $\mathbf{S}(t)$  na lewą stronę równości (uwzględniając, że  $\mathbf{S}^{-1}(t)$  jest odczytywaniem sekwencji strumieni (4.2) wspak ze zmienionym znakiem każdego  $h_i(t)$ ) oraz porównaniu współczynników występujących przy identycznych polach  $\mathbf{g}_i$ , otrzymujemy układ równań różniczkowych ze względu na funkcje  $h_i(t)$ , z warunkiem początkowym  $h_i(0) = 0, i = 1, \dots, r$ . Po ustaleniu horyzontu czasowego sterowania  $T$  oraz wybraniu dowolnej, acz odpowiednio gładkiej, trajektorii łączącej konfiguracje brzegowe, możliwe jest wyznaczenie przebiegu funkcji  $h_i(t)$  zapewniających przemieszczenie stanu do zadanego celu ruchu.

W drugim, trudniejszym etapie, na podstawie formuł Campbella-Bakera-Hausdorffa-Dynkina (w wersji dyskretnej) wyznaczane są sterowania  $u_i(t), i = 1, \dots, m$  generujące kolejne elementy występujące w znanej już sekwencji (4.2). Przykładowo, dla układu o dwóch generatorach  $\mathbf{g}_1, \mathbf{g}_2$  element  $\exp(h_3(t)[\mathbf{g}_1, \mathbf{g}_2])$  jest generowany przy pomocy sterowań kawałkami stałych w następującej sekwencji (czytanej od lewej strony ku prawej)

$$\exp(h_3(t)[\mathbf{g}_1, \mathbf{g}_2]) = \exp(\sqrt{h_3(t)}\mathbf{g}_1) \exp(\sqrt{h_3(t)}\mathbf{g}_2) \exp(-\sqrt{h_3(t)}\mathbf{g}_1) \exp(-\sqrt{h_3(t)}\mathbf{g}_2).$$

Metoda Lafferriera i Sussmanna nie wymaga, oprócz wymienionych, dodatkowych założeń. Metoda nie generuje trajektorii optymalnych. Dla układów nie posiadającej własności nilpotentności możliwe jest stosowanie metody na małych kawałkach trajektorii, na których dość wiernie można aproksymować dowolny układ bezdryfowy układem nilpotentnym. Zadanie planowania ruchu tą metodą ilustrowano przykładami dla jednokołowego robota mobilnego w pracy [42] oraz w artykule [41] dla samochodu kinematycznego.

Sussmann [81] podjął próbę uogólnienia podejścia dla układów z dryfem:

$$\dot{\mathbf{q}} = \mathbf{g}_0(\mathbf{q}) + \sum_{i=1}^m \mathbf{g}_i(\mathbf{q}) u_i \quad (4.4)$$

rozważanych wokół punktu równowagi,  $\mathbf{g}_0(\mathbf{q}) = 0$ . Zaproponowana modyfikacja metoda rozszerzenia dynamicznego układu (4.4) okazała się skuteczna, jednak warunki sterowalności są znacznie mocniejsze niż w przypadku zadania bez dryfu.

W pracy [82] rozpatrzono zadanie wyznaczania rzeczywistych sterowań  $\mathbf{u}$  na podstawie znajomości sterowań rozszerzonych  $\mathbf{v}$  dla innych sterowań niż kawałkami stałych. Przyjęto bazę harmoniczną, dla której rozwiązano zadanie planowania ruchu dla systemu o dwóch wejściach i pięciu wyjściach. Podano również szkic algorytmu śledzenia trajektorii, którego parametrem jest maksymalna odległość trajektorii rzeczywistej od pożądanej.

Oprócz bazy Ph. Halla do rozwiązywania zadania planowania ruchu używane są również inne bazy, np. baza Lyndona. Bazie Lyndona została poświęcona praca [54], gdzie oprócz szczegółowego omówienia formalnej definicji bazy, przedstawiono algorytm generowania jej elementów. W artykule [34] Jacob przedstawił klasyczne podejście wykorzystujące bazę Lyndona. Zaprezentowano wprowadzenie do baz Lyndona oraz metodę algorytmicznego wyznaczania wielomianów algebry Liego z przykładami. Niech dany będzie alfabet

$$\mathbf{Z}' = \{z_1, z_2, \dots, z_m\}, \quad (4.5)$$

gdzie każda składowa  $z_j \in \mathbf{Z}$  odpowiada sterowaniu  $u_i(\cdot)$ . Niech

$$\mathbf{Z} = (\{z_1, z_2, \dots, z_m\}, <) \quad (4.6)$$

oznacza uporządkowany alfabet. Odwrotny porządek leksykograficzny na zbiorze  $\mathbf{Z}^* \subset \mathbf{Z}$  zdefiniowany jest następująco:

$$u < v \iff \begin{cases} \text{albo} & (i) & v = fu \text{ dla pewnego } f \in \mathbf{Z}^*, \\ \text{albo} & (ii) & u = fxw, v = gyw, \quad x < y, \quad x, y \in \mathbf{Z}, \quad f, g \in \mathbf{Z}^*. \end{cases} \quad (4.7)$$

Słowo  $w \in \mathbf{Z}^*$  jest odwrotnym słowem Lyndona wtedy i tylko wtedy gdy jest mniejsze od wszystkich swoich lewych składników. Dowolne słowo Lyndona  $b \in \mathbf{L}$  może być zapisane jako

$$\begin{cases} \text{albo} & b \in \mathbf{Z} & b \text{ jest literą} \\ \text{albo} & b = ml & l < m, \quad m, l \in \mathbf{L}. \end{cases} \quad (4.8)$$

Dla najdłuższego słowa  $m$  spełniającego własność (4.8) para  $st(b) = (m, l)$  jest nazywana standardową faktoryzacją  $b$ . Dla każdego odwrotnego słowa  $b$  przypisuje się wielomian Liego  $A_b$ :

$$\begin{cases} A_b = b & \text{jeśli } b \in \mathbf{Z} \\ A_b = [A_m, A_l] & \text{jeśli } st(b) = (m, l). \end{cases} \quad (4.9)$$

Rodzina  $\{A_b | b \in \mathbf{L}\}$  jest bazą (odwrotna baza Lyndona) w przestrzeni wielomianów Liego. Bazę Lyndona do stopnia 5, dla układu o dwóch wejściach zamieszczono w tablicy 4.1. Porównując bazę Lyndona z bazą Ph. Halla (tab. 2.1) łatwo zauważyć, że elementami baz są w obu przypadkach jednomiany

Tablica 4.1. Elementy bazy Lyndona

0	$A_1 = \mathbf{g}_1$
1	$A_2 = \mathbf{g}_2$
10	$A_{21} = [A_2, A_1]$
100	$A_{211} = [A_{21}, A_1]$
110	$A_{221} = [A_2, A_{21}]$
1000	$A_{2111} = [A_{211}, A_1]$
1100	$A_{2211} = [A_{221}, A_1]$
1110	$A_{2221} = [A_2, A_{221}]$
10000	$A_{21111} = [A_{2111}, A_1]$
11000	$A_{22111} = [A_{2211}, A_1]$
10100	$A_{21211} = [A_{21}, A_{211}]$
11100	$A_{22211} = [A_2, A_{2211}]$
11010	$A_{22121} = [A_{221}, A_{21}]$
11110	$A_{22221} = [A_2, A_{2221}]$

Liego, natomiast różna jest ich postać (tożsamość Jacobiego zezwala na pewną dowolność w odrzucaniu jednego z jednomianów Liego wchodzących w skład tożsamości).

Jako sterowania przyjęto przebiegi kawałkami stałe. Rozszerzeniem uwzględniającym także sterowania wielomianowe jest publikacja [35]. Metoda planowania ruchu układów nieholonomicznych wykorzystująca bazę Lyndona jest analogiczna jak w przypadku stosowania bazy Ph. Halla: najpierw wyznaczane jest rozwiązanie w wirtualnych współrzędnych bazy Lyndona, a następnie wyznaczane są rzeczywiste sterowania realizujące te współrzędne. Współczynniki wirtualne wyznaczane są poprzez całkowanie odpowiednich wyrazów sterowań, a następnie wyliczane są wartości sterowań rzeczywistych. W artykule podano metodę określania sterowań stałych, stałych z dryfem oraz wielomianowych.

## 4.2. Metoda sterowań sinusoidalnych

Praca [88] opisuje zastosowanie sterowań sinusoidalnych do planowania ruchu robota typu samochód z przyczepami. Sterowanie sinusoidalne dobrze nadaje się do klasy systemów, które mogą być zapisane w postaci łańcuchowej. Najprostszym przykładem tej klasy jest układ dwuwęściowy  $(u_1, u_2)$  postaci

$$\begin{aligned}
 \dot{q}_1 &= u_1 \\
 \dot{q}_2 &= u_2 \\
 \dot{q}_3 &= q_2 u_1 \\
 &\vdots \\
 \dot{q}_n &= q_{n-1} u_1.
 \end{aligned} \tag{4.10}$$

Algorytm sterowania opiera się na iteracyjnym doprowadzaniu poszczególnych zmiennych stanu do pożądanych wartości docelowych, dzięki okresowości funkcji sinusoidalnych. Zakładając sterowania

$$\begin{aligned}
 u_1 &= \alpha \sin \omega t \\
 u_2 &= \beta \cos k \omega t,
 \end{aligned}$$

oraz dokonując całkowania systemu (4.10) podczas jednego okresu  $T = 2\pi/\omega$ , można zauważyć, że pierwszych  $k+1$  stanów  $(q_1, q_2, \dots, q_{k+1})$  wróci do swojego pierwotnego stanu (zakreślając zamkniętą

pętlę w przestrzeni stanu). Stan  $q_{k+2}$  zmieni się dokładnie o  $q_{k+2}(T) - q_{k+2}(0) = \alpha^k \beta T / (2^k k!)$ . Algorytm sterowania jest iteracyjny. Najpierw dobiera się odpowiednie sterowania stałe, tak aby doprowadzić współrzędne  $q_1$  i  $q_2$  do zadanych wartości. Następnie, zaczynając od  $q_3$ , dobieramy sterowania tak, aby doprowadzić kolejne współrzędne  $q_k$ ,  $k = 3, \dots$  do zadanej wartości docelowej. Należy wykonać  $n - 1$  kroków algorytmu, aby doprowadzić wszystkie  $n$  współrzędne wektora stanu do zadanych wartości.

By określić możliwość przekształcenia układu o postaci

$$\mathbf{q} = \mathbf{g}_1(\mathbf{q})u_1 + \mathbf{g}_2(\mathbf{q})u_2, \quad (4.11)$$

z polami generatorów mających formę:

$$\begin{aligned} \mathbf{g}_1(\mathbf{q}) &= \frac{\partial}{\partial q_1} + \sum_{i=2}^n \mathbf{g}_1^i \frac{\partial}{\partial q_i} \\ \mathbf{g}_2(\mathbf{q}) &= \sum_{i=2}^n \mathbf{g}_2^i \frac{\partial}{\partial q_i} \end{aligned} \quad (4.12)$$

do postaci łańcuchowej określono dystrybucje (rozpięte nad  $\mathbb{R}$ ):

$$\begin{aligned} \Delta_0 &= \text{span}\{\mathbf{g}_1, \mathbf{g}_2, \text{ad}_{\mathbf{g}_1}\mathbf{g}_2, \dots, \text{ad}_{\mathbf{g}_1}^{n-2}\mathbf{g}_2\} \\ \Delta_1 &= \text{span}\{\mathbf{g}_2, \text{ad}_{\mathbf{g}_1}\mathbf{g}_2, \dots, \text{ad}_{\mathbf{g}_1}^{n-2}\mathbf{g}_2\}, \end{aligned} \quad (4.13)$$

gdzie  $\text{ad}_{\mathbf{g}_1}^0 \mathbf{g}_2 = \mathbf{g}_2$  oraz  $\text{ad}_{\mathbf{g}_1}^k \mathbf{g}_2 = [\mathbf{g}_1, \text{ad}_{\mathbf{g}_1}^{k-1} \mathbf{g}_2]$ . Warunek dostateczny stanowi, że jeśli dla pewnego otwartego zbioru  $\mathbf{U}$ ,  $\Delta_0(\mathbf{q}) = \mathbb{R}^n \forall \mathbf{q} \in \mathbf{U}$  i dystrybucja  $\Delta_1$  jest inwolutywna na  $\mathbf{U}$ , wtedy istnieje lokalna transformacja na  $\mathbf{U}$ :  $\xi = \phi(\mathbf{q})$ ,  $\mathbf{u} = \beta(\mathbf{q})\mathbf{v}$ , taka że przekształcony system ma postać łańcuchową. Przypomnijmy, że inwolutywność dystrybucji oznacza, że operacja nawiasu Liego stosowana do dowolnych pól należących do dystrybucji nie generuje pola leżącego poza nią.

Jeżeli dystrybucja  $\Delta_1$  nie jest inwolutywna możliwe jest sprowadzenie układu do przybliżonej postaci łańcuchowej. Dla układu opisanego równaniem (4.11) o polach generatorów mających postać (4.12), rozważane jest przybliżenie rzędu  $p$  pól

$$\begin{aligned} \mathbf{g}_1(\mathbf{q}) &= \tilde{\mathbf{g}}_1(\mathbf{q}) + O(\mathbf{q})^{p+1} \\ \mathbf{g}_2(\mathbf{q}) &= \tilde{\mathbf{g}}_2(\mathbf{q}) + O(\mathbf{q})^{p+1}, \end{aligned} \quad (4.14)$$

gdzie  $O(\mathbf{q})^{p+1}$  jest małą rzędu  $p + 1$ . Dla układu przybliżonego określono, analogicznie do (4.13), dystrybucje

$$\begin{aligned} \tilde{\Delta}_0 &= \text{span}\{\tilde{\mathbf{g}}_1, \tilde{\mathbf{g}}_2, \text{ad}_{\tilde{\mathbf{g}}_1}\tilde{\mathbf{g}}_2, \dots, \text{ad}_{\tilde{\mathbf{g}}_1}^{n-2}\tilde{\mathbf{g}}_2\} \\ \tilde{\Delta}_1 &= \text{span}\{\tilde{\mathbf{g}}_2, \text{ad}_{\tilde{\mathbf{g}}_1}\tilde{\mathbf{g}}_2, \dots, \text{ad}_{\tilde{\mathbf{g}}_1}^{n-2}\tilde{\mathbf{g}}_2\}. \end{aligned} \quad (4.15)$$

Warunek sprowadzalności do przybliżonej postaci łańcuchowej zachodzi, gdy dla pewnego otwartego zbioru  $\mathbf{U}$ ,  $\tilde{\Delta}_0(\mathbf{q}) = \mathbb{R}^n \forall \mathbf{q} \in \mathbf{U}$  oraz dystrybucja  $\tilde{\Delta}_1$  jest inwolutywna na  $\mathbf{U}$ . Wtedy istnieje lokalna transformacja na  $\mathbf{U}$ :  $\xi = (\xi_1, \dots, \xi_n) = \phi(\mathbf{q})$ ,  $\mathbf{u} = \beta(\mathbf{q})\mathbf{v}$ , taka że przekształcony system ma postać łańcuchową  $p$ -go rzędu, czyli jest postaci

$$\begin{aligned} \dot{\xi}_1 &= v_1 \\ \dot{\xi}_2 &= v_2 + O(\xi)^{p+1} \\ \dot{\xi}_3 &= \xi_2 v_1 + O(\xi)^{p+1} \\ &\vdots \\ \dot{\xi}_n &= \xi_{n-1} v_1 + O(\xi)^{p+1}. \end{aligned} \quad (4.16)$$

Oryginalna metoda sterowań sinusoidalnych zakłada sprowadzanie kolejno współrzędnych wektora stanu do ich zadanych wartości. W celu umożliwienia sprowadzania grup współrzędnych do ich docelowych wartości jednocześnie autorzy omawianego artykułu wykorzystują klasyczną technikę planowania ruchu w rozszerzonej bazie Ph. Halla. Sterowania zostają wyznaczone przez odpowiedni dobór współczynników częstotliwościowych.

Podobne zagadnienie było opisane w pracy [57], gdzie autorzy zaproponowali dla systemów łańcuchowych o postaci

$$\begin{aligned} \dot{q}_i &= u_i & i &= 1, \dots, m \\ \dot{q}_{ij} &= q_i u_j & i &> j, \end{aligned} \quad (4.17)$$

następujący algorytm planowania ruchu:

1. dobierz tak sterowania  $u_i$ , aby współrzędne  $q_i$  osiągnęły zadany stan, ignorując (acz wyliczając) ewolucję współrzędnych  $q_{ij}$ ,
2. wykorzystując sinusoidy o całkowitych częstotliwościach wyznacz sterowania doprowadzające  $q_{ij}$  do oczekiwanych wartości. Na przykład sterowanie

$$u_i(t) = \sum_{k>0} (a_{ik} \sin kt + b_{ik} \cos kt) \quad (4.18)$$

zmienia stan na horyzoncie czasowym  $[0, 2\pi]$  według zależności

$$q_{ij}(2\pi) = q_{ij}(0) + \left( \sum_{k>0} \frac{a_{jk}b_{ik} - a_{ik}b_{jk}}{k} \right) \pi, \quad (4.19)$$

nie zmieniając (po okresie sterowania) wartości  $q_i$ ,  $i = 1, \dots, m$ .

Na przykładzie robota skaczącego o jednej nodze zilustrowano poprawność algorytmu. Następnie rozważono system łańcuchowy drugiego rzędu

$$\begin{aligned} \dot{q}_i &= u_i & i &= 1, \dots, m \\ \dot{q}_{ij} &= q_i u_j & i &> j \\ \dot{q}_{ijk} &= q_{ij} u_k & & \text{(modulo tożsamość Jakobiego)} \end{aligned} \quad (4.20)$$

i zaproponowano algorytm planowania ruchu w istocie podobny do uprzednio opisanego:

1. dobierz sterowania  $u_i$ , aby współrzędne  $q_i$  osiągnęły zadany stan. Działanie powoduje dryf pozostałych współrzędnych,
2. wykorzystując sinusoidy o całkowitych częstotliwościach wyznacz sterowania doprowadzające  $q_{ij}$  do zadanych wartości. Jeśli wejście  $i$ -te ma częstotliwość  $\omega_i$  wtedy ewolucja współrzędnej  $q_{ij}$  będzie zależna od składników harmoniczných o częstotliwościach  $\omega_i \pm \omega_j$ . Przez odpowiedni dobór wejść łatwo uzyskać zerowe częstotliwości (czyli stałe, które po całkowaniu po okresie - w przeciwieństwie do wyrażeń harmoniczných - się nie zerują), czyli wygenerować zmianę pewnych współrzędnych pozostawiając pozostałe (po okresie) niezmiennie,
3. używając ponownie sterowań sinusoidalnych wygeneruj ruch w kierunkach  $q_{ijk}$ , pozostałe (poprzednio ustawione) składowe stanu mają zataczać pętlę gwarantującą, że osiągnięte przez nie wartości docelowe się nie zmieniają po okresie sterowania. Należy odpowiednio dobrać częstotliwości, tak aby  $\omega_i + \omega_j \neq 0$  i jednocześnie wyrażenie  $\omega_i \pm \omega_j \pm \omega_k$  miało zerowy składnik częstotliwości.

W podstawowej wersji algorytm wymaga oddzielnego powtórzenia kroku (3) dla każdej ze składowych  $q_{ijk}$ . Możliwe jest także wygenerowanie ruchu w kilku składowych jednocześnie używając liniowej kombinacji sterowań i rozwiązując równanie wielomianowe na współczynniki wypadkowego sterowania. Przykładem ilustrującym algorytm sterowań sinusoidalnych w postaci (4.20) była generacja ruchu samochodu kinematycznego.

Kolejnym zagadnieniem rozpatrywanym w artykule były układy dwułańcuchowe postaci

$$\begin{aligned}
 \dot{q}_0 &= u_1 & \dot{q}'_0 &= u_2 \\
 \dot{q}_1 &= q'_0 u_1 & (\dot{q}'_1 &= q_0 u_2) \\
 \dot{q}_2 &= q_1 u_1 & \dot{q}'_2 &= q'_1 u_2 \\
 \dot{q}_3 &= q_2 u_1 & \dot{q}'_3 &= q'_2 u_2 \\
 &\vdots & &\vdots \\
 \dot{q}_n &= q_{n-1} u_1 & \dot{q}'_{n'} &= q'_{n'-1} u_2.
 \end{aligned} \tag{4.21}$$

Równanie na  $\dot{q}'_1$  jest ujęte w nawiasy dla zaznaczenia, że zmienna  $q'_1$  nie jest niezależną. Aby sterować układem (4.21) zaproponowano algorytm:

1. dobierz sterowania, by  $q_0$  i  $q'_0$  osiągnęły zadane wartości,
2. dla każdego  $q_k$ ,  $k \geq 1$  ustaw pożądaną wartość współrzędnej  $q_k$  używając sterowań  $u_1 = a \sin t$ ,  $u_2 = b \cos kt$ , gdzie parametry  $a$  i  $b$  spełniają zależność

$$q_k(2\pi) - q_k(0) = \frac{(a/2)^k b}{k!} \cdot 2\pi, \tag{4.22}$$

3. dla każdego  $q'_k$ ,  $k \geq 2$  ustaw pożądaną wartość  $q'_k$  używając sterowań  $u_1 = b \cos kt$ ,  $u_2 = a \sin t$ , gdzie  $a$  i  $b$  spełniają

$$q'_k(2\pi) - q'_k(0) = \frac{(a/2)^k b}{k!} \cdot 2\pi. \tag{4.23}$$

Warto zwrócić uwagę, że metoda sterowań sinusoidalnych (harmonicznych) powinna nosić nazwę sterowań okresowych, gdyż jej istotą jest stosowanie wymuszeń okresowych w celu przemieszczania pewnych współrzędnych zachowując brak ruchu (po okresie sterowania) innych współrzędnych. Docelowe wartości współrzędnych osiąga się przez manipulowanie parametrami przebiegów okresowych (amplitudy, częstotliwości, przesunięcia fazowe). Oryginalnie metoda sterowań sinusoidalnych zakłada brak przeszkód (obszarów zabronionych) w przestrzeni stanu. Wydaje się, że metoda jest dość elastyczna i może być adaptowana do przestrzeni z przeszkodami, gdyż istnieje spory margines dowolności w doborze parametrów sterowań, mogący służyć unikaniu przeszkód.

### 4.3. Metoda kontynuacji

Meta-metodą rozwiązywania zadania planowania ruchu jest wykorzystanie metody kontynuacji, szczegółowo opisanej w artykule [69]. Metody kontynuacji polegają na przejściu od rozwiązywania trudnego zadania  $\mathbf{F}(\mathbf{x}) = \mathbf{0}$  do sparametryzowanej rodziny zadań  $\mathbf{F}_t(\mathbf{x}) = \mathbf{0}$ ,  $t \in [0, 1]$ , gdzie  $\mathbf{F}_1(\mathbf{x}) = \mathbf{0} = \mathbf{F}(\mathbf{x})$  jest pierwotnym zadaniem, a rozwiązanie zadania  $\mathbf{F}_0(\mathbf{x}) = \mathbf{0}$  jest znane lub łatwe do wyznaczenia. W oparciu o powyższą parametryzację następuje (gładka) deformacja zadania  $\mathbf{F}_0(\mathbf{x}) = \mathbf{0}$  w zadanie oryginalne  $\mathbf{F}_1(\mathbf{x}) = \mathbf{0}$ , przez rozwiązanie zadań dla wzrastającej wartości  $t$ .

Matematyczna formalizacja metody kontynuacji jest następująca: dla przestrzeni topologicznych  $X$  i  $Y$  oraz problemu  $P$  wyznaczonego przez parę  $(F, U)$ , gdzie  $U$  jest zbiorem zawartym w  $Y$ , a  $F$  odwzorowaniem  $F : X \rightarrow Y$ , rozwiązaniem  $P$  jest punkt  $x \in X$  taki, że  $F(x) \in U$ . Proces kontynuacji jest ciągłym mapowaniem  $H : X \times [0, 1] \rightarrow Y$  takim, że:

1.  $H(x, 1) = F(x)$ ;
2. istnieje przynajmniej jedno znane  $x_0 \in X$  będące rozwiązaniem  $H(\cdot, 0)$ , tj.  $H(x_0, 0) \in U$ ;
3. istnieje ciągła krzywa  $\sigma$  w  $X \times [0, 1]$  taka, że  $(x(t), t)$  jest rozwiązaniem  $H(\cdot, \cdot) \in U \forall t \in [0, 1]$  i  $x(0) = 0$ ;
4. przestrzeń  $X \times [0, 1]$  ma różniczkowalną strukturę, czyli krzywa  $(x(t), t)$  jest różniczkowalna.



W cytowanym artykule przytoczono przykłady dwóch metod śledzenia krzywej  $\sigma$ : dyskretnej oraz ciągłej. W metodzie dyskretnej następuje podział przedziału  $[0, 1]$  na skończoną liczbę problemów:

$$H(x, t_n) = 0, \quad 0 = t_0 < t_1 < \dots < t_N = 1. \quad (4.24)$$

Rozpoczynając od znanego rozwiązania  $x(0)$  dla  $H(x, 0)$ ,  $x(t_{n+1})$  jest obliczane lokalnym algorytmem iteracyjnym zainicjowanym w punkcie  $x(t_n)$ . Podstawowym problemem teoretycznym jest wyznaczenie warunków na  $H$  zapewniających istnienie podziału  $0 = t_0 < t_1 < \dots < t_N = 1$  oraz istnienie procesu iteracyjnego wyznaczania  $x_k^{j+1} = \Lambda(t_k, x_k^j)$ , takiego że  $x_k$  jest w obszarze atrakcji  $x_{k+1}$ . Metoda ciągła polega na różniczkowaniu  $H(x(t), t) = 0$ , dając równanie Davidenki [91]

$$\frac{\partial}{\partial x} H(x(t), t) \frac{\partial x(t)}{\partial t} + \frac{\partial}{\partial t} H(x(t), t) = 0. \quad (4.25)$$

Warunkiem brzegowym jest  $x(0) = x_0$ . Po całkowaniu na przedziale  $[0, 1]$  otrzymuje się rozwiązanie  $x(1)$ . Głównym problemem tej metody jest konieczność unikania osobliwości  $\frac{\partial}{\partial x} H(x(t), t)$ . Podano kilka sposobów zaradzenia osobliwościom: zmiana parametru w  $t$ , najczęściej przejście do parametryzacji  $\sigma$  względem długości krzywej, dopuszczenie, aby  $t$  było z  $\mathbb{R}^k$  a nie tylko skalarem, albo odpowiedni dobór „trywialnej” funkcji  $H(x, 0)$ .

Praktyczne zastosowanie metody kontynuacji w dziedzinie robotyki przedstawił Sussmann w artykule [80]. Zamieszczono w tym artykule algorytm bazujący na utworzeniu trajektorii wychodzącej z punktu początkowego  $q_0$  i zmierzającej do pewnego punktu pośredniego  $q'_f$ , a następnie ciągłym przekształceniu tej trajektorii, tak aby osiągnąć wymagany punkt końcowy  $q_f$ .

#### 4.4. Metoda typu pontriaginowskiego

W artykule [67] rozważono sterowanie dwukołowym robotem mobilnym o dwóch kołach napędowych:

$$\begin{aligned} \dot{\phi} &= (\omega_R - \omega_L)/D \\ \dot{x} &= (\omega_R + \omega_L) \cos \phi/2 \\ \dot{y} &= (\omega_R + \omega_L) \sin \phi/2 \\ \dot{\omega}_R &= u_R \\ \dot{\omega}_L &= u_L, \end{aligned} \quad (4.26)$$

gdzie  $n = 5$ ,  $m = 2$ ,  $D$  jest odległością koła od środka platformy. Sterowaniami były przyspieszenia  $u = (u_L, u_R)$  obu kół z ograniczeniem wartości maksymalnej ( $|u_R| \leq a_{max}$ ,  $|u_L| \leq a_{max}$ ). Zadaniem optymalizacyjnym była minimalizacja czasu trwania ruchu bez ograniczeń na przestrzeń stanu. Klasycznie, wprowadzono funkcję Hamiltona  $H$  zależną od równań ruchu i zmiennych dualnych

$$H(\psi, q, u) = -1 + \sum_{j=1}^n \psi_j f_j(q, u), \quad (4.27)$$

gdzie  $f_j$ ,  $j = 1, \dots, 5$  są kolejnymi funkcjami opisującymi prawe strony równań (4.26). Uzupełniono równania ruchu (4.26) o te wynikające ze współrzędnych dualnych (równania kanoniczne Hamiltona)

$$\dot{\psi}_i = - \sum_{j=1}^n \frac{\partial f_j(q, u)}{\partial q_i} \psi_j, \quad i = 1, 2, \dots, n, \quad (4.28)$$

z warunkiem początkowym

$$\psi_i(t_0) = \lambda_i. \quad (4.29)$$

Zasada maksimum Pontryagina orzeka, że gdy  $\mathbf{u}$  jest dopuszczalnym sterowaniem przemieszczającym stan układu z położenia początkowego  $\mathbf{q}(t_0) = \mathbf{q}_0$  do stanu zadanego  $\mathbf{q}(t_1) = \mathbf{q}_1$  na horyzoncie czasu  $[t_0, t_1]$ , wtedy  $\mathbf{u}(t)$  i wynikowa trajektoria  $\mathbf{q}(t)$  są optymalne czasowo, jeśli istnieje nie tożsamościowo zerowa funkcja  $(\psi_1(t), \dots, \psi_n(t))$  spełniająca równanie (4.28) taka, że  $\mathbf{u}(t)$  maksymalizuje hamiltonian  $H$  dla każdego  $t$  z przedziału  $t_0 \leq t \leq t_1$ . Stan zmiennych dualnych jest uzależniony od rodzaju zadania. Jeśli konfiguracja końcowa platformy jest w pełni określona (położenie i orientacja) wtedy wartości zmiennych dualnych są dowolne. Jeśli określone jest tylko położenie bądź orientacja, wtedy wartości końcowe zmiennych dualnych są określone. Na podstawie zasady Maksimum Pontryagina dowiedziono, że sterowaniami czaso-optymalnymi dla rozważanego przypadku są sterowania typu bang-bang (maksymalna amplituda na przynajmniej jednym wymuszeniu w praktycznie każdej chwili czasu). Sterowania zostały sparametryzowane czasami przełączenia (zmiany znaku przyspieszenia dla każdego z kół). Autorzy przedstawili metodę wyznaczania wartości zmiennych dualnych. Pokazano, że do osiągnięcia dowolnego położenia wystarczą trzy zmiany sterowań, dodając czwartą zmianę uzyskuje się dowolną konfigurację.

Wadą metod wykorzystujących zasadę Maksimum Pontryagina jest wrażliwość na istnienie przeszkód (obszarów zabronionych) w przestrzeni stanu, oraz duża złożoność obliczeniowa, natomiast zaletą gwarantowana optymalność sterowań.

#### 4.5. Metoda typu Lapunowa

W pracy [60] przedstawiono analizę robotów kosmicznych. Kinematyka i dynamika układów kosmicznych wynika z mikrograwitacją i praw zachowania (momentu) pędu:

$$\begin{aligned} \sum_k m_k \dot{\mathbf{r}}_k &= \mathbf{0} \\ \sum_k (\mathbf{I}_k \boldsymbol{\omega}_k + m \mathbf{r}_k \times \dot{\mathbf{r}}_k) &= \mathbf{0}, \end{aligned} \quad (4.30)$$

gdzie  $m_k$  jest masą  $k$ -tego ciała,  $\mathbf{r}_k$  pozycją środka masy  $k$ -tego ciała względem początku układu,  $\boldsymbol{\omega}_k$  prędkością obrotową, a  $\mathbf{I}_k$  macierzą inercji. Łatwo pokazać, że tak opisane roboty kosmiczne są nieholonomiczne i na poziomie kinematycznym zadane jako układy bezdryfowe

$$\dot{\mathbf{q}} = \mathbf{G}(\mathbf{q})\mathbf{u}. \quad (4.31)$$

Autorzy zaprezentowali metodę wyznaczania sterowań  $\mathbf{u}$  przemieszczających stan układu z zadanego punktu  $\mathbf{q}_0$  do docelowego  $\mathbf{q}_f$  używając funkcji Lapunowa zadanej jako

$$v = \frac{1}{2} \Delta \mathbf{q}^T \mathbf{A} \Delta \mathbf{q}, \quad \text{gdzie} \quad \Delta \mathbf{q} = \mathbf{q}_f - \mathbf{q},$$

gdzie  $\mathbf{q}$  jest stanem bieżącym, a  $\mathbf{A}$  to stała, symetryczna, dodatnio-określona macierz. Pochodna  $\dot{v}$  wynosi

$$\dot{v} = -\Delta \mathbf{q}^T \mathbf{A} \dot{\mathbf{q}} = -\Delta \mathbf{q}^T \mathbf{A} \mathbf{G} \mathbf{u}. \quad (4.32)$$

Dla tego przypadku zaproponowano następującą formułę na sterowania:

$$\mathbf{u} = (\mathbf{A} \mathbf{G})^T \Delta \mathbf{q}. \quad (4.33)$$

Niezadowolające działanie algorytmu pod kątem unikania osobliwości spowodowało propozycję modyfikację algorytmu bazowego i zastosowanie algorytmu dwukierunkowego. Idea tej metody polega na planowaniu ruchu dwóch kopii układu (4.31) (dalej wyróżnionych indeksem dolnym)

$$\begin{aligned}\dot{\mathbf{q}}_1 &= \mathbf{G}_1(\mathbf{u}_1)\mathbf{u}_1, & \mathbf{q}_1(0) &= \mathbf{q}_0 \\ \dot{\mathbf{q}}_2 &= \mathbf{G}_2(\mathbf{u}_2)\mathbf{u}_2, & \mathbf{q}_2(0) &= \mathbf{q}_f,\end{aligned}$$

z których pierwszy jest zainicjowany w stanie początkowym a drugi w docelowym. Celem sterowania jest osiągnięcie, w pewnej chwili czasu, identycznego stanu przez obie kopie układu. Dlatego funkcję Lapunowa konstruuje się postaci

$$v = \frac{1}{2}\Delta\mathbf{q}^T\mathbf{A}\Delta\mathbf{q}, \quad \text{gdzie} \quad \Delta\mathbf{q} = \mathbf{q}_1 - \mathbf{q}_2.$$

Pochodna funkcji Lapunowa  $\dot{v}$  wynosi

$$\dot{v} = \Delta\mathbf{q}^T\mathbf{A}\hat{\mathbf{K}}\begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix}, \quad (4.34)$$

gdzie

$$\hat{\mathbf{K}} = (\mathbf{K}_1 \ \mathbf{K}_2). \quad (4.35)$$

Następujący dobór sterowań

$$\begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} = -(\mathbf{A}\hat{\mathbf{K}})^\# \Delta\mathbf{q} \quad (4.36)$$

gwarantuje spełnienie warunków Lapunowa, a zatem zbieżność algorytmu. Jeżeli funkcja Lapunowa zbiega do zera dla  $t = t_f$ , trajektoria systemu wynikowego łącząca stan początkowy  $\mathbf{q}_0$  oraz końcowy  $\mathbf{q}_f$  jest uzyskiwana przez odpowiednią konkatencję fragmentów trajektorii obu kopii układu wyjściowego:

$$\mathbf{q}(t) = \begin{cases} \mathbf{q}_1(t) & \text{dla } 0 \leq t \leq t_f \\ \mathbf{q}_2(2t_f - t) & \text{dla } t_f < t \leq 2t_f, \end{cases}$$

a sterowania mają postać:

$$\mathbf{u}(t) = \begin{cases} \mathbf{u}_1(t) & \text{dla } 0 \leq t \leq t_f \\ -\mathbf{u}_2(2t_f - t) & \text{dla } t_f < t \leq 2t_f. \end{cases}$$

Praca została podsumowana kilkoma przykładami symulacyjnymi robota o 6 stopniach swobody umieszczonego na pojeździe kosmicznym.

#### 4.6. Metody newtonowskie

W artykule [48] przedstawiono jedną z wersji algorytmu Newtona noszącej nazwę techniki iteracji do zaprojektowania kontrolera pracującego w trybie czasu rzeczywistego (kolejną wersję wykorzystania algorytmu Newtona w planowaniu ruchu układów nieholonomicznych można znaleźć w pracy [32]). Pokazano, że przy znanym modelu podejście to gwarantuje asymptotyczną stabilność kontrolera (pracującego w pętli sprzężenia zwrotnego), oraz charakteryzuje się odpornością na niepewność parametrów modelu. Zaproponowany algorytm nie jest planerem off-line, a strategią doboru sterowań podobną do sterowania predykcyjnego. Zadania zaplanowania ruchu dla systemu nieholonomicznego o ograniczeniach w postaci Pfaffa (3.6), wyrażonych w postaci układu bezdryfowego

$$\dot{\mathbf{q}} = \mathbf{G}(\mathbf{q})\mathbf{u} \quad (4.37)$$

jest klasyczne i polega na połączeniu trajektorią zadanej konfiguracji początkowej  $\mathbf{q}_0$  z końcową  $\mathbf{q}_f$  na znormalizowanym do jedności horyzoncie czasowym. Osiągnięty w rzeczywistości stan końcowy  $\mathbf{q}(1)$  można wyrazić jako nieliniowe odwzorowanie sterowania  $\mathbf{u}(\cdot)$  zainicjowane w stanie  $\mathbf{q}_0$ :

$$\mathbf{q}(1) = \mathbf{F}(\mathbf{q}_0, \mathbf{u}(\cdot)).$$

Zapisując błąd osiągnięcia stanu docelowego jako

$$\mathbf{e}(\mathbf{q}_0, \mathbf{u}(\cdot)) = \mathbf{F}(\mathbf{q}_0, \mathbf{u}(\cdot)) - \mathbf{q}_f \quad (4.38)$$

zadanie planowania trajektorii ulega przekształceniu w nieliniowy problem optymalizacyjny. Kolejne iteracje poprawiają początkowe wartości sterowań  $\mathbf{u}(\cdot)(\tau)$ , gdzie  $\tau$  jest zmienną iteracyjną (najczęściej dyskretną) wykorzystując, przykładowo, algorytm gradientowy

$$\frac{d\mathbf{u}}{d\tau} = -\alpha [\nabla_{\mathbf{u}} \mathbf{F}(\mathbf{q}_0, \mathbf{u}(\cdot)(\tau))]^\# \mathbf{e}(\tau),$$

gdzie  $\nabla_{\mathbf{u}} \mathbf{F}$  jest pochodną funkcji  $\mathbf{F}$  względem  $\mathbf{u}$ . Przy założeniu, że gradient jest pełnego rzędu, otrzymuje się równanie błędu w postaci zbieżnego do zera (przy  $\alpha > 0$ ) stanu układu dynamicznego

$$\frac{d\mathbf{e}}{d\tau} = -\alpha \mathbf{e}.$$

Wyznaczanie  $\nabla_{\mathbf{u}} \mathbf{F}$  odbywa się przez linearyzację układu (4.37) wzdłuż ścieżki odpowiadającej sterowaniu  $\mathbf{u}(\cdot)$ :

$$\partial \dot{\mathbf{q}} = \mathbf{A}(t) \partial \mathbf{q} + \mathbf{B}(t) \partial \mathbf{u}, \quad \partial \mathbf{q}(0) = \mathbf{0}, \quad (4.39)$$

gdzie  $\mathbf{A}(t) = \left[ \frac{\partial \mathbf{G}(\mathbf{q})}{\partial q_1} \mathbf{u}(t), \dots, \frac{\partial \mathbf{G}(\mathbf{q})}{\partial q_n} \mathbf{u}(t) \right]$ , natomiast  $\mathbf{B}(t) = \mathbf{G}(\mathbf{q}(t))$ . Rozwiązaniem równania (4.39) w chwili  $t = 1$  jest

$$\partial \mathbf{q}(1) = \int_0^1 \Phi(1, s) \mathbf{B}(s) \partial \mathbf{u}(s) ds, \quad (4.40)$$

gdzie  $\Phi$  jest macierzą fundamentalną systemu zlinearyzowanego, spełniającą równanie

$$(\nabla_{\mathbf{u}} \mathbf{F}) \mathbf{v} = \int_0^1 \Phi(1, s) \mathbf{B}(s) \mathbf{v}(s) ds. \quad (4.41)$$

Kolejnym zagadnieniem rozważanym w cytowanej pracy jest przekształcenie schematu iteracyjnego (generującego trajektorie w otwartej pętli sprzężenia) na schemat z zamkniętą pętlą sprzężenia gwarantujący śledzenie ścieżki. Powszechną metodą jest linearyzacja układu wzdłuż wybranej trajektorii i zaprojektowanie liniowego kontrolera utrzymującego układ na ścieżce. Z powodu wykorzystania linearyzacji, stabilność takiego rozwiązania jest jedynie lokalna. Autorzy proponują metodę modyfikacji planera off-line bezpośrednio do formy ze sprzężeniem zwrotnym: jednoczesne prowadzenie obliczeń i sterowanie systemem wzdłuż ścieżki w bieżącej iteracji. W każdym kroku czasowym bieżący stan jest używany jako warunek początkowy algorytmu popraw, wyznaczone są sterowania, system jest przemieszczany do nowego stanu i cykl jest powtarzany. Autorzy proponują uzależnienie zmiennej iteracyjnej od czasu  $\tau = \tau(t)$ . Pokazano, że jeżeli  $\tau(t)$  jest funkcją nieujemną, to błąd osiągnięcia konfiguracji zadanej wynosi

$$\mathbf{e}' = e^{-\alpha \tau(1)} \mathbf{e}(0)$$

i może być dowolnie mały przy odpowiednim doborze  $\alpha$  bądź  $\tau(t)$ . Artykuł kończy się symulacyjnym badaniem planera wykorzystując model jednokołowca i samochodu kinematycznego.

### 4.7. Metody przestrzeni endogenicznej

W artykule [86] przedstawiono metodę planowania ruchu manipulatora mobilnego techniką przestrzeni endogenicznej skojarzonej z algorytmem newtonowskim. W tym podejściu kinematyka

$$K_{q_0,T}(\mathbf{u}(\cdot), \mathbf{x}) = \mathbf{y}(T) = \mathbf{k}(\varphi_{q_0,T}(\mathbf{u}(\cdot)), \mathbf{x}) \quad (4.42)$$

określa położenie i orientację końcówki efektora po zastosowaniu sterowania  $\mathbf{u}(\cdot)$  przez czas  $T$ . Zróżniczkowanie kinematyki generuje Jakobian

$$\begin{aligned} J_{q_0,T}(\mathbf{u}(\cdot), \mathbf{x})(\mathbf{v}(\cdot), \mathbf{w}) &= \mathbf{D}K_{q_0,T}(\mathbf{u}(\cdot), \mathbf{x})(\mathbf{v}(\cdot), \mathbf{w}) = \\ &= \mathbf{C}(T, \mathbf{x}) \int_0^T \Phi(T, s) \mathbf{B}(s) \mathbf{v}(s) ds + \mathbf{D}(T, \mathbf{x}) \mathbf{w} \end{aligned} \quad (4.43)$$

Jakobian można przedstawić jako mapę zasięgu w  $T$  liniowej aproksymacji

$$\dot{\xi} = \mathbf{A}(t)\xi + \mathbf{B}(t)\mathbf{v}, \quad \eta = \mathbf{C}(t, \mathbf{x}) + \mathbf{D}(t, \mathbf{x})\mathbf{w} \quad (4.44)$$

systemu (2.4) wzdłuż  $(\mathbf{u}(t), \mathbf{x}, \mathbf{q}(t))$ . Macierze występujące w (4.44) mają następującą postać:

$$\begin{aligned} \mathbf{A}(t) &= \frac{\partial}{\partial \mathbf{q}}(\mathbf{G}(\mathbf{q}(t))\mathbf{u}(t)) & \mathbf{B}(t) &= \mathbf{G}(\mathbf{q}(t)) \\ \mathbf{C}(t, \mathbf{x}) &= \frac{\partial}{\partial \mathbf{q}}\mathbf{k}(\mathbf{q}(t), \mathbf{x}) & \mathbf{D}(t, \mathbf{x}) &= \frac{\partial}{\partial \mathbf{x}}\mathbf{k}(\mathbf{q}(t), \mathbf{x}), \end{aligned} \quad (4.45)$$

a macierz  $\Phi(t, s)$  spełnia równanie ewolucji układu  $\frac{\partial}{\partial t}\Phi(t, s) = \mathbf{A}(t)\Phi(t, s)$  z warunkiem początkowym  $\Phi(s, s) = \mathbf{I}_n$ . Autorzy posługują się pojęciem konfiguracji endogenicznej  $\mathbf{z} = (\mathbf{u}(\cdot), \mathbf{x})$ . Konfiguracja endogeniczna jest regularna, jeśli jakobian (4.43) jest surjekcją, co jest równoważne pełnemu rzędowi macierzy zrzętności

$$D_{q_0,T}(\mathbf{z}) = \mathbf{C}(T, \mathbf{x}) \int_{s=0}^T \Phi(T, s) \mathbf{B}(s) \mathbf{B}^T(s) \Phi^T(T, s) ds \mathbf{C}^T(T, \mathbf{x}) + \mathbf{D}(T, \mathbf{x}) \mathbf{D}^T(T, \mathbf{x}). \quad (4.46)$$

Zadanie kinematyki odwrotnej systemu (2.4) polega na znalezieniu takiej konfiguracji endogenicznej  $(\mathbf{u}_d(\cdot), \mathbf{x}_d)$ , że dla zadanego położenia  $\mathbf{y}_d$  końcówki efektora zachodzi  $K_{q_0,T}(\mathbf{u}_d(\cdot), \mathbf{x}_d) = \mathbf{y}_d$ . Najczęściej stosowaną metodą rozwiązania zadania planowania ruchu jest wykorzystanie jakobianu pseudoodwrotnego o postaci

$$(J_{q_0,T}^\#(\mathbf{u}(\cdot), \mathbf{x})\eta)(t) = \begin{bmatrix} \mathbf{B}^T(t)\Phi^T(T, t)\mathbf{C}^T(T, \mathbf{x}) \\ \mathbf{D}^T(T, \mathbf{x}) \end{bmatrix} D_{q_0,T}^{-1}(\mathbf{u}(\cdot), \mathbf{x})\eta \quad (4.47)$$

i przyjmuje formę:

$$\frac{d}{d\theta} \begin{pmatrix} \mathbf{u}_\theta(t) \\ \mathbf{x}(\theta) \end{pmatrix} = -\gamma J_{q_0,T}^\#(\mathbf{u}_\theta, \mathbf{x}(\theta))\mathbf{e}(\theta)(t), \quad (4.48)$$

gdzie  $\gamma > 0$  określa długość kroku,  $\mathbf{e}(\theta) = K_{q_0,T}(\mathbf{u}_\theta(\cdot), \mathbf{x}(\theta)) - \mathbf{y}_d$  jest błędem w przestrzeni zadaniowej. Niech  $\mathbf{z} = (\mathbf{u}(\cdot), \mathbf{x})$  oznacza konfigurację endogeniczną. Dla konfiguracji regularnych jakobian (4.43) posiada odwrotność  $\mathbf{D}K^\#(\mathbf{z})$ . Algorytm kinematyki odwrotnej można zapisać jako

$$\frac{d\mathbf{z}(\theta)}{d\theta} = -\gamma \mathbf{D}K^\#(\mathbf{z}(\theta))\mathbf{e}(\theta), \quad \mathbf{z}(0) = \mathbf{z}_0. \quad (4.49)$$

Po dyskretyzacji, algorytm (4.49) zapisuje się zależnością

$$\mathbf{z}(k+1) = \mathbf{z}(k) - \gamma \mathbf{D}K^\#(\mathbf{z}(k))\mathbf{e}(k), \quad k = 0, 1, 2, \dots \quad (4.50)$$

Zaproponowano dwie metody doboru długości kroku  $\gamma$ . Pierwsza, *Armijo*, opiera się na zdefiniowaniu dla ustalonej wartości  $m$  zbioru  $\Gamma_m = \{1, \frac{1}{2}, \dots, \frac{1}{2^m}\}$  kandydatów i doboru w każdym kroku algorytmu współczynnika  $\gamma = \arg \min_{\gamma \in \Gamma_m} F(\alpha)$ .

Druga oparta na warunku, któremu podlega **DK**, polega na zastosowaniu algorytmu predykcyjno-korekcyjnego zainspirowanego pracą [8], dając następujący algorytm korekty kroku. Na początku algorytm jest inicjowany w  $\mathbf{z}_0$  z wartością  $\gamma_1^0 = 1$ . W  $k$ -tym kroku algorytm zwraca  $\mathbf{z}^0(k)$ , dokonuje predykcji  $\gamma_k^0$  i wylicza  $\mathbf{e}^0(k) = \mathbf{K}(\mathbf{z}^0(k)) - \mathbf{y}_d$ . Następnie wartość  $\gamma_1^0$  poddawana jest serii korekt. Wyznaczana jest wartość  $\mathbf{z}^i(k) = \mathbf{z}^0(k) - \gamma_k^i \mathbf{DK}^\#(\mathbf{z}^0(k)) \mathbf{e}^0(k)$ , następnie wyliczany błąd  $\mathbf{e}^i(k) = \mathbf{K}(\mathbf{z}^i(k)) - \mathbf{y}_d$  i  $(i+1)$  korekta:

$$\gamma_k^{i+1} = \min \left\{ \frac{1}{2} \gamma_k^i, \frac{1}{h_k^{i+1}} \right\}, \quad (4.51)$$

gdzie

$$h_k^{i+1} = \frac{2 \|\mathbf{DK}^\#(\mathbf{z}^0(k)) \mathbf{e}^i(k) - (1 - \gamma_k^i) \mathbf{DK}^\#(\mathbf{z}^0(k)) \mathbf{e}^0(k)\|}{(\gamma_k^i)^2 \|\mathbf{DK}^\#(\mathbf{z}^0(k)) \mathbf{e}^0(k)\|}. \quad (4.52)$$

Wyznaczanie korekt trwa ustaloną liczbę iteracji dając  $i^*$  takie że  $\gamma_k^{i^*} = \arg \min_i \|\mathbf{e}^i(k)\|^2$ . Następnie dysponując wartością  $\gamma_k^{i^*}$  przyjmowany jest  $\mathbf{z}_0(k+1) = \mathbf{z}^{i^*}(k)$  oraz wyliczane predykcje dla kroku  $k+1$ :

$$\gamma_{k+1}^0 = \min \left\{ 1, \frac{1}{h_{k+1}^0} \right\}, \quad (4.53)$$

gdzie

$$h_{k+1}^0 = \frac{\|\mathbf{DK}^\#(\mathbf{z}^0(k+1)) (\mathbf{DK}(\mathbf{z}^0(k+1)) - \mathbf{DK}(\mathbf{z}^0(k))) \mathbf{DK}^\#(\mathbf{z}^0(k+1)) \mathbf{e}^0(k+1)\|}{\gamma_k^{i^*} \|\mathbf{DK}^\#(\mathbf{z}^0(k)) \mathbf{e}^0(k)\|}. \quad (4.54)$$

Działanie algorytmu zostało sprawdzone symulacyjnie na manipulatorze mobilnym złożonym z samochodu kinematycznego z manipulatorem RTR na pokładzie.

W artykule [85] szczegółowo opisano metodę jakobianu rozszerzonego dla platformy robotycznej. Algorytm jakobianu rozszerzonego opiera się na znalezieniu takiego rozszerzenia kinematyki, aby dokonywała mapowania endogeniczej przestrzeni konfiguracji  $X$  w siebie samą. W tym celu dokonuje się dekompozycji

$$X \cong \mathbb{R}^r \oplus X/\mathbb{R}^r \quad (4.55)$$

przestrzeni endogeniczej, gdzie  $\mathbb{R}^r$  odpowiada przestrzeni roboczej. Wprowadza się kinematykę poszerzającą

$$\mathbf{H}_{q_0, T} : X \mapsto X/\mathbb{R}^r, \quad (4.56)$$

która w połączeniu z pierwotną kinematyką daje kinematykę rozszerzoną:

$$\mathbf{L}_{q_0, T} = (\mathbf{K}_{q_0, T}, \mathbf{H}_{q_0, T}) : X \mapsto X. \quad (4.57)$$

Pochodna

$$\mathbf{DL}_{q_0, T}(\mathbf{u}(\cdot)) = \bar{\mathbf{J}}_{q_0, T}(\mathbf{u}(\cdot)) = (\mathbf{J}_{q_0, T}(\mathbf{u}(\cdot)), \mathbf{DH}_{q_0, T}(\mathbf{u}(\cdot))) \quad (4.58)$$

jest jakobianem rozszerzonym robota mobilnego. Dla regularnych konfiguracji (4.58) wprowadza się rozszerzoną odwrotność

$$\mathbf{J}_{q_0, T}^{E\#}(\mathbf{u}(\cdot)) : \mathbb{R}^r \mapsto X$$

jakobianu analitycznego:

$$\mathbf{J}_{q_0, T}^{E\#}(\mathbf{u}(\cdot)) \boldsymbol{\eta} = \bar{\mathbf{J}}_{q_0, T}^{-1}(\mathbf{u}(\cdot))(\boldsymbol{\eta}, \mathbf{0}(\cdot)), \quad (4.59)$$

gdzie  $\eta \in \mathbb{R}^r$ ,  $\mathbf{0}(\cdot) \in X/\mathbb{R}^r$ . Rozwiązanie dla algorytmu jacobianu rozszerzonego uzyskuje się z wykorzystaniem metod kontynuacji. Na przykładzie monocykla przedstawiona praktycznie została idea wyboru rozszerzenia. Zauważono, przy tym że wybór rozszerzenia nie jest jednoznaczny, a jest zależny od wymagań odnośnie zadania dodatkowego (np. unikanie przeszkód).

Przystosowanie algorytmu jacobianu rozszerzonego do przestrzeni kolizyjnej opisuje artykuł [36]. Przeszkody zostały przedstawione jako zbiór zdarzeń  $\{(t_1, \mathbf{y}_1), (t_2, \mathbf{y}_2), \dots, (t_k, \mathbf{y}_k)\}$  określających zabronione pozycje i orientacje efektora w chwilach czasowych  $0 \leq t_1 \leq t_2 \leq \dots \leq t_k \leq T$ ,  $\mathbf{y}_1, \dots, \mathbf{y}_k \in \mathbb{R}^r$ . Dla konfiguracji endogenicznej  $(\mathbf{u}_\theta(\cdot), \mathbf{x}(\theta))$  wyznaczana jest odległość w przestrzeni zadaniowej

$$e_i(\mathbf{u}_\theta(\cdot), \mathbf{x}(\theta)) = \|\mathbf{K}_{q_0, t_i}(\mathbf{u}_\theta(\cdot), \mathbf{x}(\theta)) - \mathbf{y}_i\| \quad (4.60)$$

i wyznaczana jest najmniejsza wartość

$$e_*^2(\mathbf{u}_\theta, \mathbf{x}(\theta)) = \min_{1 \leq i \leq k} e_i^2(\mathbf{u}_\theta, \mathbf{x}(\theta)) \quad (4.61)$$

związana ze zdarzeniem  $(t^*, \mathbf{y}^*)$ . Następnie określana jest specjalna konfiguracja  $(\omega(\cdot), \zeta)$  albo jako kierunek wzrostu odległości (4.61) (gradient)

$$(\omega(\cdot), \zeta) = \mathbf{J}_{q_0, t^*}^*(\mathbf{u}_\theta(\cdot), \mathbf{x}(\theta)) \frac{\mathbf{K}_{q_0, t^*}(\mathbf{u}_\theta(\cdot), \mathbf{x}(\theta)) - \mathbf{y}^*}{e_*^p(\mathbf{u}_\theta(\cdot), \mathbf{x}(\theta))}, \quad (4.62)$$

gdzie  $p \geq 2$ , albo następuje przetransformowanie  $\mathbf{y}^*$  w konfigurację endogeniczną  $(\mathbf{u}^*(\cdot), \mathbf{x}^*)$  używając algorytmu jacobianu pseudoodwrotnego i następnie wyznaczane jest

$$(\omega(\cdot), \zeta) = (\mathbf{u}_\theta(\cdot), \mathbf{x}(\theta)) - (\mathbf{u}^*(\cdot), \mathbf{x}^*). \quad (4.63)$$

Odepchnięcie od wybranej konfiguracji endogenicznej  $(\omega(\cdot), \zeta)$  następuje z wykorzystaniem algorytmu Newtona z optymalizacją w przestrzeni zerowej.

Kontynuując badania nad jacobianami rozszerzonymi, w artykule [84] zwrócono uwagę na powtarzalność algorytmów wykorzystywanych przy układach nieholonomicznych. Powtarzalność oznacza, że algorytm kinematyki odwrotnej przekształca zamkniętą ścieżkę w przestrzeni zadaniowej w zamkniętą ścieżkę w endogenicznej przestrzeni konfiguracyjnej. Autorzy zauważają, że powtarzalny algorytm generuje sterowania dla platformy i manipulatora zależne jedynie od pożądanej konfiguracji końcowej efektora, nie zależące od położenia początkowego. Na podstawie właściwości jacobianu rozszerzonego (4.59)

$$\begin{aligned} \mathbf{J}_{q_0, T}(\mathbf{u}(\cdot), \mathbf{x}) \mathbf{J}_{q_0, T}^{E\#}(\mathbf{u}(\cdot), \mathbf{x}) \eta &= \eta \\ \mathbf{D}\mathbf{H}_{q_0, T}(\mathbf{u}(\cdot), \mathbf{x}) \mathbf{J}_{q_0, T}^{E\#}(\mathbf{u}(\cdot), \mathbf{x}) \eta &= \mathbf{0}(\cdot) \end{aligned} \quad (4.64)$$

stwierdzono powtarzalność algorytmu jacobianu rozszerzonego.

Algorytm jacobianu rozszerzonego został przeniesiony w artykule [83] na przypadek podwójnie nieholonomicznego manipulatora mobilnego. Manipulator mobilny typu  $(nh, nh)$  składa się z nieholonomicznego manipulatora umieszczonego na nieholonomicznej platformie. Równanie takiego systemu ma postać:

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{G}(\mathbf{q}) \mathbf{u}^1 = \sum_{i=1}^{m_1} \mathbf{g}_i(\mathbf{q}) u_i^1 \\ \dot{\mathbf{p}} &= \mathbf{F}(\mathbf{q}) \mathbf{u}^2 = \sum_{i=1}^{m_2} \mathbf{f}_i(\mathbf{q}) u_i^2 \\ \mathbf{y} &= \mathbf{k}(\mathbf{q}, \mathbf{p}), \end{aligned} \quad (4.65)$$

gdzie  $\mathbf{q} \in \mathbb{R}^{n_1}$ ,  $\mathbf{p} \in \mathbb{R}^{n_2}$  określają odpowiednio stan platformy i manipulatora,  $\mathbf{u}^1(t) \in \mathbb{R}^{m_1}$ ,  $\mathbf{u}^2(t) \in \mathbb{R}^{m_2}$  są sterowaniami platformy i manipulatora określonymi na przedziale  $[0, T]$ ,  $\mathbf{y} \in \mathbb{R}^r$  określa współrzędne położenia i orientacji efektora. Kinematyka i jacobian manipulatora podwójnie nieholonomicznego są zdefiniowane analogicznie do (4.42) i (4.43),

$$\mathbf{K}_{\mathbf{q}_0, \mathbf{p}_0, T}(\mathbf{u}^1(\cdot), \mathbf{u}^2(\cdot)) = \mathbf{k}(\varphi_{\mathbf{q}_0, T}^1(\mathbf{u}^1(\cdot)), \varphi_{\mathbf{p}_0, T}^2(\mathbf{u}^2(\cdot))) \quad (4.66)$$

i

$$\mathbf{J}_{\mathbf{q}_0, \mathbf{p}_0, T}(\mathbf{u}^1(\cdot), \mathbf{u}^2(\cdot))(\mathbf{v}^1(\cdot), \mathbf{v}^2(\cdot)) = \mathbf{C}_1(T) \int_0^T \Phi_1(T, s) \mathbf{B}_1(s) \mathbf{v}^1(s) ds + \mathbf{C}_2(T) \int_0^T \Phi_2(T, s) \mathbf{B}_2(s) \mathbf{v}^2(s) ds \quad (4.67)$$

gdzie

$$\begin{aligned} \mathbf{A}_1 &= \frac{\partial(\mathbf{G}(\mathbf{q}(t))\mathbf{u}^1(t))}{\partial \mathbf{q}} & \mathbf{B}_1 &= \mathbf{G}(\mathbf{q}(t)) & \mathbf{C}_1 &= \frac{\partial \mathbf{k}(\mathbf{q}(t), \mathbf{p}(t))}{\partial \mathbf{q}} \\ \mathbf{A}_2 &= \frac{\partial(\mathbf{F}(\mathbf{p}(t))\mathbf{u}^2(t))}{\partial \mathbf{p}} & \mathbf{B}_2 &= \mathbf{F}(\mathbf{p}(t)) & \mathbf{C}_2 &= \frac{\partial \mathbf{k}(\mathbf{q}(t), \mathbf{p}(t))}{\partial \mathbf{p}} \end{aligned} \quad (4.68)$$

a  $\Phi_1(t, s)$  i  $\Phi_2(t, s)$  oznaczają macierze fundamentalne liniowej aproksymacji

$$\begin{aligned} \dot{\xi} &= \mathbf{A}_1(t)\xi + \mathbf{B}_1\mathbf{v}^1(t) \\ \dot{\zeta} &= \mathbf{A}_2(t)\zeta + \mathbf{B}_2\mathbf{v}^2(t) \\ \eta &= \mathbf{C}_1(T)\xi + \mathbf{C}_2(T)\zeta \end{aligned} \quad (4.69)$$

równania (4.65) wzdłuż  $(\mathbf{u}^1(t), \mathbf{u}^2(t), \mathbf{p}(t), \mathbf{q}(t))$  zainicjowane w  $(\xi_0, \zeta_0) = \mathbf{0}$ . Zaproponowane zostało rozszerzenie jacobianu o postaci:

$$\mathbf{H}_{\mathbf{q}_0, \mathbf{p}_0, T}(\mathbf{u}^1(\cdot), \mathbf{u}^2(\cdot))(t) = \left( \frac{\tilde{u}_1^1(t)}{x_{i_1}}, \dots, \frac{\tilde{u}_{m_1}^1(t)}{x_{i_{m_1}}}, \frac{\tilde{u}_1^2(t)}{x_{i_{m_1+1}}}, \dots, \frac{\tilde{u}_{m_2}^2(t)}{x_{i_{m_1+m_2}}} \right) \quad (4.70)$$

gdzie  $\mathbf{u}^1(t) = \mathbf{P}_{s_1}(t)\mathbf{x}^1 + \tilde{\mathbf{u}}^1(t)$ ,  $\mathbf{u}^2(t) = \mathbf{P}_{s_2}(t)\mathbf{x}^2 + \tilde{\mathbf{u}}^2(t)$ , w którym  $\mathbf{P}_s$  jest  $s+1$  elementowym wektorem funkcji bazowych, a  $\tilde{\mathbf{u}}$  nieskończenie wymiarową resztą. Przedstawioną metodę ilustruje przykład obliczeniowy wraz z symulacjami trójwahadła nieholonomicznego umieszczonego na samochodzie kinematycznym.

## 4.8. Manipulowalność a planowanie ruchu

Artykuł [1] opisuje propozycję kryteriów manipulowalności rozszerzonych na przypadek manipulatorów mobilnych. Autorzy opisują metodę modelowania manipulatorów i platform mobilnych. Przyjęta strategia modelowania uwzględnia cztery rodzaje kół: *stałe*, których oś posiada ustalony i niezmienny kierunek; *napędowe* o osi obrotu przechodzącej przez środek koła; *koła wózkowe*, gdzie oś obrotu nie pokrywa się ze środkiem koła, oraz *szwedzkie* zbliżone do *stałych*, z dodatkowym parametrem opisującym kierunek składnika prędkości w punkcie kontaktu. Zaproponowane zostało zbiorcze kryterium manipulowalności, uwzględniające właściwości konfiguracji manipulatora oraz stanu platformy mobilnej. Kinematyka manipulatora mobilnego została określona jako odpowiednie połączenie kinematyk platformy i manipulatora zależnością  $\mathbf{k}(\mathbf{q}, \mathbf{x})$ , gdzie  $\mathbf{q}$  to konfiguracja platformy, a  $\mathbf{x}$  – konfiguracja manipulatora. Kinematyka prędkościowa odpowiadająca kinematyce  $\mathbf{k}$  (*instantaneous kinematics configuration model*) jest następująca

$$\dot{\mathbf{k}} = \frac{\partial \mathbf{k}(\mathbf{q}, \mathbf{x})}{\partial \mathbf{q}} \dot{\mathbf{q}} + \frac{\partial \mathbf{k}(\mathbf{q}, \mathbf{x})}{\partial \mathbf{x}} \dot{\mathbf{x}}. \quad (4.71)$$



Przytoczono szereg przykładów wyliczania kinematyk. Następnie przytoczono pojęcie manipulowalności — dysponując kinematyką manipulatora  $\mathbf{y} = \mathbf{k}(\mathbf{x})$ , można zapisać równanie określające zależność prędkości przegubów  $\dot{\mathbf{x}}$  i prędkości przestrzeni zewnętrznej  $\dot{\mathbf{y}}$ :

$$\dot{\mathbf{y}} = \mathbf{J}(\mathbf{x})\dot{\mathbf{x}}. \quad (4.72)$$

Dla danej konfiguracji  $\mathbf{x}$ , manipulowalność określa podzbiór realizowalnych prędkości  $\dot{\mathbf{y}}$  przy ograniczeniu prędkości przegubów  $|\dot{\mathbf{x}}| \leq 1$ . Przyjmując, że  $\mathbf{J}(\mathbf{x}) = \mathbf{U}(\mathbf{x})\mathbf{\Sigma}(\mathbf{x})\mathbf{V}^T(\mathbf{x})$  jest rozkładem wartości osobliwych (*singular value decomposition*, SVD [53]) macierzy  $\mathbf{J}(\mathbf{x})$ ,  $\mathbf{U}(\mathbf{x})$  i  $\mathbf{V}(\mathbf{x})$  są ortogonalne, a  $\mathbf{\Sigma}(\mathbf{x})$  ma na diagonalu wartości własne  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_m$ . Zbiór osiągalnych prędkości  $\dot{\mathbf{y}}$  spełniających ograniczenie prędkości przegubów opisany jest równaniem:

$$\sum_{\sigma_i \neq 0} \left( \frac{\left( \frac{\partial}{\partial t} (\mathbf{U}^T(\mathbf{x})\dot{\mathbf{y}}) \right)_i}{\sigma_i} \right)^2 \leq 1. \quad (4.73)$$

Jest to równanie  $m$ -wymiarowej elipsoidy o półosiach  $\sigma_i$ . Zaproponowane miary manipulowalności to  $w_1 = \sigma_1 \sigma_2 \dots \sigma_m$ , proporcjonalna do objętości elipsoidy. Równoważną postacią tego wskaźnika jest  $w_1 = \sqrt{\det(\mathbf{J}(\mathbf{x})\mathbf{J}^T(\mathbf{x}))}$ . Inną możliwością opisu jest charakterystyka spłaszczenia elipsoidy mierzona przez  $w_2 = \sigma_m/\sigma_1$ , czyli odwrotność współczynnika uwarunkowania (*condition number*) jacobianu  $\mathbf{J}(\mathbf{x})$ . Autorzy zaproponowali miarę manipulowalności manipulatora jako

$$w = \sqrt{1 - \frac{\sigma_m^2}{\sigma_1^2}}.$$

Następnie używając kinematyki prędkościowej manipulatora mobilnego w analogiczny sposób wprowadzono manipulowalność manipulatora mobilnego. W dalszej części artykułu pokazano wykorzystanie manipulowalności do planowania ruchu metodą gradientową, gdzie maksymalizacja manipulowalności została osiągnięta klasycznymi metodami optymalizacji przestrzeni zerowej.

#### 4.9. Metoda optymalizacyjna

W artykule [26] przedstawiono „prawie optymalną” metodę planowania ruchu układu bezdryfowego (4.37) między zadanymi konfiguracjami  $\mathbf{q}_0$  i  $\mathbf{q}_f$ . Przyjęto reprezentację sterowań w postaci szeregu

$$\mathbf{u}(t) = \sum_{i=1}^{\infty} \alpha_i \mathbf{e}_i(t), \quad (4.74)$$

gdzie  $\mathbf{e}_i$  są ortonormalnymi funkcjami bazowymi, a  $\alpha_i \in \mathbb{R}$  ich współczynnikami. Zadanie planowania ruchu polega na wyznaczeniu współczynników  $(\alpha_1, \alpha_2, \dots) = \boldsymbol{\alpha}$  minimalizujących koszt  $J = \sum_{i=1}^{\infty} \alpha_i^2$ , takich, że trajektoria odpowiadająca sterowaniom (4.74) zainicjowana w  $\mathbf{q}_0$  osiąga  $\mathbf{q}_f$  w czasie  $t = T$ . Zadanie to jest równoważne nieliniowemu problemowi optymalizacyjnemu w nieskończenie-wymiarowej przestrzeni. Rozwiązanie dokładne jest trudne do uzyskania. Autorzy proponują zastosowanie teorii aproksymacji Ritza — używając skończenie-wymiarowych zadań optymalizacyjnych do przybliżenia nieskończenie-wymiarowych. W cytowanej pracy opisano sposób uzyskania skończenie-wymiarowych aproksymacji, na przykładzie integratora Brocketta

$$\begin{aligned} \dot{q}_1 &= u_1 \\ \dot{q}_2 &= u_2 \\ \dot{q}_3 &= -q_2 u_1 + q_1 u_2 \end{aligned} \quad (4.75)$$

rozpatrywanego w bazie sterowań fourierowskich

$$\begin{aligned} u_1(t) &= \sum_{i=1}^{\infty} a_i \sin it + \sum_{i=1}^{\infty} b_i \cos it \\ u_2(t) &= \sum_{i=1}^{\infty} \bar{a}_i \sin it + \sum_{i=1}^{\infty} \bar{b}_i \cos it \end{aligned} \quad (4.76)$$

z warunkami brzegowymi  $\mathbf{q}_0 = (0, 0, 0)^T$  i  $\mathbf{q}_f = (0, 0, \delta)^T$ . Całkując ostatnie równanie i dzieląc przez  $2\pi$  otrzymuje się

$$\frac{\delta}{2\pi} = \frac{q_3(2\pi) - q_3(0)}{2\pi} = \frac{1}{2\pi} \int_0^{2\pi} (-q_2 u_1 + q_1 u_2) dt = \sum_{i=1}^{\infty} \frac{1}{n} (\bar{a}_i b_i - \bar{b}_i a_i). \quad (4.77)$$

Należy rozwiązać zadanie minimalizacji kosztu

$$\|\alpha\|^2 = \sum_{i=1}^{\infty} a_i^2 + b_i^2 + \bar{a}_i^2 + \bar{b}_i^2 \quad (4.78)$$

przy ograniczeniu (4.77).

Pomocniczo pokazano, że jeżeli rozwiązywane zadanie planowania ruchu można zapisać w postaci optymalne rozwiązania dla znanego  $\mathbf{C} \in \mathbb{R}^3$ , gdzie  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^3$  spełniają ograniczenia

$$\mathbf{A} \times \mathbf{B} = \mathbf{C}$$

i minimalizują koszt

$$\|\mathbf{A}\|^2 + \|\mathbf{B}\|^2,$$

wtedy rozwiązania mają postać

$$\{(\mathbf{A}, \mathbf{B}) : \mathbf{A} \cdot \mathbf{B} = 0, \mathbf{A} \cdot \mathbf{C} = 0, \mathbf{B} \cdot \mathbf{C} = 0, \|\mathbf{A}\|^2 = \|\mathbf{B}\|^2 = \|\mathbf{C}\|\}, \quad (4.79)$$

a wartość kosztu równa jest  $2\|\mathbf{C}\|$ . Oraz zbiór  $\{\mathbf{A}_i, \mathbf{B}_i \in \mathbb{R}^3\}_{i=0}^{\infty}$  spełniający ograniczenie

$$\mathbf{C} = \sum_{i=0}^{\infty} c_i \cdot (\mathbf{A}_i \times \mathbf{B}_i) \quad \text{dla } c_1 = 1, c_i < c_1, i \geq 2$$

minimalizujący koszt

$$\sum_{i=1}^{\infty} \|\mathbf{A}_i\|^2 + \|\mathbf{B}_i\|^2$$

ma postać

$$\{\mathbf{A}_i = \mathbf{B}_i = \mathbf{0} \text{ dla } i \geq 2\},$$

a  $(\mathbf{A}_1, \mathbf{B}_1)$  spełnia (4.79). W przypadku problemu Brocketta odpowiada to następującym danym:

$$\mathbf{C} = \begin{bmatrix} 0 \\ 0 \\ \delta/2\pi \end{bmatrix} \quad \mathbf{A}_i = \begin{bmatrix} \bar{a}_i \\ \bar{b}_i \\ 0 \end{bmatrix} \quad \mathbf{B}_i = \begin{bmatrix} a_i \\ b_i \\ 0 \end{bmatrix}. \quad (4.80)$$

Następnie ograniczając bazę Fouriera do kilku pierwszych harmonicznych autorzy rozwiązali algorytmem Newtona zadanie planowania ruchu (dobór  $\alpha$ ) dla nieholonomicznego modelu spadającego kota.

Artykuł [74] wprowadza pojęcie „nieosobliwych pętli” pozwalających uzyskać sterowalność zlinearyzowanego układu. Pokazana została zbieżność metody iteracyjnej. Zastosowanie technik numerycznych do układów bezdryfowych opiera się na odpowiednim sterowaniu  $\mathbf{u}$  dającym nieosobliwe pętle:

- $\mathbf{u}$  jest nieosobliwe dla każdego  $\mathbf{q}$ ,
- po zastosowanie sterowania  $\mathbf{u}(t)$  na przedziale  $t \in [0, T]$  system zainicjowany w konfiguracji  $\mathbf{q}_1$  wróci do  $\mathbf{q}_1$ .

Zaproponowany został dobór sterowania  $\mathbf{u}$ :

$$\mathbf{u}(t) = \begin{cases} \mathbf{u}'(t) & \text{dla } t \in [0, T/2] \\ -\mathbf{u}'(T-t) & \text{dla } t \in (T/2, T], \end{cases} \quad (4.81)$$

gdzie  $\mathbf{u}'$  jest pierwotnym sterowaniem określonym dla przedziału  $[0, T/2]$ . Algorytm znajdujący rozwiązanie  $\mathbf{q}(T) = \mathbf{q}_f$  jest opisany w klasyczny sposób: wyznaczenie sterowania  $\mathbf{u}$  generującego nieosobliwą pętlę, obliczenie wpływu zastosowania takiego sterowania na stan systemu i wyliczenie linearyzacji wzdłuż odpowiedniej trajektorii, uzyskując zaburzenie sterowania zbliżające do pożądanego stanu końcowego. Do systemu stosowane jest następnie tak zmodyfikowane sterowanie. Jeśli osiągnięto stan końcowy procedura się zatrzymuje, w przeciwnym przypadku jest powtarzana. Przedstawiono także wykorzystanie skalowania do omijania przeszkód lub uproszczenia systemu. Zakładając, że  $\beta: \mathbb{R}^n \mapsto \mathbb{R}$  jest dowolnym gładkim odwzorowaniem, rozważa się nowy bezdryfowy system:

$$\dot{\mathbf{q}} = \beta(\mathbf{q})\mathbf{G}(\mathbf{q})\bar{\mathbf{u}}. \quad (4.82)$$

Jeśli dla takiego systemu znalezione zostanie sterowanie  $\bar{\mathbf{u}}$  przy użyciu którego system zainicjowany w  $\mathbf{q}_0$  osiąga stan docelowy  $\mathbf{q}_f$ , wtedy sterowanie  $\mathbf{u}(t) = \beta(\mathbf{q}(t))\bar{\mathbf{u}}(t)$  zastosowane do systemu pierwotnego (2.4) także przynosi odpowiednią zmianę stanu. Jeżeli  $\beta$  nigdy nie znika, właściwości systemu pierwotnego (2.4) i przeskalowanego (4.82) są identyczne. Zastosowanie skalowania do unikania przeszkód sprowadza się do takiego doboru  $\beta$ , który znika w momencie gdy  $\mathbf{q}$  osiąga stan zabroniony (kolizyjny).

Artykuł [9] opisuje metodę uwzględnienia ograniczeń nierównościowych w algorytmie Newtona. Zakładając system o postaci (2.4) i sterowaniach

$$\mathbf{u}(t) = \sum_{i=1}^{\infty} \alpha_i \phi_i(t) \quad (4.83)$$

w postaci bazy Fouriera

$$\begin{aligned} \phi_1(t) &= \frac{1}{\sqrt{2\pi}}, \quad \phi_2(t) = \frac{1}{\sqrt{\pi}} \cos 2\pi t, \quad \phi_3(t) = \frac{1}{\sqrt{\pi}} \sin 2\pi t, \\ \phi_4(t) &= \frac{1}{\sqrt{\pi}} \cos 4\pi t, \quad \phi_5(t) = \frac{1}{\sqrt{\pi}} \sin 4\pi t, \dots \end{aligned}$$

Zadanie polega na wyzerowaniu błędu stanu końcowego (analogicznie do (4.38)):  $\mathbf{e}(\alpha) = 0$ . Dla dyskretnej wersji algorytmu Newtona równanie modyfikacji współczynników wagowych sterowań przedstawia się następująco:

$$\alpha^{k+1} = \alpha^k - h [\nabla_{\alpha} \mathbf{F}(\mathbf{q}_0, \alpha)]^{\#} \mathbf{e}(\alpha), \quad (4.84)$$

gdzie  $k$  jest numerem iteracji,  $h > 0$  jest współczynnikiem dobieranym w każdej iteracji aby minimalizować wartość  $\|\mathbf{e}(\alpha^{k+1}(h))\|^2$ .

Następnie założono, że dopuszczalny obszar  $Q_d \subset Q$  opisany jest nierównością  $\mathbf{c}(\mathbf{q}) \leq \mathbf{0}$ . Uwzględnienie ograniczeń nierównościowych polega na wprowadzeniu funkcji kary zmieniającej pierwotny problem z ograniczeniami w problem bez ograniczeń:

$$z_i(\alpha) = \gamma_i \sum_{j=1}^N w(c_i(F_j(\alpha))), \quad (4.85)$$

gdzie  $\gamma_i > 0$ ,  $c_i$  jest  $i$ -tym ograniczeniem, a  $w(c_i)$  jest ciągłą skalarną funkcją równą zero dla  $c_i$  niedodatniego i rosnącą monotonicznie wraz ze wzrostem  $c_i$ . Wybrana została następująca funkcja

$$w(c) = \begin{cases} (1 - e^{-rc})^2 & \text{dla } c > 0 \\ 0 & \text{dla } c \leq 0 \end{cases} \quad r > 0. \quad (4.86)$$

Każdy ze składników  $z_i$  przemieszcza  $\mathbf{q}$  w stronę dopuszczalnego regionu, gdy dane ograniczenie jest naruszone. Zadanie rozszerza się do postaci: znajdź takie  $\alpha$ , że  $\mathbf{e}(\alpha) = 0$  i  $\mathbf{z}(\alpha) = 0$ . Grupując zerowane funkcje:

$$\mathbf{d}(\alpha) = \begin{bmatrix} \mathbf{e}(\alpha) \\ \mathbf{z}(\alpha) \end{bmatrix} \quad (4.87)$$

można zapisać równanie na korektę współczynników:

$$\alpha^{k+1} = \alpha^k - h [\nabla_{\alpha} \mathbf{H}(\mathbf{q}_0, \alpha)]^{\#} \mathbf{d}(\alpha) \quad (4.88)$$

gdzie

$$\mathbf{H}(\alpha^k) = \begin{bmatrix} \nabla_{\alpha} \mathbf{F}(\alpha^k) \\ \nabla_{\alpha} \mathbf{z}(\alpha^k) \end{bmatrix}. \quad (4.89)$$

#### 4.10. Planowanie trajektorii robotów mobilnych w środowisku kolizyjnym

Dla środowisk bezkolizyjnych znaleziono optymalne sterowania dla platformy typu monocykl z zabronionym ruchem wstecz [10] i ruchem wstecz dopuszczalnym [66].

W pracy [72] rozpatrzono praktyczne zadanie uzyskiwania trajektorii spełniających postulat minimalnego promienia krzywizny, zadanej promieniem skrętu  $R$ , dla robota typu monocykl. Poszukiwania odbywają się w przestrzeni kolizyjnej złożonej z przeszkód w postaci wielokątów. Algorytm planowania ruchu intensywnie korzysta z relacji geometrycznych i pracuje w obrębie ścieżek kanonicznych łączących konfiguracje kontaktowe (czyli wchodzące w kontakt z przeszkodami) kanonicznymi trajektoriami. Trajektoria kanoniczna składa się ze ścieżki zaczynającej się łukiem o promieniu  $R$ , kolejnym segmentem jest odcinek prostej lub łuk o promieniu  $R$  i kończy się ponownie łukiem o promieniu  $R$ . Dopuszcza się również przypadki zdegenerowane, pozbawione jednego z segmentów. Do łączenia trajektorii kanonicznych wykorzystywany jest graf widoczności. Gałęzie grafu oznaczają trajektorie kanoniczne, jeszcze niegładkie. Po przeszukaniu grafu, wynikowa trajektoria zostaje wygładzona przez użycie mniejszego promienia w okolicy wykrycia niegładkości. Pokazana została również metoda wyznaczenia promienia krytycznego dla danej trajektorii. Metoda ma jednak istotną wadę, gdyż działa jedynie dla jednego modelu i trudno się uogólnia.

Problemem dopuszczalności trajektorii dla planowania ruchu układów nieholonomicznych zajął się Laumond w pracy [46]. Rozważania nawigacji w ograniczonej przestrzeni prowadzi do rozważania ścieżki w przestrzeni dopuszczalnych konfiguracji. Polegając na dokładnych modelach środowiska używa niestety skomplikowanych struktur. Zdaniem cytowanego autora, większość artykułów nie dotyczy podstawowego problemu, a mianowicie dopuszczalności trajektorii. Kwestia ta nie ma znaczenia w przypadku robotów holonomicznych, natomiast jest kluczowa dla robotów nieholonomicznych. Dowodzi się, że układ holonomiczny może wykonać dowolnie mały ruch wokół konfiguracji bieżącej. Problem planowania ścieżek sprowadza się do odpowiedniego zamodelowania przestrzeni, wymagając, aby trajektoria była zawarta wewnątrz wzajemnie połączonych obszarów dopuszczalnych konfiguracji, prowadzących od punktu początkowego do końcowego.

Dokonyje się więc analizy właściwości topologicznych związanych ze spójnością przestrzeni konfiguracyjnej. Laumond zwraca uwagę na problemy z zamodelowaniem takiej przestrzeni w rzeczywistych warunkach: trudno jest wprowadzić kryteria optymalizacyjne w algorytmach przeszukiwania

przestrzeni konfiguracyjnej w połączeniu z wysoce złożonym środowiskiem robotów mobilnych, każda procedura planowania odbywająca się off-line oparta jest na niedokładnych modelach środowiska i nie zapewnia dopuszczalności trajektorii (zwłaszcza w przypadku ograniczonej przestrzeni).

Powiększanie przestrzeni kolizyjnej o odpowiednio dobrane parametry wykorzystujące geometryczne wymiary robota, redukuje robota do punktu w  $\mathbb{R}^6$ . Główny nacisk został położony na sprawne zbadanie przestrzeni dopuszczalnych konfiguracji (ACS — *Admissible Configuration Space*). Jednym z zauważonych w artykule problemów jest reprezentacja ACS - kształty przeszkód w środowisku nie dają się łatwo zapisać algorytmicznie. Większość rozważań oparta jest o proste wielokąty lub wielokąty i okręgi. Podana została metoda przybliżonego modelowania środowiska do postaci ACS, głównie na podstawie badania bezkontaktowości trajektorii (odległość między robotem a przeszkodą jest większa od ustalonego parametru  $\epsilon$ ). Problem wystarczająco dokładnej reprezentacji środowiska został uwzględniony poprzez odpowiednie dobranie  $\epsilon$ .

Dalej przeprowadzono analizę ograniczeń nieholonomicznych monocykla i pokazano dowód na istnienie i spełnienie warunku dostatecznego istnienia trajektorii w ACS. Następnie, posługując się dekompozycją całej ACS do wielu połączonych równoległoboków w których oddzielnie następuje proces poszukiwania trajektorii. Jedynie podczas tego etapu brane są pod uwagę kinematyczne i dynamiczne ograniczenia robota. Dzięki temu powstała metoda tworzenia trajektorii dopasowana bezpośrednio do wykorzystywanego robota. Nadmiarowość manewrów została obniżona przez optymalizację wynikowej trajektorii, jednakże zwrócono uwagę na trudności w wyznaczeniu odpowiednich kryteriów w niejednorodnej przestrzeni nieholonomicznej. Przedstawiony algorytm planowania ruchu składał się z etapu generowania trajektorii off-line, a później jej realizacji biorącej pod uwagę nieprzewidziane przeszkody. Podanych zostało kilkanaście typowych manewrów dla szczególnych postaci środowiska. Całość zaimplementowano fizycznie na robocie Hilare.

W artykule [29] przedstawiono rozwiązanie zadania osiągnięcia pożądanego konfiguracji przez platformę mobilną w obecności przeszkód. Zadanie zostało rozwiązane na poziomie kinematycznym. Problem znalezienia rozwiązania kinematyki odwrotnej zastąpiono zadaniem optymalizacyjnym z ograniczeniami równościowymi i nierównościowymi. Funkcja kryterialna została wybrana tak, aby minimalizować odległość pomiędzy stanem początkowym  $(\mathbf{q}_0, \mathbf{x}_0)$  a konfiguracją końcową  $(\mathbf{q}_f, \mathbf{x}_f)$ , zapewniającą osiągnięcie zadanego punktu docelowego w przestrzeni zadaniowej  $\mathbf{k}(\mathbf{q}_f, \mathbf{x}_f) = \mathbf{y}_d$ , gdzie  $\mathbf{k}$  to odwzorowanie wejściowo-wyjściowe manipulatora mobilnego. Dodatkowo przyjęto, że w stanie końcowym manipulator mobilny powinien osiągnąć maksymalną manipulowalność. Unikanie osobliwości manipulatora osiągnięto przez zastosowanie pól prędkości intensyfikujących się wraz ze zbliżaniem do osobliwości. Ciągłe zaburzenie prędkości stosowane jest do momentu ucieczki z osobliwego otoczenia. Jako funkcję kosztu przyjęto

$$F(\mathbf{q}, \mathbf{x}) = \rho' \frac{1}{2} \|(\mathbf{q}, \mathbf{x}) - (\mathbf{q}_0, \mathbf{x}_0)\|^2 + \rho'' D(\mathbf{x}),$$

gdzie  $\rho', \rho'' > 0$  są współczynnikami,  $D(\mathbf{x})$  jest miarą manipulowalności manipulatora holonomicznego. Problem rozwiązania zadania kinematyki odwrotnej manipulatora nieholonomicznego został zapisany jako następujące zadanie optymalizacyjne: zminimalizować  $F(\mathbf{q}, \mathbf{x})$  spełniając warunki (osiągnięcia położenia docelowego manipulatora nieholonomicznego i spełnienie równań kinematyki platformy)

$$\begin{aligned} \mathbf{k}(\mathbf{q}, \mathbf{x}) - \mathbf{y}_d &= \mathbf{0}, \\ \mathbf{A}(\mathbf{q})\dot{\mathbf{q}} &= \mathbf{0}. \end{aligned}$$

Unikanie osobliwości odbywa się poprzez dodanie optymalizacji w przestrzeni zerowej powodującego opuszczenie obszaru osobliwego. Podobny mechanizm został użyty do unikania kolizji z przeszkodami. Rozwiązanie generatora trajektorii zostało uzyskane z wykorzystaniem teorii Lapunowa.

Artykuł [61] omawia metodę znalezienia bezkolizyjnej ścieżki pomiędzy dwoma określonymi stanami. Zadanie zostaje sprowadzone do problemu optymalizacyjnego operującego na przestrzeni ciągłych ścieżek. Wprowadzone zostaje pojęcie odległości wzrostu, gdzie modelowane obiekty zależą od współczynnika skalującego  $\rho$ . Model obiektu „rośnie” wraz ze wzrostem  $\rho$ , odpowiada obiektowi dla  $\rho = 1$ , a w przypadku gdy  $\rho = 0$  redukuje się do jednego punktu. Zakładając, że  $A, B$  przedstawiają modelowane obiekty, wybiera się punkty  $p_A, p_B$  zawarte we wnętrzu obiektów  $p_A \in \text{int}A, p_B \in \text{int}B$ . Definiuje się  $\bar{A} = A - \{p_A\}, \bar{B} = B - \{p_B\}, A(\rho) = \{p_A\} + \rho\bar{A}, B(\rho) = \{p_B\} + \rho\bar{B}$ .  $A(\rho)$  i  $B(\rho)$  oznaczają modele wzrostu  $A$  i  $B$ . Funkcja wzrostu  $w$  określona jako

$$w(A, B) = \min \rho \in \mathbb{R}^+ : A(\rho) \cap B(\rho) \neq \emptyset \quad (4.90)$$

geometrycznie odpowiada takiej wartości  $\rho$ , przy której obiekty  $A(\rho)$  i  $B(\rho)$  zaczynają się stykać,  $w(A, B) < 1 \iff \text{int}A \cap \text{int}B \neq \emptyset$ . Następnie definiowana jest głębokość wnikięcia:

$$d(A, B) = \begin{cases} (R_A + R_B)(1 - w(A, B)) & w(A, B) < 1 \\ 0 & w(A, B) \geq 1, \end{cases} \quad (4.91)$$

gdzie  $R_A, R_B$  są promieniami sfer zawierających  $A$  i  $B$ . Jeśli  $A$  i  $B$  są wielościanami wypukłymi wyznaczenie  $w(A, B)$  sprowadza się do rozwiązania zadania programowania liniowego. Zaletą tak sformułowanej funkcji jest prostota obliczeń zarówno jej wartości jak i jej pochodnych po zmiennych konfiguracyjnych. Określana jest funkcja miary penetracji na całej długości ścieżki

$$F(\mathbf{q}(\cdot)) = \int \sum_{i,j} d(O_i(\mathbf{q}), O_j(\mathbf{q})) dt, \quad (4.92)$$

gdzie  $O_i, O_j$  są obiektami. Jeśli  $F(\mathbf{q}) = 0$  oznacza to, że ścieżka jest bezkolizyjna, co oznacza, że obiekty co najwyżej się stykają. Zadanie planowania ścieżki staje się równoważne zadaniu optymalizacyjnemu. Należy znaleźć  $\min F(\mathbf{q})$  spełniającą warunki brzegowe

$$\mathbf{q}(0) = \mathbf{q}_0, \quad \mathbf{q}(1) = \mathbf{q}_f,$$

i jeśli jest to wymagane, spełniającą ograniczenia nałożone na zmienne konfiguracyjne (np. ograniczenie ruchu przegubu). Inicjującą obliczenia ścieżka może być kolizyjna. Zadanie jest rozwiązywane metodą programowania liniowego.

W artykule [31] zaproponowano metodę planowania ruchu manipulatora o wielu stopniach swobody w środowisku kolizyjnym. Kluczem do efektywnego rozwiązywania zadania planowania jest dekompozycja manipulatora na kilka pod-łańcuchów i użycie etapu przygotowawczego *off-line*. Podczas tego etapu wyznaczane są konfiguracje kolizyjne ze statycznymi przeszkodami. Dla manipulatora o  $n$  stopniach swobody liczba podziałów  $p$  jest ograniczona zależnością

$$2 \geq p \geq \frac{n}{2},$$

aby każdy z pod-łańcuchów mógł mieć odpowiednią manipulowalność do unikania kolizji (miał przynajmniej 2 stopnie swobody). Podane zostały kryteria, aby zdyskretyzowana reprezentacja stanów pod-łańcuchów zapewniała odpowiednie pokrycie przestrzeni roboczej. Kolizyjność stanów jest badana standardowymi metodami. Etap planowania *on-line* składa się z trzech faz: odnalezienia par zdyskretyzowanych póz każdego z  $p$  pod-łańcuchów reprezentujących konfiguracje początkowe i końcowe; połączenia konfiguracji początkowych i końcowych poszczególnych pod-łańcuchów oraz syntezy w finalną ścieżkę. Planowanie ścieżek poszczególnych pod-łańcuchów odbywa się niezależnie. Symulacje przeprowadzono na robocie PUMA oraz manipulatorze promu kosmicznego.

W pracy [2] przedstawiono wykorzystanie metody elastycznej wstęgi do planowania ruchu robotów w dynamicznie zmieniającym się otoczeniu. Wykonywanie zaplanowanej ścieżki jest kontrolowane w czasie rzeczywistym, uwzględniając zmiany w otoczeniu robota. Zmiana wykonywanego manewru w skutek reakcji na przeszkody nie zaburza śledzenia ścieżki. W przypadku braku możliwości kontynuowania ruchu zadanie jest wstrzymywane do czasu ustąpienia przeszkód. Przyjmuje się zadaną ścieżkę jako zbiór dyskretnych konfiguracji, na które działają siły wywodzące się z przestrzeni roboczej. Siły są określone jako dwie funkcje potencjału. Potencjał zewnętrzny, odpychający  $V_{ext}$  jest określony jako funkcja odległości od przeszkód:

$$V_{ext}(\mathbf{q}) = \begin{cases} \frac{1}{2}\alpha_r (d_0 - d(\mathbf{q}))^2 & \text{jeśli } d(\mathbf{q}) < d_0 \\ 0 & \text{w przeciwnym przypadku,} \end{cases} \quad (4.93)$$

gdzie  $\mathbf{q}$  jest konfiguracją,  $d(\mathbf{q})$  jest odległością od  $\mathbf{q}$  do najbliższej przeszkody,  $d_0$  określa zasięg oddziaływania wokół przeszkód, a  $\alpha_r$  jest współczynnikiem wzmocnienia. Minimalizacja potencjału powoduje maksymalizację odległości od przeszkód. Siła wywołana potencjałem (4.93) działająca na punkt  $\mathbf{q}$  zadana jest zależnością

$$\mathbf{F}_{ext}(\mathbf{q}) = -\nabla V_{ext} = \alpha_r (d_0 - d(\mathbf{q})) \frac{\vec{d}}{\|\vec{d}\|},$$

gdzie  $\vec{d}$  jest wektorem pomiędzy  $\mathbf{q}$  a najbliższym punktem przeszkody. Aby zapobiec niekontrolowanemu wydłużeniu ścieżki pod wpływem sił odpychających wprowadza się także siłę wiążącą poszczególne węzły:

$$\mathbf{F}_{int}(\mathbf{q}) = \alpha_c \left( \frac{d_j^{i-1}}{d_j^{i-1} + d_j^i} (\mathbf{q}_j^{i+1} - \mathbf{q}_j^{i-1}) - (\mathbf{q}_j^i - \mathbf{q}_j^{i-1}) \right), \quad (4.94)$$

gdzie  $d_j^i = \|\mathbf{q}_j^i - \mathbf{q}_j^{i+1}\|$  jest odległością w niezmodyfikowanej trajektorii,  $\alpha_c$  współczynnikiem kurczenia. Na przykładzie robota humanoidalnego wprowadzono dodatkowo czynnik stabilizujący położenie środka masy  $(x_{sm}, y_{sm})$  — w postaci potencjału

$$V_{poz}(\mathbf{q}) = \frac{1}{2}\alpha_p (x_{sm}^2 + y_{sm}^2),$$

gdzie  $\alpha_p$  jest współczynnikiem liczbowym.

W artykule [7] przedstawiono lokalną metodę nawigacji wśród przeszkód. Metoda nie zapewnia dokładnego osiągnięcia konfiguracji końcowej. Algorytm opiera się na bezpośrednim sterowaniu prędkościowym robotem wykorzystując kinematykę robota, z bezpośrednim zastosowaniem pseudoodwrótności.

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{G}(\mathbf{q})\mathbf{u} \\ \mathbf{u} &= \mathbf{G}^\#(\mathbf{q})\dot{\mathbf{q}}_d = (\mathbf{G}^T(\mathbf{q})\mathbf{G}(\mathbf{q}))^{-1}\mathbf{G}^T(\mathbf{q})\dot{\mathbf{q}}_d, \end{aligned}$$

gdzie  $\dot{\mathbf{q}}_d(t)$  oznacza prędkość zmian na pożądaney trajektorii. Przyrostowe prędkości  $\dot{\mathbf{q}}_d$  generowane były na podstawie holonomicznego planera uwzględniającego przeszkody z zastosowaniem formuły

$$\dot{\mathbf{q}}_d = -\nabla_{\mathbf{q}} U(\mathbf{q}) = -\nabla_{\mathbf{q}} (U_a(\mathbf{q}) + U_r(\mathbf{q})),$$

gdzie  $U_a(\mathbf{q})$  jest polem przyciągającym, a  $U_r(\mathbf{q})$  odpychającym; określonymi jako

$$\begin{aligned} U_a(\mathbf{q}) &= \frac{k_a}{2} \|\mathbf{q} - \mathbf{q}_d\|^2, \\ U_r(\mathbf{q}) &= \begin{cases} \frac{k_r}{\gamma} \left( \frac{1}{\eta(\mathbf{q})} - \frac{1}{\eta_0} \right)^\gamma & \eta(\mathbf{q}) \leq \eta_0, \\ 0 & \text{w przeciwnym przypadku,} \end{cases} \end{aligned}$$

gdzie  $\eta(\mathbf{q})$  jest minimalną odległością od przeszkód w stanie  $\mathbf{q}$ ,  $\eta_0$  zasięgiem oddziaływania pola odpychającego,  $\gamma \geq 2$  współczynnikiem kształtu charakterystyki. Aby zniwelować wpływ minimów lokalnych pól potencjałów, autorzy proponują wykorzystanie metody pól wirowych, zastępując odpychanie antygradientowe przeszkód opływaniem wokół przeszkody. Odpowiada to ruchowi wzdłuż kierunku

$$\mathbf{F}_v(\mathbf{q}) = \pm \begin{bmatrix} \frac{\partial U_r(\mathbf{q})}{\partial y} \\ -\frac{\partial U_r(\mathbf{q})}{\partial x} \end{bmatrix},$$

gdzie  $x, y$  to współrzędne położeniowe. Wybór znaku zależy od kierunku okrążania przeszkody. Zasygnalizowano także możliwość użycia ważonej pseudoodwrotności celem wyrównania błędów i zrównoważenia niehomogenicznych jednostek (położenie, kąt). Dane z planera sterowały bezpośrednio robotem. Symulacje zostały przeprowadzone na monocyklu.

W artykule [70] przedstawiono metodę planowania ruchu robota typu samochód kinematyczny o wektorze stanu  $\mathbf{q} = (x, y, \phi, \theta)^T$  i ograniczonym promieniu skrętu. Zastosowano podejście numeryczne generujące bezpośrednio trajektorię nieholonomiczną. Problem znalezienia ścieżki przekształcony został w skończenie-wymiarowy problem optymalizacji nieliniowej. Kryteriami minimalizacyjnymi były całkowita długość ścieżki na unormowanym horyzoncie czasu  $[0, 1]$ ,  $J_1(\mathbf{q}(\cdot))$ , oraz zmiana orientacji  $J_2(\mathbf{q}(\cdot))$ :

$$J_1(\mathbf{q}(\cdot)) = \int_0^1 \sqrt{\dot{x}^2 + \dot{y}^2} dt = \int_0^1 |u_1| dt, \quad J_2(\mathbf{q}(\cdot)) = \int_0^1 \left| \frac{d\phi}{dt} \right| dt = \int_0^1 |u_2| dt,$$

gdzie  $u_1, u_2$  są, odpowiednio, sterowaniami ruchu postępowego i obrotowego koła napędowego. Zapewnienie bezkolizyjności zrealizowano przez odpowiednio dobraną karę za zbliżanie do przeszkód. Zdefiniowano ekspansywną odległość pomiędzy zbiorami  $\mathbf{P}$  a  $\mathbf{Q}$  jako

$$d(\mathbf{P}, \mathbf{Q}) = \min\{\lambda : \bar{\mathbf{p}} + \lambda\mathbf{P} \cap \bar{\mathbf{q}} + \lambda\mathbf{Q} \neq \emptyset\} - 1,$$

gdzie  $\bar{\mathbf{p}}, \bar{\mathbf{q}}$  są pewnymi punktami odniesienia wewnątrz zbiorów  $\mathbf{P}, \mathbf{Q}$ . Zadanie optymalizacyjne sformułowano jako minimalizację

$$\min_{\mathbf{u}(\cdot)} J_1(\mathbf{q}(\cdot)) + J_2(\mathbf{q}(\cdot)),$$

z zapewnieniem spełnienia ograniczeń:

$$\mathbf{q}(0) = \mathbf{q}_0, \quad \mathbf{q}(1) = \mathbf{q}_f, \quad \forall t \in [0, 1], i = 1, \dots \quad d(\mathbf{C}(\mathbf{q}), \mathbf{O}_i) \geq 0,$$

gdzie  $\mathbf{C}(\mathbf{q})$  jest przestrzenią zajmowaną przez platformę w stanie  $\mathbf{q}$ , natomiast  $\mathbf{O}_i$  jest  $i$ -tą przeszkodą. W praktyce, dla uniknięcia rozpatrywania zadania z ograniczeniami, zastosowano minimalizację funkcji kryterialnej zadanej wzorem

$$J_1 + J_2 + c_1 \|\mathbf{q}(1) - \mathbf{q}_f\|^2 + c_2 \sum_{i=1}^n \int_0^1 p(d(\mathbf{C}(\mathbf{q}(t)), \mathbf{O}_i)) dt,$$

gdzie  $c_1, c_2$  są dużymi współczynnikami liczbowymi, a  $p(\cdot)$  oznacza funkcję kary. Parametrami optymalizacyjnymi były odcinkami stałe sterowań na poszczególnych etapach ruchu. Zaproponowana procedura generuje globalnie bezkolizyjną ścieżkę. W sytuacjach dużej gęstości przeszkód w otoczeniu robota, algorytm wspomagany jest wstępnym planowaniem mapy geometrycznej, a następnie planer wyznacza trajektorie pomiędzy odpowiednio dobranymi punktami mapy.

W artykule [24] proponuje się wykorzystanie programowania dynamicznego do uzyskania dopuszczalnych trajektorii nieholonomicznych w obecności przeszkód. Przeszkody zamodelowano jako



ograniczenia w przestrzeni dopuszczalnych stanów. Autorzy proponują wykorzystanie metody progresywnych ograniczeń (PC — *progressive constraints*). Startując od początkowej ścieżki spełniającej jedynie ograniczenia holonomiczne w każdym przebiegu stopniowo dodawane są ograniczenia nieholonomiczne, zawężając dopuszczalną przestrzeń fazową. Programowanie dynamiczne zapewnia ewolucję trajektorii adaptując ją do wprowadzanych ograniczeń tak, że w każdym kroku istnieje dopuszczalna ścieżka, między stanem początkowym a końcowym, uwzględniająca aktualne ograniczenia. Modyfikacja bieżącej ścieżki odbywa się przez wprowadzenie dodatkowych ruchów wykonanych ze stanów uprzednio istniejących. Zaproponowano ignorowanie konfiguracji zbliżonych do już obecnych celem zapobieżenia nadmiernej ekspansji grafu. Symulacje przeprowadzono na modelu ciągnika z przyczepami. Wprowadzono także dodatkowe wymaganie na ciągłość sterowań, nie rozważane na etapie opisu algorytmu. Kryterium optymalizacyjnym jakości ruchu była minimalizacja liczby cofnięć robota. Przykład zastosowania metody dla modelu dźwigu przenoszącego ciężki ładunek przedstawiony jest w artykule [25].

W artykule [64] przedstawiono zasady doboru funkcji kary dla zadania omijania przeszkód nie powodujący nadmiernego zwiększenia obszaru osobliwości sterowań. Pokazano rozszerzenie metody iteracji w przestrzeni ścieżek dla dowolnych układów nieholonomicznych możliwych do sprowadzenia do postaci łańcuchowej w obecności dowolnych przeszkód. Autorzy zwrócili uwagę, że w klasycznych algorytmach newtonowskich i metodach kontynuacji jedną z trudności jest zdolność algorytmu do unikania generacji sterowań osobliwych. Algorytmy gradientowe autorzy uważają za bardziej odporne w tej kwestii, jako, że metody kontynuacji wymagają dokładnego określenia sterowań osobliwych. Dla układu łańcuchowego (4.10) jedynym sterowaniem osobliwym jest  $u_1(t) = 0$  prawie wszędzie dla  $t \in [0, T]$ .

Klasyczny sposób unikania przeszkód w metodzie kontynuacji polega na wyborze funkcji  $h(\mathbf{q})$ , której elementy są niezerowe (duże) wewnątrz przeszkody i zerowe na zewnątrz. Funkcja kary jest obliczana wzdłuż ścieżki poprzez całkowanie (dla czasu ciągłego) lub sumowanie (dla czasu dyskretnego) dając funkcję kosztu  $z(\mathbf{u}(\cdot))$ . Następnie algorytm popraw sterowań uwzględnia funkcję kosztu, minimalizując ją, oprócz wymuszania osiągnięcia stanu docelowego, jako dodatkowe kryterium. Dlatego zapewnione jest osiągnięcie zarówno konfiguracji docelowej  $\mathbf{q}_f$  jak i bezkolizyjność uzyskanej ścieżki. Na podstawie wcześniejszych prac, aby uprościć obliczenia, zaproponowano wielowymiarową postać funkcji kary:

$$z_i(\mathbf{u}) = \gamma_i \sum_{j=1}^N h(c_i(\mathbf{q}(t_j))), \quad (4.95)$$

gdzie  $N$  jest liczbą punktów, na które ścieżka została podzielona,  $i \in [1, n]$ ,  $\gamma_i > 0$  jest stałą,  $h(c)$  to nieujemna funkcja skalarna, niezerową tylko dla  $c > 0$ ,  $c_i(\mathbf{q}) < 0$  są składnikami funkcji ograniczającej opisujące dopuszczalny obszar, wolny od przeszkód.

W artykule [44] przedstawiono metodę sterowania samochodem kinematycznym z ograniczeniem na krzywiznę wynikowej ścieżki. Rozważono samochód jako układ czterowymiarowy z punktu widzenia kinematyki (uwzględniając kąt skrętu kół) i jako trójwymiarowy na potrzeby sprawdzania kolizyjności. Samochód kinematyczny (o długości 1) opisany jest równaniem:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta \cos \phi \\ \cos \theta \sin \phi \\ \sin \theta \\ 0 \end{bmatrix} v(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \omega(t) = \mathbf{g}_1(\mathbf{q})v(t) + \mathbf{g}_2(\mathbf{q})\omega(t), \quad (4.96)$$

gdzie  $v$  i  $\omega$  są odpowiednio prędkością ruchu samochodu oraz prędkością zmiany kąta skrętu kierownicy. Przestrzeń konfiguracyjna dla samochodu kinematycznego  $Q = \mathbb{R}^2 \times (\mathbb{S}^1)^2$  jest czterowymiarowa. Kąt skrętu jest ograniczony fizycznie  $|\omega| < \omega_{max}$ . Konfiguracja  $\mathbf{q} = (x, y, \phi, \theta)$  jest dopuszczalna

jeśli  $|\omega| < \omega_{max}$ . Przyjęto w artykule ograniczenia na prędkość postępową  $|v| < 1$  i skreću kół  $|\omega| < 1$ . Autorzy zaznaczają, że w wielu pracach nie uwzględnia się kąta skreću kół napędowych. Następuje uproszczenie modelu, przez podstawienie  $u_1 = v \cos \theta$ ,  $u_2 = \omega \sin \theta$ , dając w efekcie model monocykla (6.11):

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \phi \\ \sin \phi \\ 0 \end{bmatrix} u_1(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_2(t) \quad (4.97)$$

Zasadniczą różnicą jest zmiana przestrzeni dopuszczalnych sterowań. Ograniczenia na  $u_1$  i  $u_2$  przestają być niezależne. Do rozwiązania zadania planowania autorzy posługują się *ścieżkami kanonicznymi* i *krzywymi kanonicznymi*. Przytaczają, że styczna do ścieżki równa jest orientacji  $\phi$  samochodu, a kąt skreću przednich kół  $\omega$  jest związany z krzywizną  $\kappa$  ścieżki zależnością  $\kappa = \tan \omega$ . Dla danej konfiguracji  $\mathbf{q} = (x, y, \phi, \kappa)$  istnieje jednoznacznie określona dopuszczalna ścieżka przechodząca przez  $\mathbf{q}$  i utrzymująca stałą wartość  $\kappa$ . Taką *ścieżkę kanoniczną* otrzymuje się przez całkowanie systemu (4.96) dla  $v = 1, \omega = 0$ . Odpowiadająca ścieżka jest łukiem, jeśli  $\kappa \neq 0$ , albo prostą, gdy  $\kappa = 0$ . Ścieżka jest parametryzowana długością  $s$  dając *krzywą kanoniczną*  $\gamma(\mathbf{q}, s)$  związaną z  $\mathbf{q}$ . Na podstawie krzywych kanonicznych budowana jest następnie dopuszczalna ścieżka pomiędzy dwoma konfiguracjami. W tym celu zdefiniowana została pomocnicza gładka monotonicznie rosnąca funkcja  $\xi : [0, 1] \mapsto [0, 1]$  spełniająca właściwości:  $\xi(0) = 0, \xi(1) = 1, \dot{\xi}(0) = 0, \dot{\xi}(1) = 0, \ddot{\xi}(0) = 0, \ddot{\xi}(1) = 0$ . Przyjmując, że  $\mathbf{q}_1$  i  $\mathbf{q}_2$  są odpowiednio konfiguracjami początkowymi i końcowymi ścieżki zdefiniowano krzywą  $P(t) = (1 - \xi(t))\gamma(\mathbf{q}_1, t) + \xi(t)\gamma(\mathbf{q}_2, t - 1)$ . Dla  $t = 0$  krzywa  $P(\cdot)$  ma taką samą pozycję, styczną i krzywiznę jak  $\gamma(\mathbf{q}_1, t)$  oraz analogicznie dla  $\gamma(\mathbf{q}_2, t - 1)$  przy  $t = 1$  — odpowiada dopuszczalnej ścieżce prowadzącej z  $\mathbf{q}_1$  do  $\mathbf{q}_2$ . Przy tak zdefiniowanej krzywej,  $\kappa$  jest ciągłe.

Autorzy zaproponowali wykorzystanie opisanej strategii planowania w środowiskach kolizyjnych na przykładzie dwóch metod: przybliżenia ścieżki holonomicznej, używając planera geometrycznego do pierwszej fazy planowania, oraz planera probabilistycznego. W obydwu przypadkach krokiem dopełniającym była aproksymacja ścieżki przez podział na mniejsze podzadania realizowane nieholonomicznie.

Nieholonomiczność robota ma nie tylko wpływ na planowanie ruchu, ale także na relacje robota względem przeszkód. W szczególności oznacza to, w ogólnym przypadku, nieeuklidesowość odległości między robotem a przeszkodami [90].

#### 4.11. Metody pomocnicze w przestrzeniach kolizyjnych

W pracy [6] przedstawiono ogólne podejście do efektywnej reprezentacji przeszkód w otoczeniu robota wyrażonych we współrzędnych konfiguracyjnych. Niech  $A : \mathbb{Q} \times \mathbb{Y} \mapsto \mathbb{R}$  oraz  $B : \mathbb{Y} \mapsto \mathbb{R}$  będą zdefiniowane następująco:

$$A(\mathbf{q}, \mathbf{y}) = \begin{cases} 1 & \text{jeśli } \mathbf{y} \in \mathbf{A}(\mathbf{q}) \\ 0 & \text{jeśli } \mathbf{y} \notin \mathbf{A}(\mathbf{q}) \end{cases}, \quad B(\mathbf{y}) = \begin{cases} 1 & \text{jeśli } \mathbf{y} \in \mathbf{B} \\ 0 & \text{jeśli } \mathbf{y} \notin \mathbf{B} \end{cases}, \quad (4.98)$$

gdzie  $\mathbf{A}(\mathbf{q}) \subset \mathbb{Y}$  jest zbiorem punktów reprezentujących robota w konfiguracji  $\mathbf{q}$ , a  $\mathbf{B} \subset \mathbb{Y}$  oznacza przeszkody. Niech  $CB : \mathbb{Q} \mapsto \mathbb{R}$  będzie funkcją:

$$\forall \mathbf{q} \in \mathbb{Q} \quad CB(\mathbf{q}) = \int A(\mathbf{q}, \mathbf{y}) B(\mathbf{y}) d\mathbf{y}. \quad (4.99)$$

Można dowieść, że  $\mathbf{q} \in \mathbb{Q}_{free} \iff CB(\mathbf{q}) = 0$ , gdzie  $\mathbb{Q}_{free} \subset \mathbb{Q}$  jest podzbiorem konfiguracji, w których robot nie jest w kolizji z żadną z przeszkód. Aby umożliwić wyznaczenie wartości funkcji  $A, B$  oraz  $CB$  konieczny jest dobór odpowiedniej parametryzacji  $\mathbb{Q}, \mathbb{Y}$ . Przyjęte zostały dwa układy odniesienia  $F_{\mathbb{Q}}, F_{\mathbb{Y}}$ , z których  $F_{\mathbb{Q}}$  jest związany z robotem, a  $F_{\mathbb{Y}}$  jest nieruchomy. Wybiera się współrzędne

$(y_1, \dots, y_r) \in \mathbb{Y}$  i  $(q_1, \dots, q_n) \in \mathbb{Q}$ , które opisują położenie i orientację  $F_{\mathbb{Q}}$  względem  $F_{\mathbb{Y}}$ . Podstawiając współrzędne do (4.98) otrzymuje się

$$\begin{aligned} A(q_1, \dots, q_n, y_1, \dots, y_r) &= \begin{cases} 1 & \text{jeśli } (y_1, \dots, y_r) \in \mathbf{A}(q_1, \dots, q_n) \\ 0 & \text{jeśli } (y_1, \dots, y_r) \notin \mathbf{A}(q_1, \dots, q_n) \end{cases} \\ B(y_1, \dots, y_r) &= \begin{cases} 1 & \text{jeśli } (y_1, \dots, y_r) \in \mathbf{B} \\ 0 & \text{jeśli } (y_1, \dots, y_r) \notin \mathbf{B}. \end{cases} \end{aligned} \quad (4.100)$$

Natomiast dla  $CB$  zadanego przez (4.99) mamy

$$CB(q_1, \dots, q_n) = \int A(q_1, \dots, q_n, y_1, \dots, y_r) B(y_1, \dots, y_r) dy_1 \cdots dy_r. \quad (4.101)$$

Następnie autorzy redukują liczbę zmiennych występujących w funkcji  $A(q_1, \dots, q_n, y_1, \dots, y_r)$  przez wprowadzenie nowej funkcji  $A'(0, \dots, 0, q_{s+1}, \dots, q_n, y_1 - q_1, \dots, y_s - q_s, y_{s+1}, \dots, y_r)$ . Do tego celu wprowadzono przekształcenie:

$$\phi : \mathbb{Q} \times \mathbb{Y} \mapsto \mathbb{Q}' \times \mathbb{Y}'$$

zdefiniowanego jako:

$$\begin{aligned} \forall j \in \{1, \dots, s\} \quad y'_j &= y_j - q_j, & q'_j &= 0 \\ \forall i \in \{s+1, \dots, n\} \quad q'_i &= q_i, \\ \forall k \in \{s+1, \dots, r\} \quad y'_k &= y_k, \end{aligned} \quad (4.102)$$

gdzie  $1 \leq s \leq r$ , takie że funkcja  $A = (A' \circ \phi)$  spełnia

$$\forall j \in \{1, \dots, s\} \quad \frac{\partial A'}{\partial q'_j} = 0. \quad (4.103)$$

Dzięki temu, równanie (4.101) może być zapisane jako

$$CB(q_1, \dots, q_n) = \int (A' \circ \phi)(q_1, \dots, q_n, y_1, \dots, y_r) B(y_1, \dots, y_r) dy_1 \cdots dy_r. \quad (4.104)$$

Przez podstawienie

$$\begin{aligned} \bar{A}'(0, \dots, 0, q_{s+1}, \dots, q_n, q_1 - y_1, \dots, q_s - y_s, y_{s+1}, \dots, y_r) &= \\ A'(0, \dots, 0, q_{s+1}, \dots, q_n, y_1 - q_1, \dots, y_s - q_s, y_{s+1}, \dots, y_r) \end{aligned} \quad (4.105)$$

uzyskuje się postać  $CB$  (4.104) zapisaną z wykorzystaniem operatora splotu:

$$CB(q_1, \dots, q_n) = \int (\bar{A}'_{(0, \dots, 0, q_{s+1}, \dots, q_n)} * B)(q_1, \dots, q_n, y_1, \dots, y_r) dy_1 \cdots dy_r \quad (4.106)$$

do której można zastosować wysoce zoptymalizowane algorytmy obliczeń splotowych.

Przy użyciu zaproponowanej metody i dla odpowiedniego doboru układów współrzędnych zarówno w przestrzeni konfiguracyjnej jak i zadaniowej uzyskuje się reprezentację przeszkód wyrażoną jako spłot dwóch funkcji — opisujących robota i przeszkody. Użycie właściwości splotu pozwala na znaczące zmniejszenie złożoności obliczeń. Przedstawiona metoda może być bezpośrednio stosowana zarówno do robotów manipulacyjnych jak i mobilnych bez potrzeby dopasowania lub ograniczenia kształtu robota czy przeszkód.

### 4.12. Ocena przydatności ścieżek

Praca [15] wprowadza pojęcie miary zręczności dla bezdryfowych układów nieholonomicznych. Oznaczmy zbiór jednomianów Liego modulo własność antysymetrii dla dwóch generatorów jako:

$$LM = \{g_1, g_2, [g_1, g_2], [g_1, [g_1, g_2]], [g_2, [g_1, g_2]], \dots\} = \{LM_1, LM_2, \dots\} \quad (4.107)$$

oraz niezależny zbiór jednomianów tworzących bazę Ph. Halla [89]  $PH = \{PH_1, PH_2, \dots\}$ ,  $PH \subset LM$ . Niech  $\varphi(Z)$  oznacza koszt ruchu wzdłuż pola  $Z$ :

$$\begin{aligned} \varphi(X) = \varphi(Y) = t, \quad \varphi([X, Y]) &= 4\sqrt{t} \\ \varphi([X, [X, Y]]) = \varphi([Y, [X, Y]]) &= 8\sqrt[3]{t} \end{aligned} \quad (4.108)$$

gdzie  $t$  jest elementarnym czasem ruchu. Miara zręczności dla układów nieholonomicznych została zdefiniowana jako:

$$NH(q) = \max_{\xi \in P} |\det(LM(\xi)(q))| \cdot \frac{\varphi(PH(n), t)}{\varphi(LM(\xi), t)}, \quad (4.109)$$

gdzie  $P$  jest zbiorem wszystkich  $n$ -elementowych podzbiorów zbioru  $\{1, 2, \dots\}$ ,  $\xi = (\xi_1, \dots, \xi_n)$  wektorem indeksującym,  $LM(\xi)(q) = [LM_{\xi_1}(q) \ LM_{\xi_2}(q) \ \dots \ LM_{\xi_n}(q)]$  kwadratową macierzą jednomianów Liego wybraną przez  $\xi$  ze zbioru (4.107) i wyliczoną w  $q$ ,  $\varphi(PH(n), t) = \prod_{i=1}^n \varphi(PH_i, t)$ ,  $\varphi(LM(\xi), t) = \prod_{i=1}^n \varphi(LM_{\xi_i}, t)$ . Mimo, że elementy bazy Ph. Halla są niezależne w wolnej algebrze Liego, może się jednak okazać, że  $n$  pierwszych elementów wyliczonych w konkretnym stanie systemu może być zależne jako wektory. Miara ta oparta o właściwości algebry Liego w każdym punkcie przestrzeni stanu wybiera zestaw wektorów zapewniających pełną zdolność manewrową przy jednoczesnym minimalizowaniu wydatków energetycznych związanych z manewrem.

Do analizy zjawisk nieliniowych warto wykorzystać miary opisujące nieliniowości. Jedną z takich propozycji przedstawiono w artykule [76] do wyznaczania miary nieliniowości układu przy ograniczonych sterowaniach. Niech będzie dana sekwencja  $u(k)$ . Obciążenie  $u_K(k)$  sterowania jest zdefiniowane jako

$$u_K(k) = \begin{cases} u(k) & 0 \leq k \leq K \\ 0 & k > K. \end{cases} \quad (4.110)$$

Skrócona norma  $\|\cdot\|_{p,K}$  sekwencji  $u(k)$  jest zdefiniowana zależnością  $\|u\|_{p,K} = \|u_K\|_p$ . Dla nieliniowego operatora  $G: \ell_1 \mapsto \ell_2$  i zbioru  $L$  liniowych operatorów  $L: \ell_1 \mapsto \ell_2$  miara nieliniowa na ograniczonym zbiorze sterowań  $U$  przy skończonym horyzoncie  $K < \infty$  jest zdefiniowana przez

$$v_K(G, U) = \inf_{L \in L} \sup_{u \in U} \|(Gu - Lu)_K\|_p, \quad (4.111)$$

a dla nieskończonego horyzontu jest określona jako

$$v(G, U) = \inf_{L \in L} \sup_{u \in U} \|(Gu - Lu)\|_p. \quad (4.112)$$

W artykule zaproponowano procedurę relaksacji, która zapewnia zbieganie kresów dolnego i górnego do wartości miary. Dla nieliniowego BIBO-stabilnego systemu wyznaczenie kresu dolnego sprowadza się do sekwencji zadań programowania liniowego. Artykuł kończy się symulacyjną weryfikacją obliczania normy nieliniowej.

W pracy [73] zwrócono uwagę na jakość ścieżek używanych w klasycznym dwuetapowym algorytmie planowania — najpierw wyznaczana jest holonomiczna bezkolizyjna ścieżka, a następnie jest ona przekształcana w ciąg ścieżek dopuszczalnych, możliwych do realizacji przez platformę. Wprowadzone zostało pojęcie kosztu uzyskania określonego przemieszczenia w przestrzeni stanu. Koszt

został zdefiniowany na podstawie zależności geometrycznych związanych z robotem typu monocykl. Przyjęto, że koszt elementu ścieżki utrzymującej stałą pozycję (zmiana tylko orientacji) jest zerowy, a odcinka ścieżki na której robot się przemieszcza:

$$C_{nh}(\mathbf{q}_1, \mathbf{q}_2) = d_{euc}(\mathbf{q}_1, \mathbf{q}_2) \cdot |\sin(\varphi_1 - \varphi_2)|, \quad (4.113)$$

gdzie  $d_{euc}(\cdot, \cdot)$  jest odległością euklidesową pomiędzy położeniami,  $\varphi_1$  jest początkowym kątem obrotu robota, a  $\varphi_2$  pożądaną orientacją na końcu segmentu. Zdefiniowano odległość w  $\mathbb{R}^2 \times \mathbb{S}^1$ :

$$D(\mathbf{q}_1, \mathbf{q}_2) = |\varphi_1 - \varphi_{12}| + d_{euc}(\mathbf{q}_1, \mathbf{q}_2) + |\varphi_{12} - \varphi_2|, \quad (4.114)$$

gdzie  $\varphi_{12}$  jest testową orientacją punktu pośredniego, dobieraną heurystycznie. Zaproponowano następującą postać kosztu całkowitego segmentu ścieżki:

$$C(\mathbf{q}_1, \mathbf{q}_2) = D(\mathbf{q}_1, \mathbf{q}_2) + \alpha \cdot C_{nh}(\mathbf{q}_1, \mathbf{q}_2), \quad (4.115)$$

gdzie  $\alpha \in \mathbb{R}$  jest współczynnikiem wagowym. Całkowity koszt ścieżki jest sumą kosztów kawałków wchodzących w jej skład. Minimalno-kosztowa ścieżka jest wyznaczana algorytmem heurystycznym. Rozważany w artykule robot miał kształt wielokąta, tak jak przeszkody w otoczeniu. Zaproponowano także konstrukcyjny algorytm planowania ruchu.

Dosyć często podczas planowania ruchu układów nieholonomicznych najpierw planowana jest ścieżka holonomiczna, a następnie, na jej podstawie, generowany jest jej nieholonomiczny odpowiednik. W artykule [71] rozważono wpływ jakości pierwotnej ścieżki holonomicznej na rezultaty planowania nieholonomicznego. Zauważono, że pewne ścieżki są łatwiej realizowalne niż inne. Na podstawie właściwości pól wektorowych przedstawiono nieholonomiczną miarę ścieżek

$$\|\dot{\mathbf{q}}\|_{\mathbf{X}, \mu}^2 = \sum_{i=1}^n \mu_i \cdot (w_i x_i)^2$$

gdzie  $\dot{\mathbf{q}}$  oznacza infinitezymalny wektor kierunku w konfiguracji  $\mathbf{q}$ , który jest rozkładany w bazie pól wektorowych (schematycznie oznaczonych jako  $\mathbf{X}$ ) rozpinających przestrzeń stanu w tej konfiguracji (i spełniających zarazem założenia twierdzenia Chow),  $\mu_i$  są dodatnimi stałymi,  $x_i$  jest współrzędną  $\dot{\mathbf{q}}$   $i$ -tego pola wchodzącego w skład zbioru pól rozpinających przestrzeń stanu w zadanej konfiguracji, a  $w_i$  stopniem tego pola. Koszt nieholonomiczny ścieżki  $\gamma(\cdot)$  określonej na horyzoncie czasowym  $[0, 1]$  jest zadany zależnością

$$J(\gamma(\cdot)) = \int_0^1 \|\dot{\gamma}(t)\|_{\mathbf{X}, \mu}^2 dt.$$

Zaproponowany w pracy algorytm planowania ścieżek oparty na metodzie pól potencjału wykorzystuje wprowadzoną miarę kosztu nieholonomicznego (obok odległości od przeszkód) do modyfikacji położenia konfiguracji bieżącej.

W artykule [37] przedstawiono zależności pomiędzy różnymi wskaźnikami złożoności ścieżek. Zakładając wolną przestrzeń wokół trajektorii  $\mathbf{q}(\cdot)$  o rozmiarze  $\epsilon$ , wskaźnik dla sterowań typu bang-bang ma postać:

$$c_{sw}(\mathbf{q}(\cdot), \epsilon) = \inf \{\text{liczba przełączeń sterowań}\}.$$

Dla sterowań dowolnych:

$$c_t(\mathbf{q}(\cdot), \epsilon) = \inf \{\text{liczba zmian znaku sterowań}\},$$

$$c_m(\mathbf{q}(\cdot), \epsilon) = \frac{1}{\epsilon} \inf \{\text{dlugosc}(\mathbf{q}(\cdot))\},$$

$$c_e(\mathbf{q}(\cdot), \varepsilon) = \inf \{ \text{liczba kół o promieniu } \varepsilon \text{ potrzebna do otoczenia ścieżki} \}.$$

Zdefiniowana została także  $\varepsilon$ -norma złożoności jako

$$\|\dot{\mathbf{q}}\|_{\varepsilon, \mathbf{X}} = \max_i \left\{ \frac{|\dot{q}_i^{\mathbf{X}}|}{\varepsilon^{\text{stopień}(\mathbf{X}_i)}} \right\}, \quad (4.116)$$

gdzie  $\dot{q}_i^{\mathbf{X}}$  oznacza  $i$ -tą współrzędną  $\dot{\mathbf{q}}$  wyrażoną w bazie pól wektorowych  $\mathbf{X}$ . Autor zilustrował sposób obliczania normy (4.116) na przykładzie kilku ścieżek samochodu z przyczepami.

## 5. Algorytmy

W celu rozwiązania zadania postawionego w rozdziale 3 zaproponowano następującą metodę postępowania:

- (1) przyjęcie początku układu globalnego w punkcie będącym rzutem prostopadłym punktu docelowego manipulatora mobilnego na płaszczyznę ruchu platformy. Orientacja układu globalnego dobieramy tak, aby jego oś  $z$  była normalna do płaszczyzny ruchu. W ten sposób uzyskujemy uproszczony opis ruchu platformy mobilnej, poruszającej się po płaszczyźnie z ustaloną (najczęściej równą 0) wartością współrzędnej  $z$ -owej;
- (2) wyznaczenie bezpiecznej konfiguracji transportowej manipulatora i jej osiągnięcie z konfiguracji początkowej;
- (3) wyznaczenie (pod-) zbioru pól docelowych platformy. Pojęcie docelowej platformy definiujemy jako takie usytuowanie platformy, które zapewnia bezkolizyjne osiągnięcie przez manipulator położenia docelowego manipulatora mobilnego bez konieczności rekonfigurowania platformy. Ponieważ wyznaczenie wszystkich pól docelowych platformy jest zadaniem złożonym obliczeniowo, należy wyznaczać jedynie mało liczny podzbiór pól docelowych charakteryzujący się dodatkowymi własnościami. Przykładowo, każda z pól docelowych jest taka, że manipulator osiąga punkt docelowy planowania ruchu na brzegu swojej przestrzeni roboczej. Tak określony podzbiór pól docelowych jest brzegiem obszaru wszystkich pól docelowych. Podobnie jak nadzbiór, zbiór brzegowych pól docelowych jest zwykle mocy continuum, więc przy ograniczonej możliwości obliczeń analitycznych, konieczne jest dalsze ograniczenie mocy zbioru. Dlatego w praktycznych przypadkach, w których złożoność obliczeniowa jest istotnym ograniczeniem, proponujemy, aby wyznaczane były tylko niektóre brzegowe pozy docelowe, lub podczas planowania ruchu platformy była wyznaczana jedna brzegowa poza docelowa znajdująca się, w pewnym sensie, najbliższej bieżącej pozy platformy;
- (4) wyznaczenie bezkolizyjnej ścieżki platformy od pozy początkowej do wybranej (ewentualnie dynamicznie zmiennej) brzegowej pozy docelowej;
- (5) wyznaczenie lokalnie optymalnych sterowań zapewniających maksymalizację odległości trajektorii od przeszkód,
- (6) optymalizacja długości realizowanej ścieżki — redukcja liczby planowań do minimum,
- (7) realizacja ścieżki,
- (8) w przypadku osiągnięcia konfiguracji umożliwiającej osiągnięcie celu zakończenie pracy, w przeciwnym przypadku wykonanie dodatkowego manewru zbliżającego stan końcowy do wnętrza obszaru dopuszczalnych konfiguracji platform.

W kolejnych podrozdziałach uszczegółowiono ogólny schemat metody planowania ruchu przedstawiony powyżej.

## 5.1. Konfiguracja transportowa manipulatora

Zadanie osiągnięcia transportowej konfiguracji manipulatora składa się z dwóch etapów: w pierwszym dla zadanej funkcji kryterialnej opisującej pożądane własności transportowe manipulatora wyznaczana jest konfiguracja optymalna, która z kolei, w etapie drugim, ma być osiągnięta z zadanej, początkowej konfiguracji manipulatora. Etap pierwszy determinuje jedynie optymalne położenie rzutu środka masy manipulatora na płaszczyźnie platformy, a konfiguracją optymalną manipulatora nazywamy jedną z realizacji tego położenia przez manipulator. Ponieważ położenie środka masy manipulatora w układzie lokalnym związanym z platformą jest funkcją jego konfiguracji, zatem rzut tego punktu na płaszczyznę platformy dziedziczy tę własność. Niech funkcją  $cm(\mathbf{x})$  opisuje ten rzut.

Matematycznie zadanie wyznaczenia optymalnej konfiguracji transportowej manipulatora jest następujące: wyznaczyć taką trajektorię  $\mathbf{x}(\cdot)$  manipulatora zainicjowaną w zadanej konfiguracji początkowej  $\mathbf{x}(0) = \mathbf{x}_0$ , by jej przebieg był bezkolizyjny z przeszkodami  $\forall t \in [0, T] \quad B(\mathbf{x}) \cap \mathbf{O} = \mathbf{O}$  oraz manipulator osiągnął optymalną wartość  $cm(\mathbf{x}(T)) = cm^*$ , gdzie  $T$  jest horyzontem czasowym planowania. Zadanie jest więc analogonem klasycznego problemu kinematyki odwrotnej (tutaj kinematyka prostą zastępuje położenie rzutu środka masy manipulatora) z ograniczeniami na bezkolizyjność ruchu. W literaturze robotycznej znajduje się wiele metod rozwiązania tego problemu [33, 62]. Większość z nich ograniczenie wynikające z istnienia przeszkód wprowadza jako karę do funkcji kryterialnej, której drugim elementem jest składnik zapewniający “przyciąganie” do punktu  $cm^*$ . W warstwie algorytmicznej może być wykorzystana klasyczna metoda Newtona.

Rozwiązanie zadania planowania jest zagwarantowana teoretycznie, gdy konfiguracja początkowa i konfiguracja optymalna leżą w jednej komponencie spójności przestrzeni bezkolizyjnej (przestrzeń konfiguracyjna manipulatora w wyniku istnienia przeszkód może być podzielona na wiele rozłącznych obszarów spójności). Warunek ten jest trudny do sprawdzenia, gdyż obszary zabronione w przestrzeni konfiguracyjnej nie są zadane analitycznie. Nie należy zatem wykluczyć, że zadanie powyżej postawione nie może być rozwiązane w pełni, co oznacza, że wygenerowano jedynie część trajektorii spełniającej warunek bezkolizyjności, lecz nie zakończonej w konfiguracji optymalnej. Heurystyczny i praktyczny sposób postępowania w tym przypadku polega na takim ruchu platformy, który spowoduje zwiększenie odległości minimalnej manipulatora od przeszkód, a następnie podjęciu kolejnej próby osiągnięcia konfiguracji optymalnej. Proces ruchu platformy i manipulatora może być powtarzany iteracyjnie.

Warto także zauważyć, że algorytm Newtona najczęściej wykorzystywany w planowaniu ruchu manipulatora jest typowym przedstawicielem rodziny algorytmów antygradientowych, które mogą utknąć w minimum lokalnym funkcji optymalizowanej. Dlatego warto rozważyć możliwość wprowadzenia albo pewnego elementu losowości po osiągnięciu w konfiguracji lokalnie optymalnej, bądź heurystycznych sposobów opuszczania minimów lokalnych. Przykład ostatnio wymienionego podejścia (dla robotów mobilnych) zawiera praca [23].

Układ podrozdziału jest następujący: w sekcji 5.1.1 przeanalizowano zasady wyliczania rozkładu sił nacisku wywieranych przez manipulator na punktu podparcia kół. Szczegółowe obliczenia wykonano dla platform trój- i cztero kołowych najczęściej występujących w praktyce. Z kolei przedstawiono, w sekcji 5.1.2, gdzie leży optymalne położenie rzutu punktu środka masy platformy na płaszczyznę nośną platformy mobilnej. Przeanalizowano także, jaki wpływ na optymalne przyłożenie siły wywieranej przez manipulator na platformę ma odchylenie płaszczyzny ruchu platformy od wektora prostopadłego do siły grawitacji oraz uwzględnienie zjawisk poślizgu i tarcia.

### 5.1.1. Przenoszenie sił nacisku na podłoże

Analizując kołowe manipulatory mobilnych należy wziąć pod uwagę stabilność platformy wraz z umieszczonym na niej manipulatorem. W skrajnych przypadkach możliwe stają się niepożądane



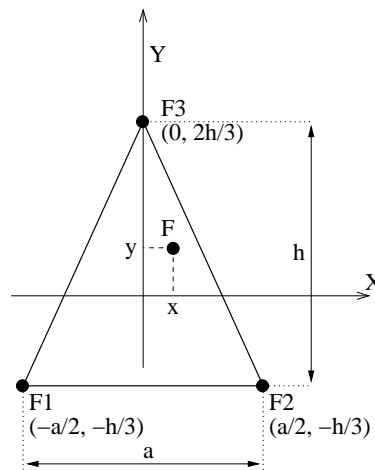
zachowania układu, takie jak: wywrócenie platformy czy utrata przyczepności kół. W celu wyeliminowania niekorzystnych zjawisk, należy wyliczyć rozkład sił w punktach podparcia (kołach). Analiza taka jest wystarczająca w warunkach statycznych, gdy platforma nie porusza się. Do analizy rozkładu sił wykorzystamy zasadę zachowania momentu sił oraz pierwszą zasadę dynamiki Newtona [68] po przyjęciu następujących założeń:

- platforma jest sztywna, co oznacza, że nie ulega deformacji pod wpływem nacisku;
- siły opisywane są względem punktów podparcia kół na płaszczyźnie  $XY$  stanowiącej płaszczyznę ruchu platformy;
- wektor grawitacji działa wzdłuż osi  $z$ ,
- siły reakcji działają punktowo, w miejscu styku koła z podłożem;
- wypadkowa siła nacisku manipulatora  $F$  jest przyłożona w punkcie o współrzędnych  $(x, y)$  względem środka masy platformy.

W dalszej części podrozdziału ograniczymy się do analizy dwóch, występujących najczęściej w praktyce, idealizowanych typach platform: o trzech i czterech kołach [49]. Inne rodzaje platform można rozważać podobnie.

### Platforma trójkątowa

Pierwszą konstrukcją jest platforma o rzucie w kształcie trójkąta równoramiennego o długości podstawy  $a$  i wysokości  $h$  (rys. 5.1). Z platformą wiążemy kartezjański układ współrzędnych  $OXY$ , którego początek umieszczony jest w środku masy trójkąta. Punkty podparcia (styczności kół z podłożem) mają współrzędne:  $K_1(-a/2, -h/3)$ ,  $K_2(a/2, -h/3)$  oraz  $K_3(0, 2h/3)$  i w tych punktach działają siły,  $F_1, F_2, F_3$ , odpowiednio. Do wyliczenia rozkładu sił oddziałujących na koła, wykorzystujemy pierwszą



Rysunek 5.1. Punkty podparcia i geometria trójkątowa

zasadę dynamiki Newtona [68] stanowiącą, że siły działające na ciało pozostające w spoczynku się równoważą. Zatem suma sił przenoszonych przez podpory równa jest sile nacisku  $F$ :

$$F_1 + F_2 + F_3 = F. \quad (5.1)$$

Zasada zachowania momentu sił orzeka, że suma momentów względem dowolnego punktu ciała sztywnego jest stała i wynosi 0,

$$\sum M_O = 0. \quad (5.2)$$

Korzystając z zależności (5.1) i (5.2) (względem punktu środka masy i sił rozłożonych wzdłuż ortogonalnych osi  $OX$  i  $OY$ ) uzyskujemy liniowe równanie w postaci:

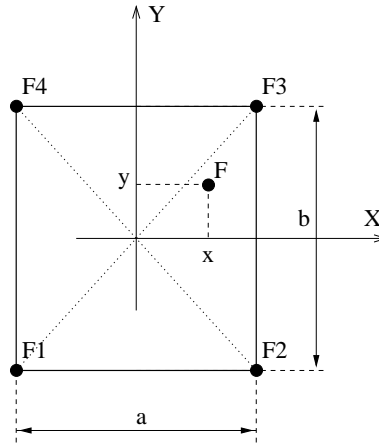
$$\begin{bmatrix} 1 & 1 & 1 \\ -a/2 & a/2 & 0 \\ -h/3 & -h/3 & 2h/3 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} F \\ x \cdot F \\ y \cdot F \end{bmatrix}. \quad (5.3)$$

Po przekształceniach (5.3), rozkład sił w punktach podparcia jest następujący:

$$\begin{cases} F_1 = F \left( \frac{1}{3} - \frac{x}{a} - \frac{y}{2h} \right) \\ F_2 = F \left( \frac{1}{3} + \frac{x}{a} - \frac{y}{2h} \right) \\ F_3 = F \left( \frac{1}{3} + \frac{y}{h} \right) \end{cases} \quad (5.4)$$

### Platforma czterokołowa

Kolejną opisywaną konstrukcją, jest platforma podparta w czterech punktach. Spośród wielu możliwych rozmieszczeń punktów podparcia, wybrana została konfiguracja najprostsza, a zarazem najczęściej spotykana: platforma prostokątna z kołami umieszczonymi w wierzchołkach. Osie układu współrzędnych są równoległe do boków prostokąta, a punkt początkowy układu leży na przecięciu przekątnych (w środku masy). Platforma jest symetryczna względem osi lokalnego układu współrzędnych. Punkty podparcia (kół) mają następujące współrzędne:  $K_1(-a/2, -b/2)$ ,  $K_2(a/2, -b/2)$ ,  $K_3(a/2, b/2)$ ,  $K_4(-a/2, b/2)$  (por. rys. 5.2) i w nich działają analogicznie indeksowane siły. Metoda



Rysunek 5.2. Platforma czterokołowa

wyznaczania rozkładu sił jest identyczna jak dla przypadku trójkątowca. Suma sił przenoszonych na koła jest równa sile nacisku:

$$F_1 + F_2 + F_3 + F_4 = F. \quad (5.5)$$

Uzupełniając zależność (5.5) o równania wynikające z (5.2) wyrażoną względem osi  $OX$  i  $OY$ , uzyskujemy zależność liniową:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ -a/2 & a/2 & a/2 & -a/2 \\ -b/2 & -b/2 & b/2 & b/2 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} = \begin{bmatrix} F \\ x \cdot F \\ y \cdot F \end{bmatrix}. \quad (5.6)$$

Układ (5.6) jest nadmiarowy (4 niewiadome, 3 równania), zatem zbiór rozwiązań jest rozmaitością jednowymiarową. Istotnie, przyjmując  $F_i = 0$  dla  $i = 1, \dots, 4$  mamy cztery rozwiązania (5.6):

$$\begin{aligned} &\{F_1 = 0, F_2 = F\left(\frac{1}{2} - \frac{y}{b}\right), F_3 = F\left(\frac{x}{a} + \frac{y}{b}\right), F_4 = F\left(\frac{1}{2} - \frac{x}{a}\right)\}, \\ &\{F_1 = F\left(\frac{1}{2} - \frac{y}{b}\right), F_2 = 0, F_3 = F\left(\frac{1}{2} + \frac{x}{a}\right), F_4 = F\left(-\frac{x}{a} + \frac{y}{b}\right)\}, \\ &\{F_1 = F\left(-\frac{x}{a} - \frac{y}{b}\right), F_2 = F\left(\frac{1}{2} + \frac{x}{a}\right), F_3 = 0, F_4 = F\left(\frac{1}{2} + \frac{y}{b}\right)\}, \\ &\{F_1 = F\left(\frac{1}{2} - \frac{x}{a}\right), F_2 = F\left(\frac{x}{a} + \frac{y}{b}\right), F_3 = F\left(\frac{1}{2} - \frac{y}{b}\right), F_4 = 0\}, \end{aligned}$$

z których każde dwa są liniowo niezależne, a trzy - zależne. Zbiór rozwiązań dopuszczalnych opiszemy kombinacją liniową dwóch dowolnych rozwiązań częściowych  $F^{(j)} = \{F_1^j, F_2^j, F_3^j, F_4^j\}$ ,  $j = 1, 2$

$$F^K(\alpha) = \alpha F^{(1)} + (1 - \alpha) F^{(2)} = (F_1^K(\alpha), F_2^K(\alpha), F_3^K(\alpha), F_4^K(\alpha)), \quad (5.7)$$

gdzie  $\alpha$  jest parametrem rzeczywistym. Eliminacja parametru  $\alpha$  (wybór jednego z rozwiązań) jest możliwa po przyjęciu wartościującej funkcji kryterialnej  $Q(F^K(\alpha))$  i rozwiązaniu zadania optymalizacyjnego:

$$\alpha^* = \arg \min_{\alpha} Q(F^K(\alpha)). \quad (5.8)$$

Funkcje kryterialne  $Q(F^K(\alpha))$  uzasadnione praktycznie to:

—  $Q_1(F^K(\alpha)) = (F_1^K(\alpha) + F_3^K(\alpha) - F_2^K(\alpha) - F_4^K(\alpha))^2$ , wynikająca z założenia, że przemieszczenie  $\Delta$  środka platformy przekłada się na przemieszczenia podpór  $\Delta_1$  do  $\Delta_4$  (wirtualnie umieszczonych w punktach  $K_1, \dots, K_4$ ), tak że przemieszczenie  $\Delta$  jest średnią arytmetyczną przemieszczeń na każdej przekątnej:

$$\Delta = \frac{\Delta_1 + \Delta_3}{2} = \frac{\Delta_2 + \Delta_4}{2}. \quad (5.9)$$

Zakładając, że podpory oparte są na jednakowych sprężynach, wartość siły reakcji (dla małych przemieszczeń) jest proporcjonalna do amplitudy przemieszczeń [68]

$$F_i = k \cdot \Delta_i. \quad (5.10)$$

ze współczynnikiem proporcjonalności  $k$ . Z równań (5.9) i (5.10) wynika, że

$$F_1 + F_3 = F_2 + F_4. \quad (5.11)$$

—  $Q_2(F^K(\alpha)) = (F_1^K(\alpha) + F_4^K(\alpha) - F_2^K(\alpha) - F_3^K(\alpha))^2$ , preferująca rozkłady nacisków o równych wartościach po lewej i prawej stronie platformy,

—  $Q_3(F^K(\alpha)) = (F_1^K(\alpha) - F_2^K(\alpha))^2 + (F_3^K(\alpha) - F_4^K(\alpha))^2$ , wymuszająca równe naciski na osiach przednich i tylnych.

Po podstawieniu funkcji kryterialnej  $Q_2(F^K(\alpha))$  do równania (5.7), parametr  $\alpha$  się upraszcza. Dla kryteriów  $Q_1(F^K(\alpha))$  i  $Q_3(F^K(\alpha))$  oraz dowolnego doboru funkcji  $F^{(1)}$  i  $F^{(2)}$  rozwiązanie optymalne (po podstawieniu  $\alpha^*$  do równania (5.7)) przedstawia się następująco:

$$\begin{cases} F_1 = F\left(\frac{1}{4} - \frac{x}{2a} - \frac{y}{2b}\right) \\ F_2 = F\left(\frac{1}{4} + \frac{x}{2a} - \frac{y}{2b}\right) \\ F_3 = F\left(\frac{1}{4} + \frac{x}{2a} + \frac{y}{2b}\right) \\ F_4 = F\left(\frac{1}{4} - \frac{x}{2a} + \frac{y}{2b}\right). \end{cases} \quad (5.12)$$

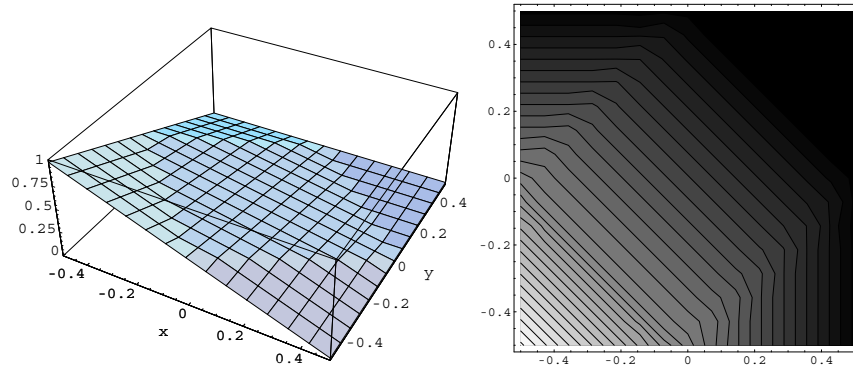
Rozkład sił dany równaniem (5.12) nie opisuje poprawnie rzeczywistości w przypadku, gdy jedna z sił staje się ujemna (koło traci kontakt z podłożem). W tej sytuacji platforma staje się układem o trzech punktach podparcia. W przypadku utraty kolejnego punktu styczności z podłożem platforma straci stabilność — ten wariant nie będzie dalej rozpatrywany.

Rozkład sił w modzie trójpodporowym, obliczony analogicznie jak dla platformy trójkątnej, przedstawia się następująco (w  $i$ -tej kolumnie wyliczono siły przy brak kontaktu koła  $i$ -tego z podłożem):

	1	2	3	4
$F_1$	0	$F\left(\frac{1}{2} - \frac{y}{b}\right)$	$F\left(-\frac{x}{a} - \frac{y}{b}\right)$	$F\left(\frac{1}{2} - \frac{x}{a}\right)$
$F_2$	$F\left(\frac{1}{2} - \frac{y}{b}\right)$	0	$F\left(\frac{1}{2} + \frac{x}{a}\right)$	$F\left(\frac{x}{a} - \frac{y}{b}\right)$
$F_3$	$F\left(\frac{x}{a} + \frac{y}{b}\right)$	$F\left(\frac{1}{2} + \frac{x}{a}\right)$	0	$F\left(\frac{1}{2} + \frac{y}{b}\right)$
$F_4$	$F\left(\frac{1}{2} - \frac{x}{a}\right)$	$F\left(-\frac{x}{a} + \frac{y}{b}\right)$	$F\left(\frac{1}{2} + \frac{y}{b}\right)$	0

(5.13)

Model opisany równaniami (5.12) i (5.13) wydaje się dobrze odpowiadać rzeczywistości. Przykład wartości siły  $F_1$  dla różnego położenia punktu  $F$  przedstawia rys. 5.3. Wykres składa się z pięciu wycinków płaszczyzn, z czego środkowa odpowiada modowi czteropodparciowemu, a narożne - wariantom trójpodporowym. Przejście między modem 4- i 3-podporowym jest klasy  $C^0$  (tylko ciągłe).



Rysunek 5.3. Wartość siły  $F_1$  w zależności od położenia punktu  $F$

### 5.1.2. Optymalna lokalizacja siły nacisku

W poprzednim podrozdziale miejsce działania siły wywieranej przez manipulator było ustalone, parametr  $(x, y)$ . Obecnie zoptymalizujemy jego położenie. Dysponując modelem przenoszenia sił nacisku wywieranych przez manipulator na punkty styczności kół z podłożem, opisanym w rozdziale 5.1.1, można wyznaczyć mapy zalecanych rzutów  $(x, y)$  skumulowanej siły wywieranej przez manipulator. Mapy służą do wyznaczenia konfiguracji transportowej manipulatora. Zbadano następujące funkcje kryterialne:

- (1) preferującą jednakowe sumy sił po przekątnych

$$W_1(x, y) = [F_1(x, y) + F_3(x, y) - (F_2(x, y) + F_4(x, y))]^2, \quad (5.14)$$

- (2) równego sumarycznego nacisku lewej i prawej strony platformy

$$W_2(x, y) = [F_1(x, y) + F_4(x, y) - (F_2(x, y) + F_3(x, y))]^2, \quad (5.15)$$

(3) równego nacisku osi przedniej i tylnej

$$W_3(x, y) = (F_1(x, y) - F_2(x, y))^2 + (F_3(x, y) - F_4(x, y))^2, \quad (5.16)$$

(4) maksymalnego, równego nacisku na oś tylną (zakładając, że jest to oś napędowa) przy zapewnieniu kołom przednim gwarantowanego minimum obciążenia  $p \cdot F$ ,  $p \in [0, 1]$

$$\begin{aligned} l_1(x, y) &= (F_1(x, y) - F_2(x, y))^2 \\ l_2(x, y) &= F_3(x, y) + F_4(x, y) \\ W_4(x, y, p) &= l_1(x, y) + l_2(x, y) + \begin{cases} 0 & l_2(x, y) \geq pF \\ \gamma(pF - l_2(x, y)) & \text{gdy przeciwnie} \end{cases} \end{aligned} \quad (5.17)$$

gdzie  $\gamma > 0$  jest współczynnikiem wagowym. Pierwszy składnik ( $l_1(x, y)$ ) kryterium  $W_4$  osiąga wartości minimalne dla  $x = 0$ , czyli dla siły  $F$  działającej w osi  $OY$  platformy. Po podstawieniu do  $l_2(x, y)$  danych ze wzorów (5.12) oraz (5.13) i uproszczeniu otrzymujemy  $l_2(x, y) = (y/b + 1/2)F$ . Do rozstrzygnięcia pozostaje jedynie wpływ części związanej z karą za zmniejszenie nacisku na oś przednią. W tym przypadku rozwiązanie składa się z dwóch części. Gdy  $l_2(x, y) \geq pF$ , funkcja kryterialna zadana jest wyrażeniem

$$W_4^I(x, y, p) = l_1(x, y) + (y/b + 1/2).$$

$l_1(x, y)$  osiąga minimum dla  $x = 0$ , natomiast drugi składnik dla wartości  $y$  położonych jak najbliżej osi napędzanej. Po uwzględnieniu warunku  $l_2(x, y) \geq pF$  otrzymujemy

$$y_*^I(p) = b(p - 1/2). \quad (5.18)$$

W drugim przypadku, gdy  $l_2(x, y) < pF$ , funkcja kryterialna po uproszczeniu przyjmuje postać:

$$W_4^{II}(x, y, p) = l_1(x, y) + [\gamma p - (\gamma - 1)(y/b + 1/2)]F. \quad (5.19)$$

Funkcja (5.19) osiąga minimum dla dużych wartości  $y$ . Uwzględniając warunek  $l_2(x, y) < pF$ , otrzymujemy identyczną z (5.18) postać optymalnego położenia  $y$ , tym razem ważną dla całego rozpatrywanego przedziału:

$$y_*(p) = b(p - 1/2). \quad (5.20)$$

Wykresy funkcji  $W_1$ – $W_4$  (dla kryterium czwartego z  $p = \frac{1}{5}$ ,  $\gamma = 100$ ) przedstawiono na rys. 5.4.

### 5.1.3. Wpływ pochylenia platformy na optymalne położenie siły

Rozważania rozdziałów 5.1.1 i 5.1.2 zakładały, że platforma spoczywa na płaszczyźnie prostopadłej do wektora grawitacji. W tym przypadku wysokość punktu przyłożenia siły nad platformą (mierzona współrzędną  $z$ ) nie odgrywała żadnej roli. Osłabimy to wymaganie i rozważymy przypadek, w którym płaszczyzna platformy jest odchylona o kąt  $\alpha$  względem płaszczyzny normalnej do wektora grawitacji, a siła  $F$  jest przyłożona w punkcie  $(x, 0, z)$  (rys. 5.5). Zmianie ulega wartość momentu powodowanego siłą  $F$ , który wynosi:

$$M_F = rF \sin \gamma = rF \sin(\beta - \alpha) = rF (\sin \beta \cos \alpha - \cos \beta \sin \alpha). \quad (5.21)$$

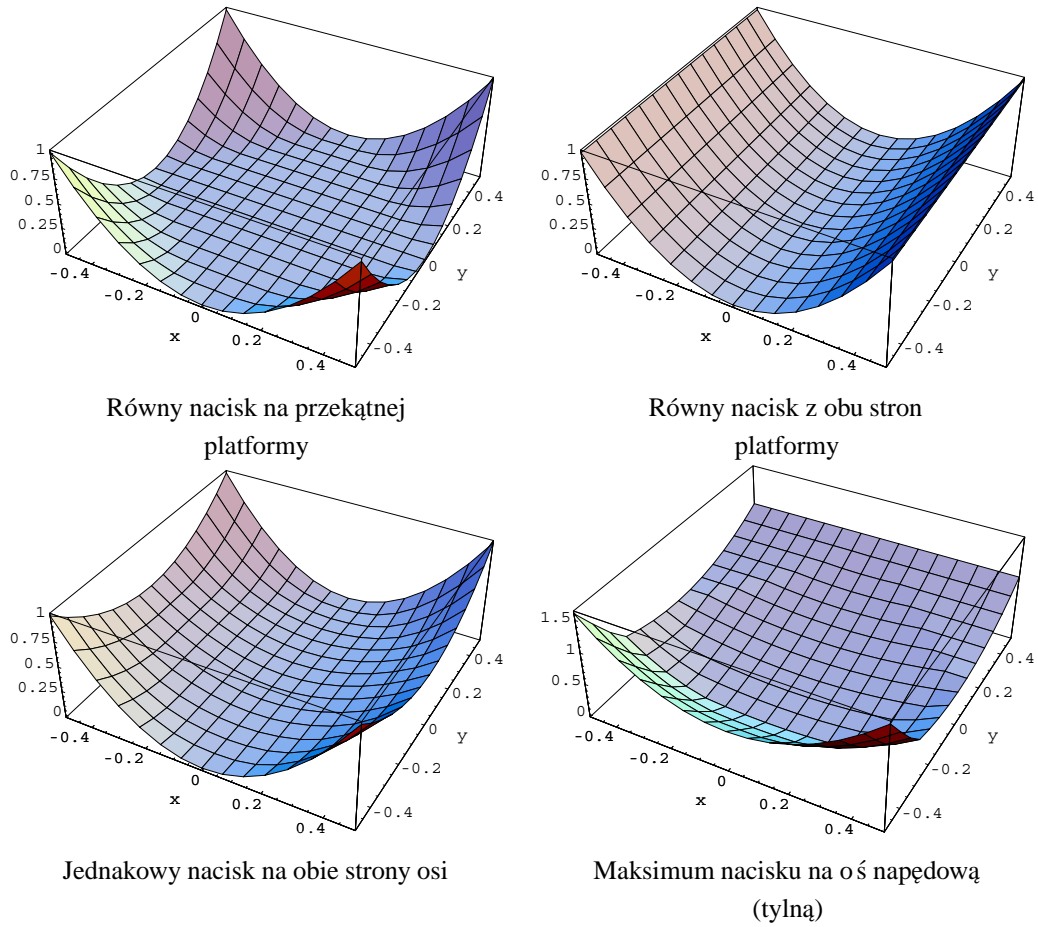
Z zależności geometrycznych wynika, że  $\sin \beta = x/r$  oraz  $\cos \beta = z/r$ , a uwzględniając równanie (5.21) otrzymujemy:

$$M_F = rF \left( \frac{x}{r} \cos \alpha - \frac{z}{r} \sin \alpha \right) = F (x \cos \alpha - z \sin \alpha). \quad (5.22)$$

Powyższe równanie reprezentuje rzut punktu  $(x, 0, z)$  na płaszczyznę prostopadłą do siły grawitacji. Dla przypadku  $\alpha = 0$  równanie (5.22) upraszcza się do postaci

$$M_F = xF,$$

wykorzystanej uprzednio.



Rysunek 5.4. Wykresy funkcji kryterialnych

#### 5.1.4. Wpływ tarcia i zjawisko poślizgu na optymalne położenie siły

W przypadku odchylenia platformy od poziomu, należy uwzględnić także występowanie sił tarcia, rys. 5.6. Warunek braku poślizgu jest następujący: siły usiłujące przemieścić ciało — siła przesuwająca  $F_{pi} = F_i \sin \alpha$  i siła wzajemnych oddziaływań  $F_w$  — muszą być co najmniej równoważone przez siłę tarcia  $F_{ti} = F_i \mu_i \cos \alpha$ , gdzie  $\mu_i$  jest współczynnikiem tarcia  $i$ -tego ciała o podłoże:

$$\begin{cases} F_1 \sin \alpha - F_w \leq F_1 \mu_1 \cos \alpha \\ F_2 \sin \alpha + F_w \leq F_2 \mu_2 \cos \alpha. \end{cases} \quad (5.23)$$

Z układu (5.23) otrzymujemy warunek utrzymania platformy w spoczynku:

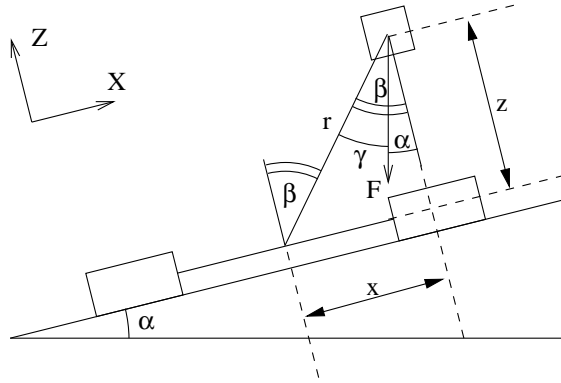
$$(F_1 + F_2) \sin \alpha \leq (F_1 \mu_1 + F_2 \mu_2) \cos \alpha. \quad (5.24)$$

Dla platformy podpartej w większej ilości punktów ( $N$ ), uogólnienie równania (5.24) jest natychmiastowe:

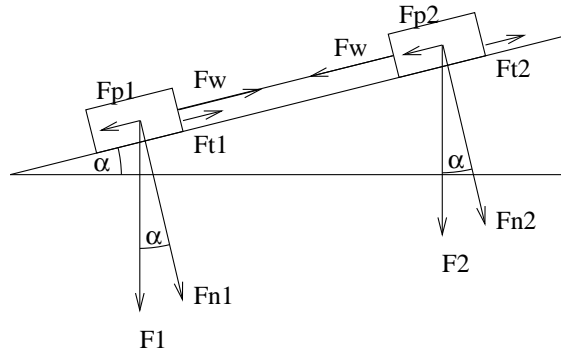
$$\sum_{i=1}^N F_i \sin \alpha \leq \sum_{i=1}^N F_i \mu_i \cos \alpha, \quad (5.25)$$

a maksymalny dopuszczalny kąt odchylenia platformy od poziomu opisuje wzór:

$$\alpha \leq \arctg \sum_{i=1}^N \frac{F_i}{F} \mu_i, \quad \text{gdzie} \quad F = \sum_{i=1}^N F_i. \quad (5.26)$$



Rysunek 5.5. Odchylenie platformy od płaszczyzny normalnej do wektora grawitacji



Rysunek 5.6. Siły tarcia

Gdy podłoże jest jednorodne, wtedy współczynniki tarcia w poszczególnych punktach podparcia platformy przyjmują jednakowe wartości  $\mu = \mu_1 = \mu_2 = \dots = \mu_N$ , a równanie (5.26) upraszcza się do postaci:

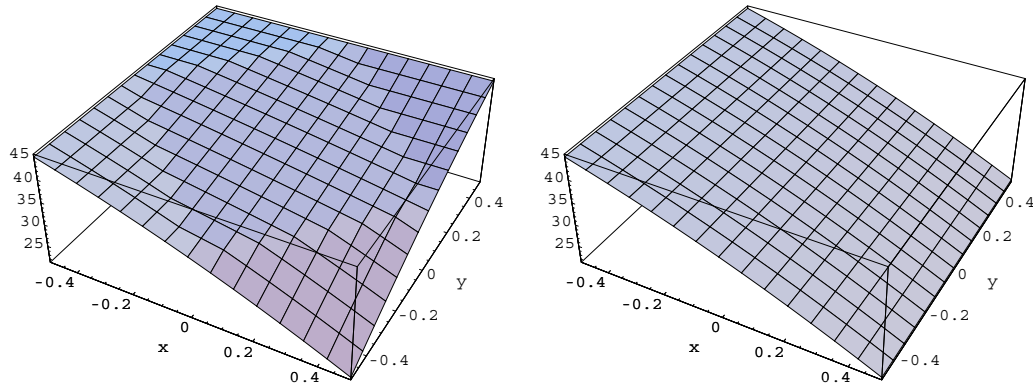
$$\alpha \leq \arctg \mu. \quad (5.27)$$

Przeprowadzono symulacje zależności kąta  $\alpha$  od punktu przyłożenia siły, dla różnych współczynników tarcia. Na rys. 5.7 przedstawiono sytuację, gdy na grunt piaszczysty ( $\mu = 0,4$ ) trafiło prawe dolne koło oraz oba prawe koła (pozostałe koła stoją na betonie,  $\mu = 1$ ). Na podstawie wykresów można zauważyć, że w przypadku jazdy po niejednorodnym terenie (zmieniające się  $\mu$ ) najbezpieczniejszą strategią jest utrzymywanie środka masy manipulatora nad środkiem masy platformy, gdyż wtedy wahania współczynnika tarcia poszczególnych kół są najbardziej tłumione.

### 5.1.5. Położenie środka masy manipulatora

Dla stateczności manipulatora mobilnego pożądane jest, aby rzut środka masy manipulatora opisany współrzędnymi  $(x, y)$  znajdował się możliwie najbliżej środka masy platformy oraz by współrzędna  $z$ -owa była możliwie niewielka. Zakładając że manipulator zbudowany jest z  $k$ -ramion o znanych masach  $m_i$ ,  $i = 1, \dots, k$ , położenie środka masy manipulatora względem układu platformy zadane jest równaniem:

$$\begin{bmatrix} x_{cm} \\ y_{cm} \\ z_{cm} \end{bmatrix} = \frac{1}{\sum_{i=1}^k m_i} \begin{bmatrix} \sum_{i=1}^k m_i (\beta_i x_i(\mathbf{x}) + (1 - \beta_i) x_{i-1}(\mathbf{x})) \\ \sum_{i=1}^k m_i (\beta_i y_i(\mathbf{x}) + (1 - \beta_i) y_{i-1}(\mathbf{x})) \\ \sum_{i=1}^k m_i (\beta_i z_i(\mathbf{x}) + (1 - \beta_i) z_{i-1}(\mathbf{x})) \end{bmatrix} \quad (5.28)$$



Prawe dolne koło stoi na piasku

Oba prawe koła stoją na piasku

Rysunek 5.7. Maksymalny kąt pochylenia platformy

gdzie  $x_{cm}, y_{cm}, z_{cm}$  są współrzędnymi środka masy manipulatora wyrażonymi w układzie odniesienia platformy,  $(x_i(\mathbf{x}), y_i(\mathbf{x}), z_i(\mathbf{x}))$ ,  $i = 1, \dots, k$  są współrzędnymi końców ogniw manipulatora w układzie platformy,  $\beta_i$ ,  $i = 1, \dots, k$  są współczynnikami określającym rozkład masy ( $\beta_i = 0$  gdy masa jest na początku ramienia,  $\beta = 1$  gdy na końcu). Jeśli manipulator przewozi ładunek, należy masę ładunku  $m_l$  uwzględnić jako skupioną na końcu efektora.

## 5.2. Zbiór póż docelowych platformy

W klasycznym zadaniu planowania trajektorii platform nieholonomicznych należy osiągnąć jedną, wyróżnioną pozę zadaną [33]. Jeżeli podczas wyznaczania ruchu do pozy docelowej (na etapie planowania), lub realizacji wyznaczonego scenariusza ruchu (na etapie sterowania) wystąpi uchyb w jej osiągnięciu (np. wskutek niedokładności numerycznych bądź zbyt dużych kroków czasowych) – osiągnięta jest poza bliska pozy zadanej [17]. W planowaniu ruchu platformy (nie wspomaganej przez manipulator) konieczne jest wtedy dosterowywanie do pozy docelowej lub replanowanie toru ruchu. Gdy zbiór póż docelowych jest bardziej liczny niż jednoelementowy, osiągnięcie pozy bliskiej zadanej może być wystarczające z punktu widzenia realizacji zadania przez manipulator mobilny. Możliwości dostosowawcze manipulatora, ze względu na jego holonomiczność w typowym przypadku, są znacznie większe niż możliwości ruchowe nieholonomicznej platformy. Zatem pożądanym jest określenie zbioru póż docelowych platformy  $\mathbb{Q}_D \subset \mathbb{Q}$  gromadzącego pozy, z których możliwe jest bezkolizyjne osiągnięcie zadanego punktu docelowego  $\mathbf{y}_d \in \mathbb{Y}$  w przestrzeni zadaniowej przy pomocy jedynie manipulatora:

$$\mathbb{Q}_D = \{\mathbf{q} \in \mathbb{Q} : \exists \mathbf{x}(\cdot), \mathbf{x}(0) = \mathbf{x}_m, \mathbf{k}(\mathbf{q}, \mathbf{x}(T)) = \mathbf{y}_d, \forall t \in [0, T] (\mathbf{x}(t), \mathbf{q}) \text{ bezkolizyjna}\}. \quad (5.29)$$

Dalsze rozważania służyć będą określeniu wybranych póż i zostaną przeprowadzone na typowej platformie poruszającej się po płaszczyźnie opisanej przy pomocy dwóch współrzędnych położenia  $x, y$  i jednej orientacji  $\phi$ . Sposób postępowania przebiega w kilku etapach:

- (1) wybrać rodzinę punktów startowych platformy i przy pomocy klasycznego algorytmu Newtona dla kinematyki  $\mathbf{k}(x, y, \phi, \mathbf{x}) = \mathbf{y}$ , z zadanymi: położeniem końcowym  $\mathbf{y}_d$  i nieruchomym manipulatorem w konfiguracji transportowej  $\mathbf{x}_m$  wyznaczyć  $(x^*, y^*, \phi^*)$  spełniające równanie kinematyki i zapewniające osiągnięcie punktu zadanego,
- (2) wyznaczyć punkty należące do brzegu obszaru póż dopuszczalnych korzystając z póż otrzymanych w etapie pierwszym jako inicjujących algorytm Newtona.



W pierwszym etapie, dla współrzędnych położeniowych pozy początkowej platformy  $\mathbf{P}(x_p, y_p)$  wyznaczane są odpowiadające im współrzędne biegunowe:

$$\varphi_p = \arctan(y_p, x_p), \quad r_p = \sqrt{x_p^2 + y_p^2}, \quad (5.30)$$

gdzie  $\varphi_p$  określa azymut pozy bieżącej względem początku globalnego układu współrzędnych. Wykorzystanie współrzędnych biegunowych pozwala na łatwe określenie grupy punktów startowych  $\mathbf{P}_i = (\varphi_{p_i}, r_{p_i})$ :

$$\varphi_{p_i} = \varphi_p + \Delta\varphi_i, \quad \Delta\varphi_i \in [-\pi/2, \pi/2], \quad r_{p_i} = r_i \quad (5.31)$$

znajdujących się na łuku okręgu scentrowanego w początku układu globalnego. Z każdego z tych punktów inicjowana jest klasyczna (a zatem najmniej złożona obliczeniowo) wersja algorytmu Newtona, zob. (A.1) (ewentualnie z karą za zbliżanie do przeszkód w przestrzeni kolizyjnej). Uzyskane rozwiązania niekoniecznie muszą wyznaczać brzeg obszaru pól dopuszczalnych. Dlatego stosujemy, w drugim etapie, algorytmem zapewniającym optymalizację (oprócz kryterium podstawowego — (5.29)) także dodatkowego czyniącego atrakcyjnym zbiór brzegowy  $\mathbb{Q}_D$ .

W tym celu zastosowano warianty algorytmu Newtona w dwóch wersjach: z minimalizacją w przestrzeni zerowej (A.2), (A.3) i minimalizującą funkcje jakości (A.4) (istnieją również mniej złożone, od algorytmu Newtona, sposoby obliczeniowe [51]). Każdy z tych algorytmów pozostawia manipulator nieruchomy i utrzymuje zadany punkt docelowy, lecz zmienia pozy platformy by osiągnąć brzeg obszaru dopuszczalnego. Można zaproponować kilka wariantów funkcji dodatkowych:

— minimalizacja odległości położenia bieżącego od punktu startowego  $\varphi_{p_i}, r_{p_i}$ :

$$d_1(\mathbf{q}(\varphi, r)) = \|(\varphi, r) - (\varphi_{p_i}, r_{p_i})\|,$$

— maksymalizacja odległość od początku układu globalnego,

$$d_2(\mathbf{q}(x = q_1, y = q_2)) = -\sqrt{q_1^2 + q_2^2},$$

— minimalizacja odległość od początku układu globalnego,

$$d_3(\mathbf{q}(x = q_1, y = q_2)) = \sqrt{q_1^2 + q_2^2},$$

— wzdłuż wytyczonego, na podstawie pozy początkowej, kierunku  $\varphi_p = \arctan(y_p, x_p)$  maksymalizując odległość od początku układu globalnego,

$$d_4(\mathbf{q}(x = q_1, y = q_2), \varphi) = -\sqrt{q_1^2 + q_2^2} + [(\varphi - \varphi_p) \bmod 2\pi]^2,$$

gdzie  $\alpha \bmod 2\pi \in [-\pi, \pi]$ ,

— preferowanie punktów wzdłuż kierunku  $\varphi_p$  minimalizując odległość od początku układu globalnego,

$$d_5(\mathbf{q}(x = q_1, y = q_2), \varphi) = \sqrt{q_1^2 + q_2^2} + [(\varphi - \varphi_p) \bmod 2\pi]^2.$$

Funkcje maksymalizujące odległość od początku układu globalnego,  $d_2(\mathbf{q}), d_4(\mathbf{q})$  pozwalają wyznaczyć zewnętrzny brzeg obszaru dopuszczalnego, natomiast funkcje minimalizujące  $d_3(\mathbf{q}), d_5(\mathbf{q})$  — wewnętrzną (jeśli obszar dopuszczalny jest spójny). Ponieważ niektóre funkcje jakości zawierają składniki o różnych jednostkach i zakresach, w praktycznych przypadkach wskazane jest ważenie odpowiednimi współczynnikami liczbowymi wpływu poszczególnych składowych.

Na podstawie wyznaczonych punktów składających się na brzeg obszaru dopuszczalnego póz platformy można określić, czy zaburzenia współrzędnych orientacji uniemożliwiają osiągnięcie punktu docelowego  $\mathbf{y}_d$ . Utrzymując ustalony stan platformy, z jedną ze współrzędnych zmienioną względem pierwotnie wyznaczonej wartości, rozwiązuje się zadanie (5.29) dopuszczając jedynie zmiany konfiguracji manipulatora  $\mathbf{x}$ . Dzięki temu można zwiększyć różnorodność konfiguracji wyznaczonych pierwotnym algorytmem. Kryteria stopu mogą być dwojakie: osiągnięcie rozwiązania lub przekroczenie założonej liczby iteracji. Zakładając niewielkie zmiany we współrzędnych można oszacować maksymalną liczbę iteracji potrzebnych do osiągnięcia rozwiązania.

### 5.2.1. Wybór bieżącej pozy docelowej platformy

Założmy, że zbiór póz docelowych  $\mathbb{Q}_D$  (lub jego podzbiór  $\mathbb{Q}'_D$ ) został wyznaczony. Algorytm planowania ruchu platformy wymaga określenia strategii wyboru bieżącej pozy docelowej, która stanowi aktualny cel ruchu platformy. Z obliczeniowego punktu widzenia wykluczone jest planowanie wielu ścieżek od pozy początkowej do tych należących do (pod-) zbioru  $\mathbb{Q}'_D \subset \mathbb{Q}_D$  i wybór (korzystając z miary oceny ścieżek nieholonomicznych – i dodatkowo bezkolizyjnych) najlepszej z nich do realizacji. Efektywniejszym obliczeniowo sposobem określenia pozy docelowej  $\mathbf{q}_f$  jest szybka selekcja jednego punktu ze zbioru  $\mathbb{Q}'_D$  na podstawie bieżącej pozy platformy (inicjującą jej wartością jest poza początkowa). Heurystycznie uzasadnionym wyborem jest przyjęcie za  $\mathbf{q}_f$ :

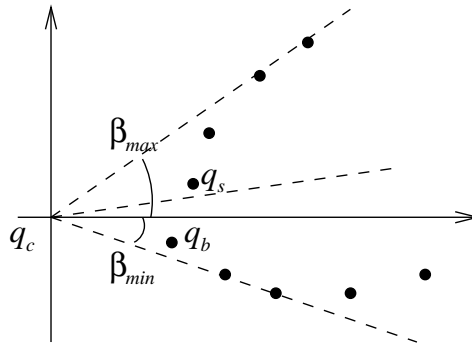
— najbliższy geometrycznie punkt zbioru  $\mathbb{Q}_D$ :

$$\mathbf{q}_b = \arg \min_{\mathbf{q} \in \mathbb{Q}'_D} \|\mathbf{q} - \mathbf{q}_c\|,$$

— punkt położeniowo „centralny” zbioru docelowego jest wyznaczony w trzech krokach:

- (0) rozpatruje się wyłącznie składowe położenia punktów  $\mathbf{q} \in \mathbb{Q}'_D$ ,
- (1) wyznaczyć zbiór punktów  $\tilde{\mathbb{Q}}_D \subset \mathbb{Q}'_D$  leżących w półpłaszczyźnie zwróconej ku punktowi  $\mathbf{q}_c$ ,  
 $w = \#\tilde{\mathbb{Q}}_D$ ,
- (2)  $\forall i \in \{1, \dots, w\}$  wyznaczyć kąt  $\beta_i$ , pod jakim jest „widoczny” punkt  $\mathbf{q}_i \in \tilde{\mathbb{Q}}_D$  z punktu startowego  $\mathbf{q}_c$  (kąt jaki tworzy wektor  $\mathbf{q}_i - \mathbf{q}_c$  względem układu odniesienia),
- (3) wybierz punkt  $\mathbf{q}_s = \mathbf{q}_i : \arg \min_{i \in \{1, \dots, w\}} \left| \beta_i - \frac{\max_j \beta_j + \min_j \beta_j}{2} \right|$ .

Oba sposoby wyboru zilustrowano na rysunku 5.8.



Rysunek 5.8. Idee metod wyboru pozy docelowej  $\mathbf{q}_f$  ze zbioru póz docelowych  $\mathbb{Q}'_D$

### 5.3. Wyznaczenie lokalnie optymalnej ścieżki platformy

Po wyznaczeniu zbioru konfiguracji docelowych platformy (por. podrozdział 5.2) można przystąpić do rozwiązania zadania wyznaczenia ścieżki łączącej bieżącą konfigurację platformy z jednym z punktów należących do zbioru jej konfiguracji docelowych. Ten etap planowania składa się z dwóch części [52]: planowania holonomicznej ścieżki bezkolizyjnej oraz jej poprawy z wykorzystaniem miary uwzględniającej nieholonomiczność układu sterowanego, ciągle zachowując bezkolizyjność ruchu.

#### 5.3.1. Planowanie ścieżki holonomicznej

W literaturze robotycznej znanych jest wiele metod planowania. Niektóre z nich przedstawiono w rozdziale 4. Do planowania ścieżki holonomicznej wykorzystamy metodę elastycznej wstęgi, która posiada wiele zalet. Jest bowiem ogólna, tolerancyjna na złożony kształt i nieregularność przeszkód, zapewnia możliwość ważenie bezpieczeństwa ruchu i długości ścieżki wynikowej, zapewnia lokalną optymalność ścieżki względem przyjętego kryterium jakości. Jest meta-metodą, ponieważ dopuszcza kreatywność badacza w jej ukonkretnieniu dla dedykowanego problemu. Daną wejściową dla metody jest jakakolwiek ścieżka bezkolizyjna (wyznaczona np. metodą grafu widoczności lub inną). Algorytm elastycznej wstęgi kodujemy w następujących krokach (uwaga: obliczenia są wykonywane tylko na współrzędnych położeniowych wektora  $\mathbf{q}$ ):

**Krok 1** Pobierz bezkolizyjną ścieżkę początkową łączącą stan początkowy  $\mathbf{q}_t$  z końcowym  $\mathbf{q}_f$ .

**Krok 2** Zdyskretyzuj ścieżkę przyjmując punkty pośrednie  $\mathbf{q}_i$ ,  $i = 0, 1, \dots, M$ .  $\mathbf{q}_0 = \mathbf{q}_t$ ,  $\mathbf{q}_M = \mathbf{q}_f$ .

**Krok 3** Wyznacz kolejno zmodyfikowane punkty pośrednie ścieżki

$$\mathbf{q}_i^+ \leftarrow \mathbf{q}_i + \xi_e \cdot \mathbf{F}_{el}(\mathbf{q}_i) + \xi_o \cdot \mathbf{F}_{odp}(\mathbf{q}_i), \quad i = 1, \dots, M-1, \quad (5.32)$$

gdzie  $\xi_e$  jest dodatnim współczynnikiem wagowym odpowiadającym sile elastyczności, której celem jest skracanie ścieżki, a  $\xi_o$  współczynnikiem siły odpychającej od przeszkód i zapewniającej bezpieczeństwo projektowanej ścieżce. Siłę elastyczności wyliczamy z zależności

$$\mathbf{F}_{el}(\mathbf{q}_i) = \frac{\mathbf{q}_{i-1} - \mathbf{q}_i}{\|\mathbf{q}_{i-1} - \mathbf{q}_i\|} + \frac{\mathbf{q}_{i+1} - \mathbf{q}_i}{\|\mathbf{q}_{i+1} - \mathbf{q}_i\|}, \quad (5.33)$$

która przyjmuje wartość minimalną równą  $\mathbf{0}$ , gdy punkty  $\mathbf{q}_{i-1}, \mathbf{q}_i, \mathbf{q}_{i+1}$  są współliniowe. Siłę odpychającą zadana jest wzorem

$$\mathbf{F}_{odp}(\mathbf{q}_i) = \sum_{j=1}^{\#\mathbf{O}} \mathbf{f}_{odp}(\mathbf{q}_i, \mathbf{o}_j), \quad \mathbf{O} = \cup_j \mathbf{o}_j, \quad (5.34)$$

gdzie  $\mathbf{f}_{odp}(\mathbf{q}_i, \mathbf{o}_j)$  jest wypadkową siłą odpychania punktu ścieżki  $\mathbf{q}_i$  od przeszkody  $\mathbf{o}_j \in \mathbf{O}$ . Siła odpychająca od przeszkody zdefiniowana jest najczęściej jako wypadkowa sił oddziaływania na punkt  $\mathbf{q}_i$  pochodząca od wybranych punktów z brzegu przeszkody. Często stosowana jest tutaj metafora jednoimiennych ładunków elektrycznych umieszczonych w  $\mathbf{q}_i$  i na brzegu przeszkody. Wtedy siłę wypadkową wyliczamy zgodnie z przepisem:

- (1) wybierz zbiór punktów  $\mathbf{o}_{j,k}$ ,  $k = 1, \dots, \#\mathbf{o}_j$  należących do przeszkody  $\mathbf{o}_j$ ,
- (2) nadaj wartość początkową sile wypadkowej  $\mathbf{f}_{odp}(\mathbf{q}_i, \mathbf{o}_j) = \mathbf{0}$ ,
- (3) powtarzaj kroki (4)-(6) dla  $k = 1, \dots, \#\mathbf{o}_j$
- (4) wyznacz kierunek działania siły  $\mathbf{r}^* \leftarrow \mathbf{q}_i - \mathbf{o}_{j,k}$
- (5) oraz odległość  $d_k \leftarrow \|\mathbf{r}^*\|$

(6) jeśli  $d_k < \delta_o$ , wtedy

$$\mathbf{f}_{odp}(\mathbf{q}_i, \mathbf{o}_j) \leftarrow \mathbf{f}_{odp}(\mathbf{q}_i, \mathbf{o}_j) + \bar{\mathbf{r}}/d_k^3, \quad (5.35)$$

gdzie  $\delta_o$  jest przyjętym dodatnim parametrem rzeczywistym określającym zakres poniżej którego należy siłę składową dodawać (pomijanie małych wartości sił przyspiesza działanie algorytmu),

**Krok 4** nadaj niepołożeniowym współrzędnym ścieżki odpowiednie wartości. Współrzędna orientująca platformy w nowych stanach  $\mathbf{q}^+$  powinna mieć taką wartość, aby odpowiadała stycznej do ścieżki współrzędnych położeniowych;

**Krok 5** jeżeli skumulowana zmiana położenia punktów była większa niż o założona wartość parametru  $\delta$  ( $\sum_i \|\mathbf{q}_i^+ - \mathbf{q}_i\| > \delta$ ) zaktualizuj wartości stanów  $\mathbf{q}_i \leftarrow \mathbf{q}_i^+, i = 1, \dots, M-1$  i przejdź do kroku 3.

W przeciwnym przypadku algorytm kończy działanie przekazując jako wyjście uzyskaną ścieżkę.

Uwagi do przedstawionego algorytmu:

- położenie nacisku na współrzędne położeniowe odzwierciedla ich decydujący wpływ na zapewnienie bezkolizyjności ruchu platformy;
- istnieją inne sposoby określania siły elastyczności (5.33) uwzględniające także długości wektorów  $\mathbf{q}_{i-1} - \mathbf{q}_i, \mathbf{q}_{i+1} - \mathbf{q}_i$ , a nie tylko ich kierunki (będące efektem normowania);
- dla przeszkód będących wielobokami możliwe jest analityczne wyliczenie siły oddziaływania oddychającego przez modelowanie oddziaływania przeszkody jako wartości pola wektorowego oddziaływania naładowanych zbioru prętów (krawędzi przeszkody). Kluczową operacją jest w tym przypadku całkowanie;
- przy modelach posiadających większą liczbę współrzędnych niż dwie położeniowe i jedną orientującą platformę (np. kąty obrotu kół) należy w kroku 4 do wyznaczenia ich nowych wartości stosować techniki interpolacyjne z sąsiednich poz platformy (najczęściej najmniej złożona interpolacja liniowa dla dwóch najbliższych sąsiadów);
- w zależności (5.35) zwiększenie wykładnika powoduje zwiększenie siły przy zbliżaniu punktu testowego do brzegu przeszkody;
- liczba punktów dyskretyzujących ścieżkę  $M$  powinna się zmieniać z iteracji na iterację, jeśli ścieżka się istotnie wydłuża bądź skraca;
- po kroku 5 należy przyjąć, że zaimplementowano procedurę interpolującą poz platformy w ścieżkę (by w kroku 2 tę ścieżkę zdyskretyzować). W praktycznych przypadkach jawna interpolacja, jeśli to konieczne, jest stosowana dopiero po zakończeniu działania algorytmu.

### 5.3.2. Poprawienie ścieżki holonomicznej

Zaplanowana ścieżka holonomiczna jest bezpieczna i optymalizuje długość euklidesową. Jednak geodezyjne układów nieholonomicznych (najkrótsze krzywe spełniające równanie układu) nie podlegają prawom geometrii euklidesowej [55]. Przykład parkowania samochodu (układ nieholonomiczny) przekonuje o istotnie większej trudności tego manewru, mierzonej zarówno energetycznie jak i skomplikowaniem ruchów, w porównaniu do jazdy na wprost na tę samą odległość euklidesową. Co więcej, ruch układu nieholonomicznego o minimalnej energii pomiędzy dwoma stanami bardzo rzadko będzie przebiegał po linii prostej. Do planowania optymalnych ścieżek nieholonomicznych należałoby wykorzystać skomplikowane narzędzia geometrii sub-riemannowskiej i rachunku wariacyjnego. W dysertacji posłużymy się znacznie prostszym i szybszym sposobem oceny ścieżek charakterystycznym dla Lie-algebraicznych metody planowania [21]. Korzystając z konsekwencji formuły gCBHD i podsumowanej zależnością (2.9) nietrudno zauważyć, że ścieżka będzie mało kosztowna, gdy styczna do niej jest wyznaczana przez pola wektorowe tak niskiego stopnia jak to tylko możliwe (najlepiej

przez generatory układu) [18]. Sformalizujmy to spostrzeżenie proponując miarę bazującą na lokalnej ewaluacji pól wektorowych rozpinających przestrzeń prędkości. Najpierw wyliczymy koszt ruchu na małym fragmencie ścieżki (trajektorii)  $\mathbf{q}(t)$ ,  $t \in [0, T]$ . Korzystając z równania (2.14) wyznaczamy wektor sterowań rozszerzonych  $\mathbf{v}$  odpowiadający pewnemu przemieszczeniu  $\dot{\mathbf{q}}$  odpowiadającemu stycznej do trajektorii. Lokalny koszt trajektorii  $\mathbf{q}(t)$  wyliczony na krótkim przedziale czasu  $[t, t + \Delta T]$  można opisać wzorem

$$c(t, \mathbf{q}(t), \Delta T) = \sum_{i=1}^s \psi(\mathbf{g}_i(\mathbf{q}(t)), \Delta T) \cdot |\mathbf{v}_i(t)|, \quad (5.36)$$

gdzie  $\psi$  zadane jest zależnością (2.9),  $\Delta T$  jest długością lokalnego horyzontu czasowego, natomiast  $s$  jest liczebnością zbioru pól spełniających w chwili  $t$  warunek rzędu (LARC) uzupełnionych o wszystkie pola o stopniu nie przewyższającym stopnia pól spełniających warunek tw. Chow (2.5)

$$\text{rank}(\mathbf{g} = \{\mathbf{g}_1, \dots, \mathbf{g}_s\}) = n, \quad \mathbf{g} = \{\mathbf{a} : \text{stopień}(\mathbf{a}) = \max_i \text{stopień}(\mathbf{g}_i)\}. \quad (5.37)$$

$\mathbf{v}$  w równaniu (5.36) oznacza współczynniki rozkładu wektora prędkości  $\dot{\mathbf{q}}(t)$  w chwili  $t$  w bazie złożonej z pól  $\mathbf{g}$ . Kombinację pól wektorowych uzyskujemy korzystając z ważonej pseudo-odwrotności Moora-Penrose'a. Dla uzyskania tłumienia (kosztownych) pól wysokiego stopnia wprowadzono do równania (2.14) macierz wag zadaną jako

$$\mathbf{Q}_0 = \text{diag} \{(\Delta T)^{-\text{stopień}(\mathbf{g}_1)/2}, \dots, (\Delta T)^{-\text{stopień}(\mathbf{g}_s)/2}\}. \quad (5.38)$$

Globalny koszt trajektorii  $\mathbf{q}(\cdot)$  uzyskuje się naturalnie – przez zsumowanie kosztów lokalnych (5.36) wzdłuż trajektorii

$$C(\mathbf{q}(\cdot), \Delta T) = \sum_{r=0}^{T/\Delta T} c(r\Delta T, \mathbf{q}(r\Delta T), \Delta T) \quad (5.39)$$

(podobne kryterium oceny ścieżek było rozważane w pracy [71]) Zaletą zdefiniowanego wskaźnika jakości ścieżki (5.39) jest prostota i łatwość wyznaczania. Duża szybkość ewaluacji wskaźnika pozwala na efektywne zbadanie wielu ścieżek i wybór optymalnej.

Ścieżki poddawane ocenie mogą być generowane na wiele sposobów. Efektywnym obliczeniowo podejściem wydaje się zastosowanie algorytmu popraw składającego się z czterech kroków:

**Krok 1** pobranie ścieżki wejściowej, wyznaczonej na etapie planowania holonomicznego  $\mathbf{q}(\cdot)$  i jej zdyskretyzowanie, gdy była w postaci analitycznej  $\mathbf{q}_i$ ,  $i = 0, 1, \dots, M$  oraz wyznaczenie kosztu ścieżki  $\text{koszt}(\mathbf{q}(\cdot)) = \text{koszt\_min}$

**Krok 2** poddanie ścieżki  $\mathbf{q}$  lokalnym zaburzeniom — utworzenie zbioru ścieżek testowych  $\mathbf{q}_j^{\text{test}}$ ,  $j = 1, \dots$ . Lokalność zaburzenia polega bądź na modyfikacji jednego lub niewielu punktów ścieżki, bądź na modyfikacji wszystkich punktów z ograniczoną (małą) amplitudą zaburzenia. Każda ze ścieżek testowych musi być ponadto bezkolizyjna oraz posiadać także małe bezkolizyjne otoczenie;

**Krok 3** określenie kosztu realizacji ścieżek testowych

$$\forall j = 1, \dots \quad \text{koszt\_j} = \text{koszt}(\mathbf{q}_j^{\text{test}}),$$

**Krok 4** wyznaczenie kolejnego przybliżenia ścieżki lokalnie optymalnej: jeśli

$$\exists j = 1, \dots \quad \min_j \text{koszt}(\mathbf{q}_j^{\text{test}}) < \text{koszt\_min},$$

wtedy ścieżka o minimalnym koszcie spośród testowych zastępuje ścieżkę  $\mathbf{q}(\cdot)$ , przypisywany jest jej koszt bieżąco minimalny  $\text{koszt\_min} = \min_j \text{koszt}(\mathbf{q}_j^{\text{test}})$ , oraz następuje powrót do kroku 2. W przeciwnym przypadku algorytm kończy działanie, gdyż wyznaczono ścieżkę lokalnie optymalną  $\mathbf{q}(\cdot)$ .

W kolejnych podrozdziałach zaproponowano dwie metody zaburzania ścieżki: lokalnych wariacji oraz lokalnego programowania dynamicznego. Algorytmy tych metod generują ścieżki lokalnie optymalne (tj. w niewielkim otoczeniu wyselekcjonowanych ścieżek nie ma ścieżki o mniejszym koszcie (5.39)). Gdy lokalna optymalność jest niewystarczająca i pożądane jest zwiększenie prawdopodobieństwa otrzymania rozwiązania globalnie optymalnego, należy rozszerzyć obszar poszukiwań i zastosować technikę wielokrotnego startu z różnymi ścieżkami inicjującymi. Parametrami obu algorytmów są: współczynnik kroku  $\xi_d$ , określający wielkość zaburzeń jakiemu poddawane są współrzędne punktów testowych ścieżki, oraz współczynnik wzmocnienia  $\xi_{nh}$ , którym skalowane jest przesunięcie wnoszone od optymalizatora nieholonomicznego — siła nieholonomiczną  $\mathbf{F}_{nh}(\mathbf{q}_i)$  (jako dodatkowy składnik we wzorze (5.32)).

### 5.3.3. Metoda lokalnych wariacji

Mniej złożoną implementacyjnie i obliczeniowo z proponowanych metod zaburzania ścieżki jest metodą lokalnych wariacji [22]. Wprowadźmy oznaczenie  $w(\mathbf{q}_i, \Delta T)$  na koszt ruchu z punktu  $\mathbf{q}_{i-1}$  przez  $\mathbf{q}_i$  do  $\mathbf{q}_{i+1}$  zadany jako

$$w(\mathbf{q}_i, \Delta T) = c(t_{i-1}, \mathbf{q}(t_{i-1}), \Delta T) + c(t_i, \mathbf{q}(t_i), \Delta T), \quad (5.40)$$

gdy ruch odbywa się na małym horyzoncie czasu  $2 \cdot \Delta T$ . Algorytm metody lokalnych wariacji jest zadany następującą sekwencją kroków:

**Krok 1** wczytaj ścieżkę inicjującą, która staje się ścieżką bieżącą;

**Krok 2** zdyskretyzuj ścieżkę bieżącą:  $\mathbf{q}_i, i = 0, \dots, M$ ;

**Krok 3** wybierz pewien punkt pośredni na ścieżce  $\mathbf{q}_i = \mathbf{q}(i \cdot \Delta T), i \in \{1, M-1\}$ ,

**Krok 4** zaburz wybrany punkt

$$\mathbf{q}_i^* \leftarrow \mathbf{q}_i + (r_1 s_1, \dots, r_n s_n)^T, \quad (5.41)$$

gdzie wektor  $\mathbf{s} = (s_1, \dots, s_n)^T$  określa maksymalne zmiany wartości współrzędnych, a wektor współczynników  $\mathbf{r} = (r_1, \dots, r_n)^T \in [-1, 1]^n$  wpływa na rzeczywisty zakres zmian współrzędnych;

**Krok 5** porównaj koszt bieżącej ścieżki i zmodyfikowanej (zmiany są lokalne, więc tylko niewielki fragment ścieżki wymaga przeliczenia kosztu ruchu):

$$\text{jeśli } w(\mathbf{q}_i^*, \Delta T) < w(\mathbf{q}_i, \Delta T) \text{ wtedy } \mathbf{q}_i \leftarrow \mathbf{q}_i^* \quad (5.42)$$

**Krok 6** sprawdź warunek stopu (dookreślony w uwagach): jeśli warunek jest spełniony zatrzymaj wykonywanie algorytmu i zwróć bieżącą ścieżkę jako daną wyjściową; w przeciwnym przypadku kontynuuj od kroku 2.

Podany algorytm dopuszcza pewną dowolność w kilku krokach. Wybór punktu pośredniego (krok 3) można dokonać na wiele sposobów, z których wymienimy:

- wybór losowy,
- w kolejnych iteracjach punkty pośrednie są wybierane od początkowego do końcowego, tj. indeks punktu rośnie od 1 do  $M-1$ ,
- jak powyżej, tym razem indeksy punktów maleją od  $M-1$  do 1,
- jako  $\mathbf{q}_{i^*}$  jest wybierany punkt odpowiadający najbardziej kosztownemu fragmentowi ścieżki w bieżącej iteracji:  $w(\mathbf{q}_{i^*}, \Delta T) = \max_{j \in \{1, M-1\}} w(\mathbf{q}_j, \Delta T)$ .

Wektor maksymalnych dopuszczalnych zmian  $\mathbf{s}$  powinien zależeć od typu współrzędnych (kątowe, translacyjne) i zakresu ruchu na poszczególnych współrzędnych. Wektor współczynników skalujących  $\mathbf{r}$  wybieramy zwykle wśród:

- wierzchołków hiperkostki;  $r_j (j \in \{1, n\})$  wybrane losowo ze zbioru  $\{-1, 1\}$ ;

- wnętrza hiperkostki;  $r_j$  ( $j \in \{1, n\}$ ) wybrane losowo z przedziału  $[-1, 1]$ ;
- sfery,  $\mathbf{r}$  wybrane losowo, ale spełniające warunek  $\|\mathbf{r}\| = 1$ .

Dość naturalnym warunkiem stopu algorytmu jest brak poprawy ścieżki, gdy zaburzenia dokonano we wszystkich możliwych punktach pośrednich. Gdy jest ich wiele, warto rozważyć inne kryteria zakończenia obliczeń ustalające rozsądny kompromis pomiędzy czasem obliczeń a jakością bieżąco najlepszego rozwiązania. Proponowane są dwa warunki stopu:

- obliczenia zostają zatrzymane po upływie  $N$  iteracji, warunek ten nie uwzględnia dynamiki zmian kosztu ścieżki,
- obliczenia zostają zakończone gdy przez  $N_P$  iteracji nie nastąpi poprawa ścieżki.

Możliwe jest także połączenie obu warunków.

### 5.3.4. Metoda lokalnego programowania dynamicznego

Aby rozszerzyć zakres przestrzenny modyfikacji ścieżek rozpatrywanych przez algorytm lokalnych wariacji zaproponowano metodę lokalnego programowania dynamicznego [20]. W metodzie tej odchodzi się od punktowego zaburzenia wybranego punktu ścieżki na rzecz jednoczesnego zaburzenia całej ścieżki. Idea metody powiela schemat klasycznego programowania dynamicznego, jednak ograniczonego (w pojedynczej iteracji) do niewielkiego sąsiedztwa bieżąco optymalnej ścieżki. Ograniczenie liczebności punktów sąsiedztwa ma na celu redukcję złożoności obliczeniowej, która dla algorytmu klasycznego jest ogromna. Na wstępie generowany jest zbiór punktów bezkolizyjnych znajdujących się w sąsiedztwie bieżącej ścieżki, oraz testowane są wszystkie możliwe drogi przechodzące przez punkty ścieżki i jej otoczenia. Ścieżka o najmniejszym koszcie staje się ścieżką wokół której poszukiwana jest kolejna, lepsza, w następnej iteracji. Procedura jest powtarzana do momentu otrzymania stacjonarnej (lokalnie najlepszej) ścieżki. Struktura danych do algorytmu programowania dynamicznego zawiera dla każdego punktu jego bieżąco minimalny koszt drogi do stanu docelowego oraz wskaźnik rodzica, czyli punktu bezpośrednio najbliższego na tej drodze. Algorytm jest następujący:

**Krok 1** wczytaj ścieżkę inicjującą, która staje się ścieżką bieżącą,

**Krok 2** zdyskretyzuj ścieżkę bieżącą  $\mathbf{q}_i$ ,  $i = 0, \dots, M$  i przypisz punktom odpowiadające im chwile czasowe  $t_i = i \cdot \Delta T$ . Dla każdego punktu pośredniego  $\mathbf{q}_i$ ,  $i = 1, \dots, M - 1$ :

a) wygeneruj zbiór sąsiadów  $\mathbf{q}_i^j \in \mathbf{S}_i$ ,  $j = \#\mathbf{S}_i$ ; ( $\#\mathbf{S}_0 = \#\mathbf{S}_M = 1$ )

b) przypisz bieżąco najlepsze koszty dla każdego punktu według następującej reguły:

$$\text{koszt}(\mathbf{q}_0) \leftarrow 0, \text{koszt}(\mathbf{q}_i^j) \leftarrow \infty, i = 0, \dots, M, j = 1, \dots, \#\mathbf{S}_i,$$

**Krok 3** Dla  $i = M - 1, \dots, 0$ , oraz dla  $j = 1, \dots, \#\mathbf{S}_i$ ,  $r = 1, \dots, \#\mathbf{S}_{i+1}$ :

a) wyznacz koszt  $\tilde{c}(\mathbf{q}_i^j, \mathbf{q}_{i+1}^r) = c(i\Delta T, \mathbf{q}_i^j, \Delta T)$  ruchu z punktu  $\mathbf{q}_i^j$  do  $\mathbf{q}_{i+1}^r$ , korzystając z zależności (5.36),

b) Jeżeli  $\text{koszt}(\mathbf{q}_{i+1}^r) > \text{koszt}(\mathbf{q}_i^j) + \tilde{c}(\mathbf{q}_i^j, \mathbf{q}_{i+1}^r)$  wtedy:

b1) przypisz nowy najlepszy koszt  $\text{koszt}(\mathbf{q}_{i+1}^r) \leftarrow \text{koszt}(\mathbf{q}_i^j) + \tilde{c}(\mathbf{q}_i^j, \mathbf{q}_{i+1}^r)$

b2) ustaw wskaźnik punktu  $\mathbf{q}_{i+1}^r$  na punkt  $\mathbf{q}_i^j$  jako rodzica;

**Krok 3** bieżąco najefektywniejsza ścieżka jest składana przez śledzenie wskaźników rodzica, poczynając od punktu  $\mathbf{q}_0$ . Jeżeli najefektywniejsza ścieżka jest identyczna ze ścieżką przed otaczaniem, algorytm kończy działanie wyprowadzając wynikową najefektywniejszą ścieżkę. W przeciwnym przypadku następuje skok do kroku 2.

Złożoność obliczeniowa powyższego algorytmu zależy liniowo od liczby punktów zdyskretyzowanej ścieżki, liniowo od liczby iteracji oraz kwadratowo od liczności stanów w zbiorze  $\mathbf{S}_i$  (zakładając równą liczbę w każdym ze zbiorów dla  $i = 1, \dots, M - 1$ ). Aby zachować niską złożoność obliczeniową pożądane jest utrzymanie niewielkiej liczebności zbiorów  $\mathbf{S}_i$ . Do zbioru  $\mathbf{S}_i$  należy wybrać elementy z podprzestrzeni prostopadłej w  $\mathbf{q}_i$  do segmentu fragmentu ścieżki  $\mathbf{q}_{i-1}, \mathbf{q}_i, \mathbf{q}_{i+1}$ . Proces wyznaczania stanów

należących do tej podprzestrzeni może być długotrwały. Dlatego dla zadań o niewielkim wymiarze wektora stanu, można zdefiniować otoczenie punktu  $\mathbf{q}_i$  poprzez zaburzanie każdej ze współrzędnych  $\mathbf{q}_i$  małą wartością dodatnią lub ujemną. Wartość zaburzająca może być różna dla różnych współrzędnych, co jest szczególnie przydatne w przypadku, gdy współrzędne wchodzące w skład wektora stanu nie są homogeniczne (jednakowe pod względem jednostek i zakresu zmian).

#### 5.4. Dobór sterowań zapewniający bezpieczeństwo manewru platformy

W środowiskach bezkolizyjnych, podczas planowania ścieżki lokalnymi metodami Lie-algebraicznymi, nie jest istotny przebieg trajektorii pomiędzy punktem początkowym a końcowym każdej lokalnej operacji planowania. Inaczej jest w przypadku środowisk kolizyjnych, gdzie wymagana jest kontrola bezkolizyjności przebiegu trajektorii dla każdej chwili czasowej.

Do osiągnięcia założonego lokalnego ruchu wybieramy pewną bazę w przestrzeni sterowań, a następnie przyjmujemy jej skończoną reprezentację. W tym przypadku jednoznaczne określenie sterowań sprowadza się do podania wektora ich współczynników. Minimalna liczba parametrów nie może być mniejsza od wymiarowości przestrzeni stanu, co jest konsekwencją założonej sterowalności układu. Najczęściej reprezentacja jest stała i nie zależy od numeru kolejnego lokalnego planowania. Dla środowisk kolizyjnych liczba współrzędnych tego wektora musi istotnie przewyższać  $n$ , aby przez redundancję w przestrzeni sterowań uzyskać możliwość nie tylko kształtowania trajektorii przejściowych wiodących do identycznej pozycji docelowej platformy ale także jej optymalizacji pod kątem maksymalizacji odległości trajektorii od przeszkód. Istnieją dwa sposoby sprawdzania bezkolizyjności. Pierwszy z nich operuje w przestrzeni fizycznej, co oznacza, że przyjmujemy pewien wektor parametrów sterowań, któremu odpowiada trajektoria zainicjowana w zadanej pozycji platformy i kolizyjność tej trajektorii jest sprawdzana w przestrzeni stanu. Innym, specyficznym dla metod Lie-algebraicznych, sposobem sprawdzania kolizyjności jest sprawdzanie kolizyjności w przestrzeni prędkości. Temu sposobowi poświęcimy więcej uwagi. Pola wektorowe w danym punkcie przestrzeni stanu zapewniające spełnienie warunku rzędu (2.5) tworzą pewien układ współrzędnych. Nie wchodząc w naturę fizyczną (długości i kierunki tych pól w początkowej dla lokalnego planowania pozycji platformy) możemy przyjąć, że wektory tych pól tworzą abstrakcyjny układ ortonormalny. Wektory parametrów sterowań wyznaczają trajektorie w tym abstrakcyjnym układzie. Zauważmy, że trajektorie takie (a przynajmniej ich pewna ilość) może być wyliczona w trybie pre-planowania, gdyż zależą jedynie od (abstrakcyjnych) pól wektorowych a nie od ich parametrów fizycznych (długości i kierunki w konkretnym punkcie). Następnie przeszkody są transformowane ze świata fizycznego do układu abstrakcyjnego i tam odbywa się sprawdzenie kolizyjności. Dla poprawienia efektywności (szybkości) procedury sprawdzającej bezkolizyjność należy przetransformować jedynie przeszkody znajdujące się w najbliższych (lokalnym) zakresie planowania. Z teoretycznego punktu widzenia interesująca jest także charakterystyka zakresu ruchu w układzie abstrakcyjnym. Ten aspekt planowania został przedstawiony w referacie [50] oraz w dodatku A.3, gdzie zdefiniowano kilka wskaźników obszerności manewru bezdryfowego układu nieholonomicznego. Wskaźniki te charakteryzują jaki obszar i jak usytuowany zajmuje trajektoria abstrakcyjna w stosunku do wypadkowego ruchu dla ustalonej iteracji.

Poniżej przedstawiamy konkretny przykład zastosowania takiego podejścia. Niech będzie dany układ platformy nieholonomicznej o trójwymiarowej przestrzeni stanu  $n = 3$  i dwóch sterowaniach,  $m = 2$ , z generatorami  $\mathbf{X}, \mathbf{Y}$  spełniający warunek rzędu (2.5) w każdej pozycji  $\mathbf{q}$  polami wektorowymi  $\mathbf{X}, \mathbf{Y}, [\mathbf{X}, \mathbf{Y}]$ . Lokalną trajektorię, zainicjowaną w pozycji bieżącej  $\mathbf{q}_c$ , rozpatrywać będziemy w układzie współrzędnych wyznaczonym przez abstrakcyjne pola  $\mathbf{X}, \mathbf{Y}, [\mathbf{X}, \mathbf{Y}]$ . Każdą przeszkodę w przestrzeni stanu (czyli każdy jej punkt) można jednoznacznie przekształcić do tego układu. Niech punkt  $\mathbf{o}$  należy do przeszkody (na płaszczyźnie ruchu platformy). Jego współrzędne wyrażone w przestrzeni stanu, oznaczone jako  $\mathbf{o}^{ext}$ , zawierają położeniowe współrzędne  $\mathbf{o}$ , natomiast wszystkie pozostałe współ-



rzędne przyjmują dowolne wartości. Obraz stanów w przestrzeni abstrakcyjnych wektorów pól wektorowych wyliczamy jako:

$$\mathbf{G}(\mathbf{q}_c)^\# \cdot (\mathbf{o}^{ext} - \mathbf{q}_c). \quad (5.43)$$

Nieosobliwość macierzy  $\mathbf{G}(\mathbf{q})$  jest gwarantowana założeniem o sterowalności układu, czyli spełnieniu warunku rzędu algebry Liego (2.5). Jeśli teraz przyjmujemy pewne sterowania  $\mathbf{u}(\cdot)$  działające na horyzoncie  $\tau \in [0, T]$ , odpowiadająca im trajektoria kreśli w przestrzeni abstrakcyjnej krzywą, której bieżący punkt ma współrzędne o współrzędnych  $\boldsymbol{\alpha}(\tau, \mathbf{u}(\cdot))$ , zależnych jedynie od sterowań działających w przedziale  $[0, \tau]$ . Ponieważ zarówno przeszkody jak i trajektoria są w tej samej przestrzeni można dokonać nie tylko sprawdzenia kolizyjności, lecz także optymalizować parametry sterowań pod kątem maksymalizacji odległości od przeszkód. W tym celu warto korzystać z ogólnych własności obrotów w przestrzeni sterowań. Niech sterowania  $(\tilde{u}(\tau), \tilde{v}(\tau))^T$  będą obróconymi sterowaniami  $(u(\tau), v(\tau))^T$  [16]

$$\begin{bmatrix} \tilde{u}(\tau) \\ \tilde{v}(\tau) \end{bmatrix} = Rot(\gamma) \begin{bmatrix} u(\tau) \\ v(\tau) \end{bmatrix}, \quad \text{gdzie } Rot(\gamma) = \begin{bmatrix} \cos \gamma & -\sin \gamma \\ \sin \gamma & \cos \gamma \end{bmatrix}. \quad (5.44)$$

Wtedy, niektóre ze współczynników  $\tilde{\boldsymbol{\alpha}}$  są obracane, inne nie zmieniają wartości:

$$\begin{aligned} \begin{bmatrix} \tilde{\alpha}_X(\tau) \\ \tilde{\alpha}_Y(\tau) \end{bmatrix} &= Rot(\gamma) \begin{bmatrix} \alpha_X(\tau) \\ \alpha_Y(\tau) \end{bmatrix}, \\ \tilde{\alpha}_{[X,Y]}(\tau) &= \alpha_{[X,Y]}(\tau), \\ \begin{bmatrix} \tilde{\alpha}_{[X,[X,Y]]}(\tau) \\ \tilde{\alpha}_{[Y,[X,Y]]}(\tau) \end{bmatrix} &= Rot(\gamma) \begin{bmatrix} \alpha_{[X,[X,Y]]}(\tau) \\ \alpha_{[Y,[X,Y]]}(\tau) \end{bmatrix}. \end{aligned} \quad (5.45)$$

Algorytm doboru sterowań jest następujący:

**Krok 1** zdyskretyzuj ścieżkę wejściową  $M+1$  punktami  $\mathbf{q}_i$   $i = 0, \dots, M$ , gdy sąsiednie punkty znajdują się blisko siebie. Ustaw licznik punktu bieżącego ścieżki na  $i \leftarrow 0$ .

**Krok 2** w punkcie bieżącym  $\mathbf{q}_i$  dokonaj rzutowania przeszkód  $\mathbf{o}$  wyrażonych we współrzędnych stanu  $\mathbf{o}^{ext}$ , do przestrzeni abstrakcyjnych pól  $\mathbf{o}^{abst} = \mathbf{G}(\mathbf{q}_i)^\#(\mathbf{o}^{ext} - \mathbf{q}_i)$ ,

**Krok 3** wyznacz płaszczyznę rzutowania  $\pi$  o równaniu  $Ax + By + Cz + D = 0$  (gdzie  $x, y, z$  odpowiadają abstrakcyjnym polom  $\mathbf{X}, \mathbf{Y}, [\mathbf{X}, \mathbf{Y}]$ ) o normalnej równoległej do zamierzonego kierunku ruchu  $\vec{n} = \mathbf{q}_{i+1} - \mathbf{q}_i$ . Współczynniki wyznaczające płaszczyznę zadane są przez

$$\vec{n} = (A, B, C), \quad D = -\mathbf{q}_c \circ \vec{n}, \quad (5.46)$$

**Krok 4** zrzuć przeszkody  $\mathbf{o}^{abst}$  na płaszczyznę  $\pi$  i wyraż współrzędne w układzie płaszczyzny  $(\eta, \xi)$  (układ płaszczyzny jest dowolnym układem ortonormalnym o początku w  $\mathbf{q}_i$  i leżącym na płaszczyźnie  $\pi$ ). Każda rozpatrywana przeszkoda  $k = 1, \dots$ , tworzy zbiór:  $(\eta_{o_k}, \xi_{o_k})$

**Krok 5** dla przyjętej reprezentacji sterowań zadanych wektorem parametrów  $\mathbf{p}$ , wyznacz rodzinę współczynników sterowań  $\mathbf{p}$  realizujących przemieszczenie stanu od  $\mathbf{q}_i$  do  $\mathbf{q}_{i+1}$ ,

**Krok 6** dla wybranych sterowań  $j = 1, \dots$  z tej rodziny wygeneruj trajektorie zadane współrzędnymi  $\boldsymbol{\alpha}^j(\tau)$ ,  $\tau \in [0, T]$  w układzie abstrakcyjnym. Każda z tych trajektorii jest rzutowana na płaszczyznę  $\pi$  i wyznacza w lokalnym układzie płaszczyzny  $\pi$  zbiór  $(\eta_{\alpha^j(\cdot)}, \xi_{\alpha^j(\cdot)})$ ,

**Krok 7** wyznacz maksymalną z minimalnych odległość punktów poszczególnych trajektorii od przeszkód w lokalnym układzie współrzędnych płaszczyzny  $\pi$

$$d = \max_j \min_k \sqrt{(\eta_{o_k} - \eta_{\alpha^j})^2 + (\xi_{o_k} - \xi_{\alpha^j})^2}, \quad (5.47)$$

gdzie zmienna  $i$  indeksuje przeszkody, a  $j$  testowane trajektorie,

**Krok 8** sprawdź warunek zakończenia iteracji  $i$ -tej: jeżeli  $d < d_{\min}$ , gdzie  $d_{\min}$  jest założonym marginesem bezpieczeństwa oraz rzeczywista trajektoria odpowiadająca sterowaniom wyznaczającym  $d$  jest bezkolizyjna i jej koniec jest odpowiednio blisko stanu  $\mathbf{q}_{i+1}$ , zwiększ licznik iteracji  $i \leftarrow i + 1$ , w przeciwnym przypadku zmień reprezentację sterowań lub jako punkt końcowy trajektorii  $i$ -tej wybierz bliższy  $\mathbf{q}_i$  i powrót do Kroku 5,

**Krok 9** Jeśli osiągnięto odpowiednio dokładnie punkt docelowy, należy zakończyć algorytm. W przeciwnym przypadku następuje powrót do kroku 2.

Uwagi dotyczące algorytmu są następujące: nie wszystkie przeszkody są aktywne podczas sprawdzania lokalnej bezkolizyjności manewru. W teście kolizyjności uwzględnia się jedynie te z najbliższego sąsiedztwa bieżącego stanu platformy. W kroku 8, współczynniki sterowań odpowiadające najbardziej (lub wystarczająco) oddalonej od przeszkód trajektorii, wyznaczającej odległość  $d$ , są podstawiane do modelu platformy i wyznaczana jest rzeczywista trajektoria przez całkowanie numerycznie. W praktycznych realizacjach algorytmu najczęściej wykorzystywana jest metoda Rungego-Kutty czwartego rzędu [65]. Sterowania wynikowe są konkatencją sterowań z kolejnych lokalnych planowań.

Planowanie ruchu metodą Lie-algebraiczną jest procesem lokalnym, gdyż dokładność realizacji zadanego kierunku osiągania kolejnych punktów ścieżki maleje wraz ze wzrostem odległości między pozą początkową i końcową dla danej iteracji. W powyższym algorytmie liczba planowań może być duża, gdy ścieżka realizowana jest długa – a jest to cecha niepożądana. By fakt ten uzasadnić, przeanalizujemy klasyczny przykład generacji (aproksymacji) nawiasu Liego  $[\mathbf{X}, \mathbf{Y}]$  przez efektywny (i krótki) czas  $t$  przy pomocy sterowań kawałkami stałych:

$$\exp(t[\mathbf{X}, \mathbf{Y}]) = \exp(\sqrt{t}\mathbf{X}) \circ \exp(\sqrt{t}\mathbf{Y}) \circ \exp(-\sqrt{t}\mathbf{X}) \circ \exp(-\sqrt{t}\mathbf{Y}), \quad (5.48)$$

gdzie  $\circ$  oznacza konkatencję,  $\exp(\mathbf{Z})$  oznacza trajektorię pola  $\mathbf{Z}$ , a prawą stronę (5.48) czytamy od lewej strony ku prawej (najpierw aplikowane sterowanie wzdłuż pola  $\mathbf{X}$  o amplitudzie 1, itd.). Energia sterowań dla scenariusza (5.48) wynosi  $4 \cdot (\pm 1)^2 \sqrt{t}$ , a jej wartość przypadająca na jednostkę długości wynosi  $4/\sqrt{t}$ . Jeśli teraz zmniejszymy czas generacji o połowę, to energia przypadająca na jednostkę długości wynosi  $4\sqrt{t/2}/(t/2) = \sqrt{2} \cdot 4/\sqrt{t}$ . Zatem wydłużanie czasu ruchu efektywnego przynosi wymierne korzyści w postaci zmniejszania energii sterowań a nie pokonanie tej samej efektywnej odległości. Dlatego warto zwiększyć zakresu działania metody lokalnej, nawet akceptując zmniejszenie dokładności realizacji kolejnych podcelów, ciągle zachowując jednak bezkolizyjność ścieżki. Aby zmniejszyć liczbę lokalnych planowań modyfikujemy algorytm powyższy do następującego:

**Krok 1** z zadanej ścieżki wejściowej przyjmij punkt początkowy  $\mathbf{q}_0$  jako pozę początkową dla  $i \leftarrow 0$  iteracji,

**Krok 2** przyjmij dokładność  $\delta_i$  osiągnięcia podcelu  $i$ -tej iteracji. Dokładność musi być bliska zeru, gdy podcelem lokalnym jest poza końcowa  $\mathbf{q}_f$  ścieżki zadanej,

**Krok 3** w jednowymiarowym procesie optymalizacyjnym uzmienniana jest poza docelowa  $\mathbf{q}_{i+1}$  dla  $i$ -tej iteracji, położona na ścieżce zadanej bliżej pozy  $\mathbf{q}_f$ . Optymalna wartość  $\mathbf{q}_{i+1}^*$  odpowiada minimalnej odległości mierzonej wzdłuż ścieżki zadanej od pozy docelowej  $\mathbf{q}_f$  przy zachowaniu bezkolizyjności trajektorii. W tym kroku algorytmu wielokrotnie wykorzystywana jest część algorytmu poprzedniego dotycząca planowania jednej iteracji. Część trajektorii wynikowej odpowiadająca iteracji  $i$ -tej jest bezkolizyjna, acz rzeczywiście osiągnięta poza docelowa  $\mathbf{q}_{i+1}^r$  iteracji niekoniecznie musi być dowolnie blisko pozy docelowej iteracji  $\mathbf{q}_{i+1}^*$ , spełnia jedynie warunek  $\|\mathbf{q}_{i+1}^r - \mathbf{q}_{i+1}^*\| < \delta_i$ .

**Krok 4** jeżeli  $\|\mathbf{q}_{i+1}^r - \mathbf{q}_f\| < \delta_i$ , osiągnięto pozę docelową ścieżki i należy zakończyć obliczenia wyznaczając jako wynik trajektorię będącą konkatencją trajektorii kolejnych iteracji. W przeciwnym przypadku zwiększyć licznik iteracji:  $i \leftarrow i + 1$  i powrócić do kroku 2.

W wyniku działania algorytmu segmenty przypadające na jedną iterację mogą być istotnie dłuższe niż w algorytmie poprzednim, szczególnie wtedy, gdy dla generacji lokalnej trajektorii wymagane jest użycie samych generatorów układu (3.1), oraz gdy, lokalnie, przeszkody są odległe od planowanego segmentu trajektorii.

### 5.5. Rozwiązywalność zadania planowania ruchu platformy

Obecnie przedstawimy warunek wystarczający rozwiązania zadania planowania ruchu platformy nałożony na wygenerowaną ścieżkę (po etapie jej planowania holonomicznego i nieholonomicznej modyfikacji)

**Twierdzenie 5.5.1.** *Gdy ścieżka  $\mathbf{q}(s)$ ,  $s \in [0, s_{fin}]$  bezdryfowego układu (3.1) o analitycznych polach wektorowych  $\mathbf{g} = \{\mathbf{g}_1, \dots, \mathbf{g}_m\}$  i spełniającego warunek rzędu (2.5) jest bezkolizyjna wraz z jej otwartym otoczeniem  $\mathbf{B}$ , tj.*

$$\exists \varepsilon > 0 \quad \forall s \quad B(\mathbf{q}(s), \varepsilon) - \mathbf{O} = \mathbf{O}, \quad \mathbf{B} = \bigcup_{s \in [0, s_{fin}]} B(\mathbf{q}(s), \varepsilon), \quad (5.49)$$

gdzie  $B(\mathbf{q}, \varepsilon)$  oznacza kulę scentrowaną w punkcie  $\mathbf{q}$  o promieniu  $\varepsilon$ , wtedy istnieją takie sterowania  $\mathbf{u}(t)$ ,  $t \in [0, T]$  ( $T$  ograniczony horyzont czasowy) generowane metodą Lie-algebraiczną, że trajektoria zainicjowana w stanie  $\mathbf{q}(s=0)$  i odpowiadająca sterowaniom  $\mathbf{u}(\cdot)$  osiąga stan docelowy  $\mathbf{q}(s_{fin})$  dowolnie dokładnie, oraz leży wewnątrz zbioru  $\mathbf{B}$ , czyli jest bezkolizyjna.

**Dowód 1.** *Najpierw udowodnimy twierdzenie nieco słabsze, po nałożeniu dodatkowego warunku, że każdy punkt zbioru  $\mathbf{B}$  jest dowolnie blisko (w sensie pewnej metryki) stanu  $\mathbf{q}(0)$  (w szczególności oznacza to bliskość także stanu  $\mathbf{q}(s_{fin})$ ).*

*Skoro układ spełnia warunek rzędu, więc istnieje  $n$  pól wektorowych zwartościowanych w  $\mathbf{q}(0)$  (zatem wektorów), które, po zgrupowaniu, tworzą macierz nieosobliwą. Układów pól spełniających tę własność w  $\mathbf{q}(0)$  może być wiele. Dlatego wybieramy taki układ, który zawiera pola o minimalnych stopniach (w przypadku wielu równoważnych układów - dowolny). Do tego zbioru dołączmy wszystkie pola o stopniu nie większym od pola o stopniu maksymalnym z pól uprzednio wyselekcjonowanych tworząc zbiór poszerzony. Zatem wektor  $\mathbf{v} = \mathbf{q}(s_{fin}) - \mathbf{q}(0)$  może być jednoznacznie (przy pomocy pseudoodwrotności Moore'a-Penrose'a) przedstawiony jako kombinacja wektorów zbioru poszerzonego. Co więcej, współczynniki kombinacji mają ograniczony moduł. Z bliskości stanów wnioskujemy, że sterowania mogą być generowane w dowolnie krótkim czasie. Na mocy formuły gCBHD (2.6) oraz własności (2.9) wnioskujemy, że dla małych czasów  $T$  zaburzający wpływ pól wyższych stopni, niż tych ze zbioru poszerzonego, na przemieszczenie stanu o wektor  $\mathbf{v}$  można uczynić dowolnie małym, czyli stan końcowy  $\mathbf{q}(0) + \mathbf{v}$  może być osiągnięty z dowolną dokładnością. Teraz wybieramy odpowiednio bogatą rodzinę sterowań (o liczbie parametrów co najmniej równej  $n$ ) które mogą wygenerować zgodnie ze wzorami podobnymi w strukturze do tych przedstawionych dla przypadku szczególnego ( $n = 3, m = 2$ ) (2.7), (2.8) współczynniki kombinacji liniowej wektorów zbioru poszerzonego. Ponieważ przedział sterowania jest skończony, a sterowania ograniczone, zatem trajektoria (na mocy twierdzenia o ciągłej zależności rozwiązania równania różniczkowego od warunków początkowych) musi się zawierać w pewnej kuli scentrowanej w  $\mathbf{q}(0)$  o promieniu  $\varepsilon_1$ . Na mocy założenia o dowolnie bliskości punktów  $\mathbf{q}(0)$  i  $\mathbf{q}(s_{fin})$  wnioskujemy, że  $\varepsilon_1 \leq \varepsilon$  i mamy też twierdzenia słabszego.*

*Też twierdzenia 5.5.1 uzyskujemy przez taki podział ścieżki, że między jej sąsiednimi punktami ruch odbywa się w kuli o promieniu co najwyżej  $\varepsilon$ .*

## 6. Symulacje

W rozdziale przedstawiono testy symulacyjne oprogramowania planowania ścieżek. W pierwszych dwóch podrozdziałach zaprezentowano modele manipulatorów i platform mobilnych, następnie omówione wykorzystane sterowania, kolejno przedstawione zostaną przeszkody a na końcu wyniki symulacji.

### 6.1. Modele manipulatorów

W pracy przebadano trzy różne rodzaje manipulatorów zamontowanych na platformie: manipulator „pusty” (brak manipulatora), dwuwahadło planarne oraz dwuwahadło umieszczone na obrotowym stopniu swobody.

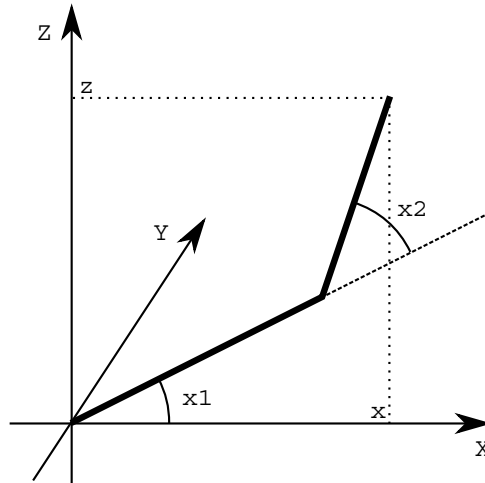
#### Dwuwahadło planarne

Parametry geometryczne manipulatora:  $l_1, l_2$  — długości ramion. Wektor konfiguracji:

$$\mathbf{x} = [x_1 \ x_2]^T, \quad (6.1)$$

gdzie  $x_1, x_2$  są kątami odchylenia ramienia względem poprzedniego stopnia. Pełna kinematyka manipulatora:

$$\mathbf{k}_{2w}(\mathbf{x}) = \begin{bmatrix} \sin(x_1 + x_2) & \cos(x_1 + x_2) & 0 & l_1 \sin x_1 + l_2 \sin(x_1 + x_2) \\ 0 & 0 & 1 & 0 \\ \cos(x_1 + x_2) & -\sin(x_1 + x_2) & 0 & l_1 \cos x_1 + l_2 \cos(x_1 + x_2) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.2)$$



Rysunek 6.1. Dwuwahadło

Za kinematykę uważamy kinematykę we współrzędnych:

$$\mathbf{k}_{2w}(\mathbf{x}) = \begin{bmatrix} l_1 \sin x_1 + l_2 \sin(x_1 + x_2) \\ 0 \\ l_1 \cos x_1 + l_2 \cos(x_1 + x_2) \end{bmatrix}. \quad (6.3)$$

### Dwuwahadło planarne ze stopniem obrotowym

Parametrami geometrycznymi są długości ramion:  $l_1, l_2$ , wektor konfiguracji

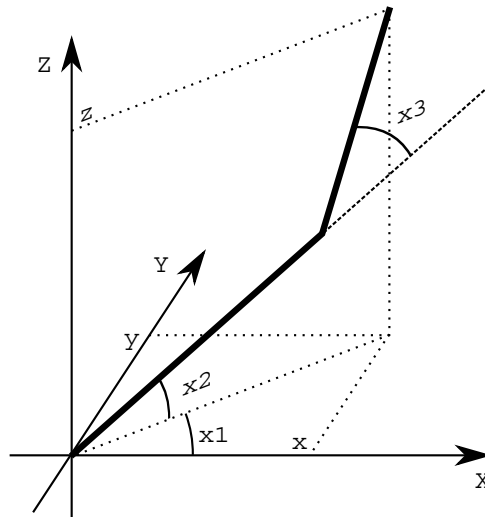
$$\mathbf{x} = [x_1 \ x_2 \ x_3]^T, \quad (6.4)$$

gdzie  $x_1$  jest kątem obrotu płaszczyzny dwuwahadła względem osi Z,  $x_2, x_3$  są kątami obrotu ramion wahadła względem poprzedniego stopnia. Kinematyka pełna manipulatora:

$$\mathbf{k}_{r2w}(\mathbf{x}) = \begin{bmatrix} \cos x_1 \sin(x_2 + x_3) & \cos x_1 \cos(x_2 + x_3) & -\sin x_1 & \cos x_1 (l_1 \sin x_2 + l_2 \sin(x_2 + x_3)) \\ \sin x_1 \sin(x_2 + x_3) & \sin x_1 \cos(x_2 + x_3) & \cos x_1 & \sin x_1 (l_1 \sin x_2 + l_2 \sin(x_2 + x_3)) \\ \cos(x_2 + x_3) & -\sin(x_2 + x_3) & 0 & l_1 \cos x_2 + l_2 \cos(x_2 + x_3) \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (6.5)$$

Przyjmuje się kinematykę we współrzędnych:

$$\mathbf{k}_{r2w}(\mathbf{x}) = \begin{bmatrix} \cos x_1 (l_1 \sin x_2 + l_2 \sin(x_2 + x_3)) \\ \sin x_1 (l_1 \sin x_2 + l_2 \sin(x_2 + x_3)) \\ l_1 \cos x_2 + l_2 \cos(x_2 + x_3) \end{bmatrix}. \quad (6.6)$$



Rysunek 6.2. Dwuwahadło ze stopniem obrotowym

## 6.2. Modele platform

W pracy przyjęto do rozważań klasę układów nieholonomicznych o dwóch sterowaniach

$$\dot{\mathbf{q}}(t) = \mathbf{X}(\mathbf{q}(t))u(t) + \mathbf{Y}(\mathbf{q}(t))v(t) \quad (6.7)$$

przykładami takich układów są:

- monocykl (*unicycle*)
- dwukołowiec

Modele układów wynikają z ograniczeń nieholonomicznych w postaci Pfaffa:

$$A(q)\dot{q} = 0 \quad (6.8)$$

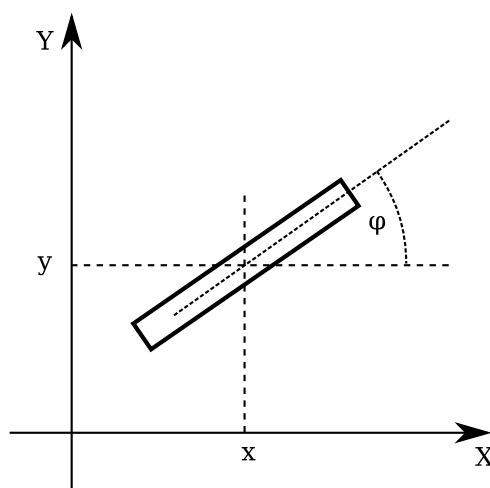
generowane założeniem braku poślizgu kół.

### 6.2.1. Monocykl

Monocykl jest najprostszą platformą robotyczną. Posiada wektor stanu

$$q = [x \ y \ \varphi]^T \quad (6.9)$$

na który składają się współrzędne kartezjańskie środka platformy  $x, y$  oraz  $\varphi$  kątem obrotu od osi  $X$ .



Rysunek 6.3. Monocykl

Ograniczenie ruchu bez poślizgu przyjmuje postać

$$\begin{bmatrix} \sin \varphi & -\cos \varphi & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = 0. \quad (6.10)$$

Stąd wyprowadzić można bezdryfowy układ sterowania w postaci

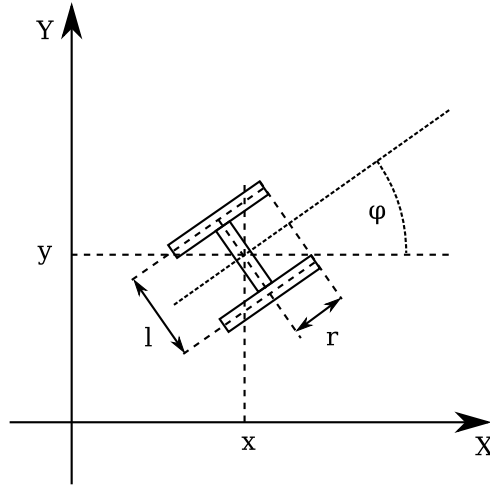
$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos \varphi \\ \sin \varphi \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v(t) = \mathbf{g}_1(q)u(t) + \mathbf{g}_2(q)v(t). \quad (6.11)$$

Sterowanie  $u(\cdot)$  odpowiada ruchowi postępowemu a  $v(\cdot)$  obrotowi platformy. Pola wektorowe monocykla:

$$\begin{aligned}
 \mathbf{g}_1 &= [\cos \varphi \quad \sin \varphi \quad 0]^T \\
 \mathbf{g}_2 &= [0 \quad 0 \quad 1]^T \\
 [\mathbf{g}_1, \mathbf{g}_2] &= [\sin \varphi \quad -\cos \varphi \quad 0]^T \\
 [\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]] &= [0 \quad 0 \quad 0]^T \\
 [\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]] &= [\cos \varphi \quad \sin \varphi \quad 0]^T = \mathbf{g}_1 \\
 [\mathbf{g}_1, [\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]]] &= [0 \quad 0 \quad 0]^T \\
 [\mathbf{g}_2, [\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]]] &= [0 \quad 0 \quad 0]^T \\
 [\mathbf{g}_2, [\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]]] &= [-\sin \varphi \quad \cos \varphi \quad 0]^T = -[\mathbf{g}_1, \mathbf{g}_2]
 \end{aligned} \tag{6.12}$$

### 6.2.2. Dwukołowiec

Dwukołowiec jest platformą wyposażoną w dwa koła o niezależnym napędzie. Parametrami geo-



Rysunek 6.4. Dwukołowiec

metrycznymi platformy mobilnej są: promień kół  $r$  oraz odległość pomiędzy kołami  $l$ . Wektor stanu

$$\mathbf{q} = [x \quad y \quad \varphi]^T \tag{6.13}$$

składa się z położenia środka platformy  $x, y$  oraz z obrotu platformy  $\varphi$  względem osi  $Z$ .

Układ sterowania związany z dwukołowcem:

$$\dot{\mathbf{q}} = \begin{bmatrix} \frac{r}{2} \cos \varphi \\ \frac{r}{2} \sin \varphi \\ -\frac{r}{l} \end{bmatrix} \omega_l + \begin{bmatrix} \frac{r}{2} \cos \varphi \\ \frac{r}{2} \sin \varphi \\ \frac{r}{l} \end{bmatrix} \omega_p, \tag{6.14}$$

gdzie  $\omega_l, \omega_p$  są prędkościami obrotu koła lewego i prawego. Pola wektorowe dla dwukołowca są następujące:

$$\begin{aligned}
 [\mathbf{g}_1, \mathbf{g}_2] &= \begin{bmatrix} \frac{r^2}{l} \sin \varphi & -\frac{r^2}{l} \cos \varphi & 0 \end{bmatrix}^T \\
 [\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]] &= \begin{bmatrix} -\frac{r^3}{l^2} \cos \varphi & -\frac{r^3}{l^2} \sin \varphi & 0 \end{bmatrix}^T \\
 [\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]] &= \begin{bmatrix} \frac{r^3}{l^2} \cos \varphi & \frac{r^3}{l^2} \sin \varphi & 0 \end{bmatrix}^T \\
 [\mathbf{g}_1, [\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]]] &= \begin{bmatrix} -\frac{r^4}{l^3} \sin \varphi & \frac{r^2}{l^3} \cos \varphi & 0 \end{bmatrix}^T \\
 [\mathbf{g}_2, [\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]]] &= \begin{bmatrix} \frac{r^4}{l^3} \sin \varphi & -\frac{r^2}{l^3} \cos \varphi & 0 \end{bmatrix}^T \\
 [\mathbf{g}_2, [\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]]] &= \begin{bmatrix} -\frac{r^4}{l^3} \sin \varphi & \frac{r^2}{l^3} \cos \varphi & 0 \end{bmatrix}^T.
 \end{aligned} \tag{6.15}$$

### 6.3. Przeszkody

Na użytek rozprawy, przyjęto reprezentacje przeszkód w przestrzeni wielokątów zadanych przez kolejne współrzędne wierzchołków. Środowisko manipulatora składa się z dowolnej liczby przeszkód zawartych w zbiorze  $\mathbf{O}$ . Pojedyncza przeszkoda ma oznaczenie  $\mathbf{o}_i$ ,  $i \in \{1, \dots, \#\mathbf{O}\}$ , i jest zbiorem kolejnych wierzchołków,  $\mathbf{o}_{i,j}$  których współrzędne wyrażone są w globalnym układzie odniesienia.

Przyjęta reprezentacja przeszkód pozwala w prosty sposób wyznaczyć odległość testowanej trajektorii od przeszkód (patrz dodatek A.2). Dzięki temu możliwa jest dobór sterowań zapewniający maksymalizację odległości trajektorii od przeszkód.

### 6.4. Sterowania

W lokalnym (wokół ustalonego punktu przestrzeni stanu) zadaniu planowania ruchu na podstawie znanych wartości  $\boldsymbol{\alpha}$  (znany kierunek ruchu do celu) należy wyznaczyć sterowania  $u(\cdot)$ ,  $v(\cdot)$  je realizujące, czyli rozwiązać pewne zadanie odwrotne. Rozwiązanie równania (2.8) — wyznaczenie funkcji  $u(\cdot)$ ,  $v(\cdot)$  jest zadaniem skomplikowanym. Dlatego przyjmuje się zwykle bazę w przestrzeni sterowań (funkcje harmoniczne, wielomianowe), wyrażając sterowania jako kombinację liniową funkcji bazowych. Poszukiwanie sterowań zastępuje się wyznaczaniem ich współczynników.

W pracy przyjęto sterowania z bazy Fouriera o jednej harmonicznej, choć możliwy jest również wybór sterowań z innych baz [19]. Zaletą takiego wyboru jest efektywna możliwość wpływania na energię sterowań. Baza ortonormalna funkcji  $f_i(s)$  na przedziale  $[0, T]$  spełnia warunek:

$$\int_0^T f_i(s) f_j(s) ds = \delta_{ij}, \tag{6.16}$$

gdzie

$$\delta_{ij} = \begin{cases} 1 & \text{gdy } i = j \\ 0 & \text{gdy } i \neq j. \end{cases}$$



Dla bazy Fouriera funkcje bazowe przyjmują następującą postać:

$$\begin{aligned} f_0(s) &= \sqrt{\frac{1}{T}}, \\ f_{2n-1}(s) &= \sqrt{\frac{2}{T}} \sin n\omega s, \\ f_{2n}(s) &= \sqrt{\frac{2}{T}} \cos n\omega s, \end{aligned}$$

gdzie  $\omega = \frac{2\pi}{T}$ .

Przyjęta postać sterowań:

$$\begin{aligned} u(s) &= a_0 \sqrt{\frac{1}{T}} + A \sqrt{\frac{2}{T}} \sin(\omega s + \varphi), \\ v(s) &= b_0 \sqrt{\frac{1}{T}} + B \sqrt{\frac{2}{T}} \sin(\omega s + \psi). \end{aligned} \quad (6.17)$$

Energia sterowań (6.17) wynosi:

$$E = \int_0^T (u^2(t) + v^2(t)) dt = a_0^2 + b_0^2 + A^2 + B^2 \quad (6.18)$$

Współczynniki gCBHD, porównaj (2.8), dla sterowań (6.17) wynoszą:

$$\begin{aligned} \alpha_X &= \sqrt{T} a_0 \\ \alpha_Y &= \sqrt{T} b_0 \\ \alpha_{[X,Y]} &= \frac{T}{2!\pi} \left( \sqrt{2} (Ab_0 \cos \varphi - Ba_0 \cos \psi) + AB \sin(\varphi - \psi) \right) \\ \alpha_{[X,[X,Y]]} &= \frac{T^{3/2}}{3!\pi^2} \frac{3}{4} \left( 2A^2 b_0 + A^2 b_0 \cos(2\varphi) - 2ABa_0 \cos(\varphi - \psi) - ABa_0 \cos(\varphi + \psi) \right. \\ &\quad \left. + \sqrt{2} (-2Aa_0 b_0 \sin \varphi + A^2 B \sin(2\varphi - \psi) - A^2 B \sin \psi + 2a_0^2 B \sin \psi) \right) \\ \alpha_{[Y,[X,Y]]} &= \frac{T^{3/2}}{3!\pi^2} \frac{3}{4} \left( -2B^2 a_0 + 2ABb_0 \cos(\varphi - \psi) - B^2 a_0 \cos(2\psi) + ABb_0 \cos(\varphi + \psi) \right. \\ &\quad \left. + \sqrt{2} (AB^2 \sin \varphi - 2Ab_0^2 \sin \varphi + AB^2 \sin(\varphi - 2\psi) + 2Ba_0 b_0 \sin \psi) \right). \end{aligned} \quad (6.19)$$

## 6.5. Parametry algorytmu planowania ruchu

Działanie algorytmu planowania ruchu manipulatora mobilnego zależy od wielu parametrów. Jako parametry nominalne do testów przyjęto:

Warunki przyjęte za nominalne:

- Dokładność realizacji ścieżki  $\delta_d = 0,01$ .
- Liczba punktów dyskretyzowanej ścieżki  $N = 100$ .
- Współczynnik odpychania od przeszkód  $\xi_o = 0,001$ .
- Współczynnik elastyczności  $\xi_e = 0,5$ .
- Zasięg oddziaływania odpychającego od przeszkód  $\delta_o = 1$ .

- Odległości pomiędzy stanem osiągniętym z całkowania sterowań, a stanem docelowym liczone z uwzględnieniem rodzaju współrzędnych (liniowe lub kątowe).
- Środowisko składające się z trzech przeszkód, pomiędzy którymi powinna przebiegać trajektoria.
- Aby uniezależnić wyniki od doboru szczególnej wartości współrzędnych punktu startowego, wylosowano 100 punktów z obszaru gwarantującego przejście między przeszkodami – trójkąt o wierzchołkach  $(-10, -2.5)$ ,  $(-10, -1)$ ,  $(-3, -2.5)$ . Przedstawione wyniki są średnią arytmetyczną 100 prób.
- Liczba podpunktów całkowania: 1000.
- Brak ograniczeń na wartości parametrów sterowań  $(a_0, a_1, b_0, b_1, \varphi, \psi)$ .

Symulacje zostały przeprowadzone dla dwóch rodzajów platform: monocykla i dwukółowca i dla trzech rodzajów manipulatorów umieszczonych na platformie: bez manipulatora, dwuwahadło, dwuwahadło ze stopniem obrotowym w podstawie. Trajektorie i sterowania dla wariantu nominalnego zostały przedstawione na wykresach 6.5 i 6.6.

## 6.6. Badania symulacyjne

### 6.6.1. Zbiór konfiguracji docelowych

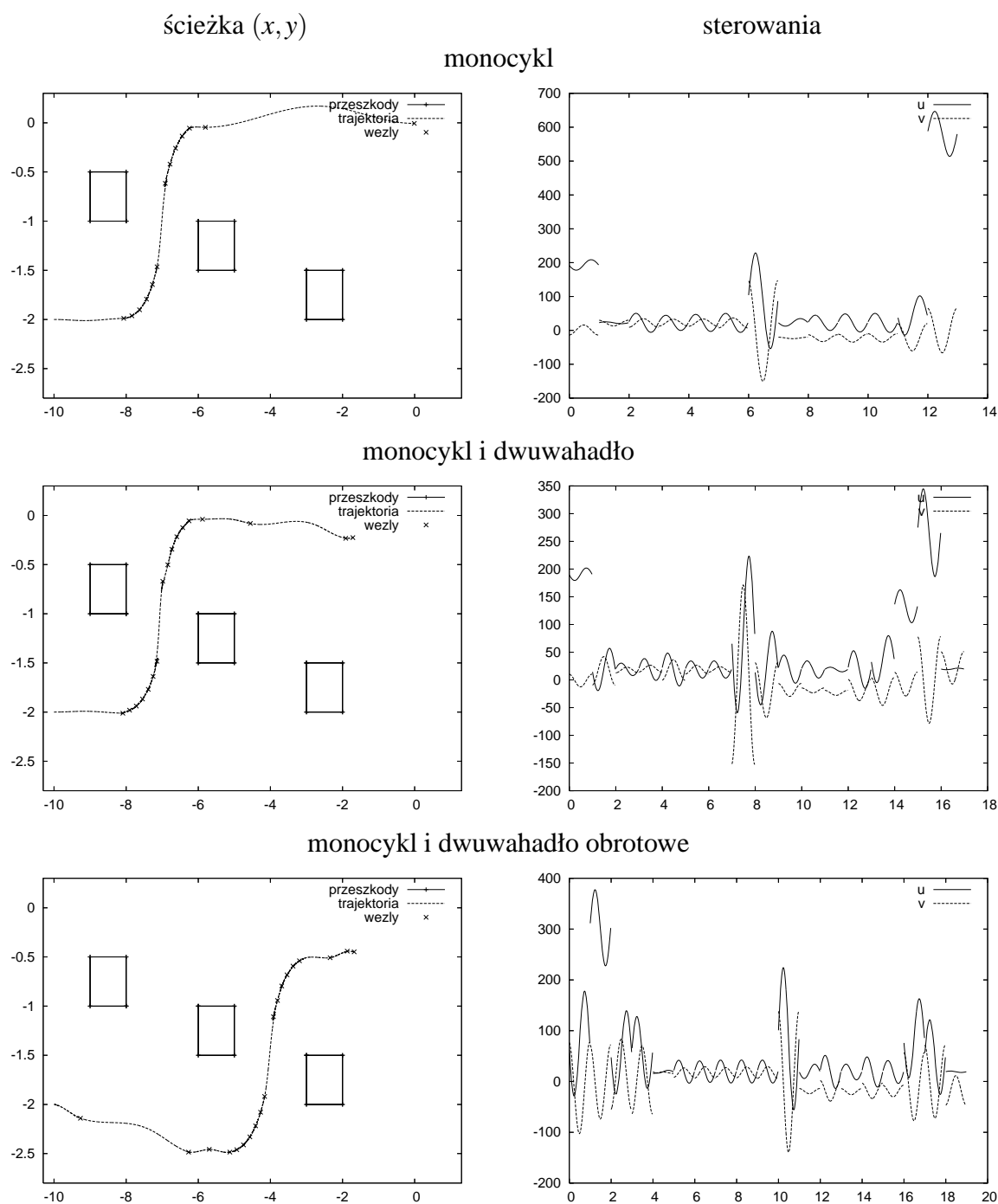
Symulacje przeprowadzono na modelu monocykla z zamontowanym dwuwahadłem. Kryterium podstawowym było osiągnięcie konfiguracji docelowej  $\mathbf{y}_f = TrZ(1)$ . Długości ramion manipulatora  $(l_1, l_2)$  wybrano na dwa sposoby: aby utworzyć wewnętrzny brzeg ( $l_1 = 1 < l_2 = 3.75$ ) oraz aby go nie tworzyć ( $l_1 = 1 > l_2 = 0.75$ ). Stan początkowy platformy  $(x_p, 0, 0^\circ)$  był wybierany na dwa sposoby: w odległości przekraczającej sumę długości ramion manipulatora ( $x_p = -20$ ), „daleko” oraz w zasięgu manipulatora ( $x_p = -3.5$  lub  $x_p = -1$ ), „blisko”.

Opis wartości odpowiada zastosowanemu kryterium dodatkowemu. Średnia liczba iteracji każdego wariantu algorytmu wynosiła 177. Na rysunku 6.7 przedstawiono stany platform w momencie osiągnięcia konfiguracji docelowej  $\mathbf{y}_d$  odpowiednio dla przypadków „bliskiego” i „dalekiego”. Funkcje  $d_1(\mathbf{q})$  i  $d_2(\mathbf{q})$  wykazują znaczną zależność od odległości startowej. Na rysunku 6.8 przedstawiono stan przy użyciu funkcji  $d_4(\mathbf{q})$ . W tym przypadku odległość początkowa nie ma znaczenia. Na rysunku 6.9 przedstawiono wariant manipulatora posiadający wewnętrzny brzeg. W tym przypadku punkt startowy został umieszczony „daleko”. Funkcja  $d_5(\mathbf{q})$  zachowuje się podobnie jak  $d_4(\mathbf{q})$  — lepiej odtwarza kształt brzegu w pobliżu końców testowanych zakresów kątów.

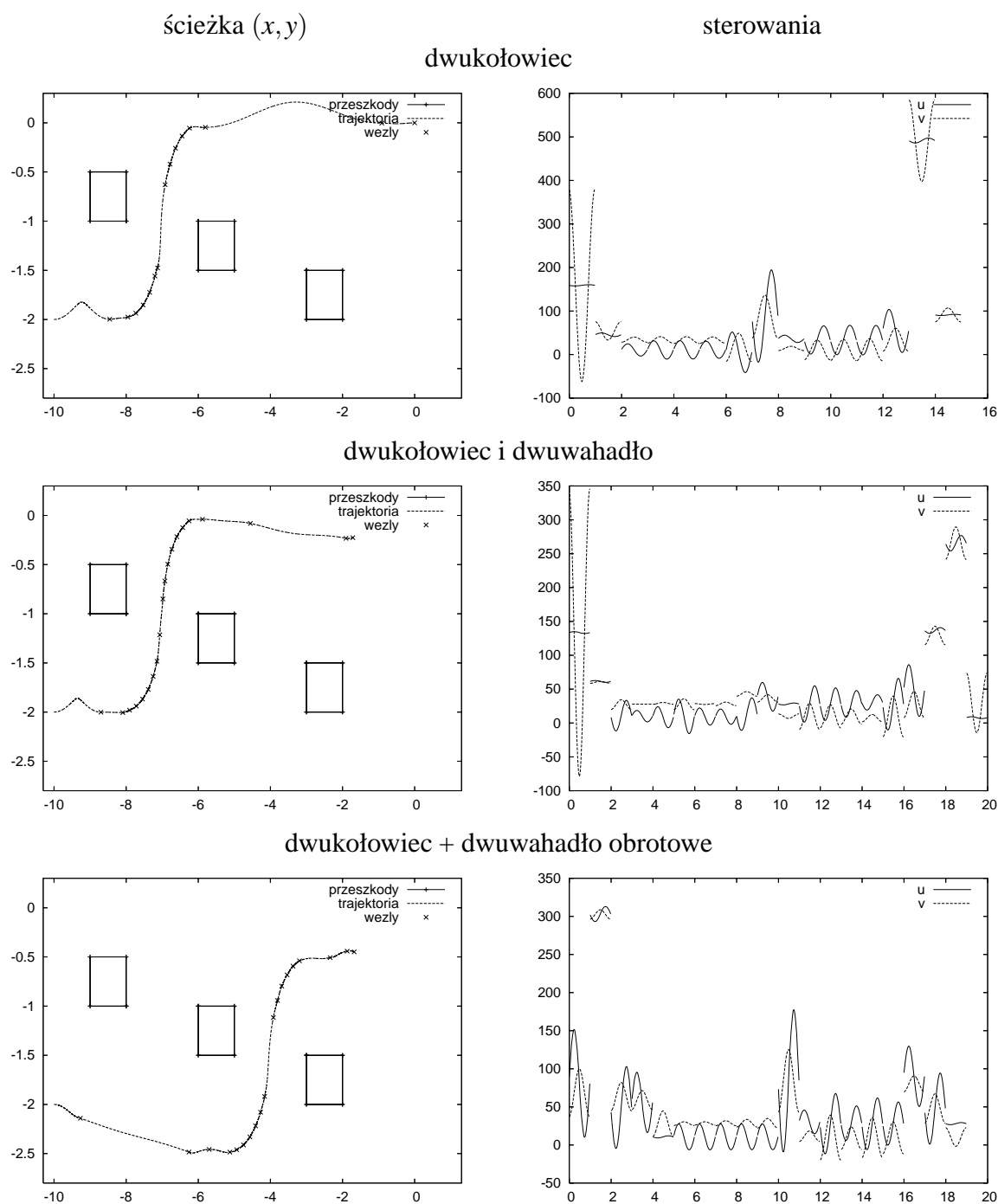
Przetestowano również określanie orientacji. Dla manipulatora typu dwuwahadło, orientacja platformy musiała być określona — rozwiązanie zadania ze zmienionym stanem platformy nie było możliwe. W przypadku manipulatora typu dwuwahadło zamocowanego na przegubie obrotowym, algorytm rozwiązywał zadanie średnio w 161 krokach.

### 6.6.2. Zmiana dokładności realizacji ścieżki

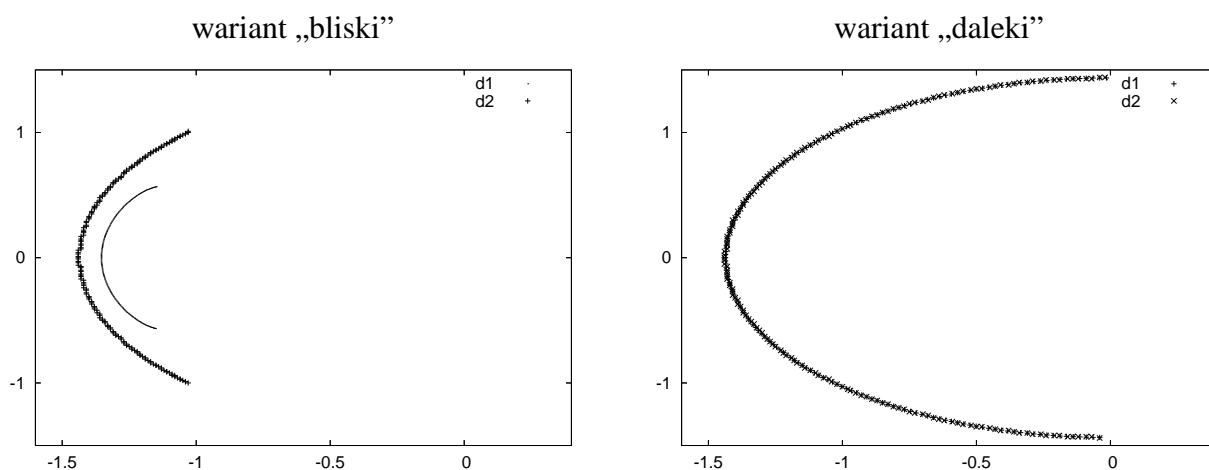
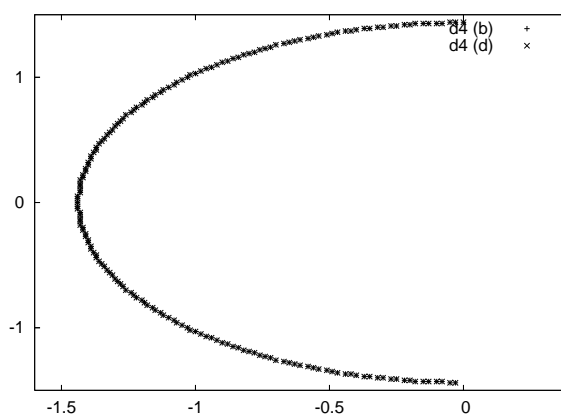
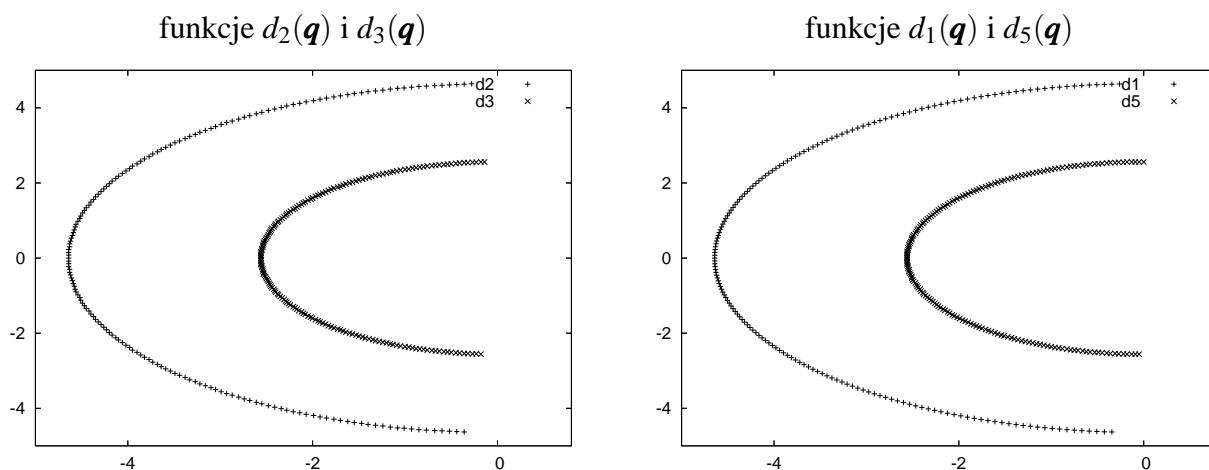
Przetestowano wpływ dokładności realizacji zadanej ścieżki (maksymalnej dopuszczalnej odchyłki pomiędzy stanem pożądanym a zrealizowanym). Dokładność zmieniana była w pięciu krokach w zakresie 0.001 do 0.25. W tablicy 6.1 przedstawiono odpowiednio liczbę elementów ścieżki wynikowej, sumaryczną energię sterowań oraz czas symulacji dla zmieniającej się wartości dokładności realizacji ścieżki. Wykres 6.10 przedstawia zmianę wyglądu ścieżek dla różnych wartości parametru.



Rysunek 6.5. Ścieżka i sterowania dla nominalnych parametrów monocykla



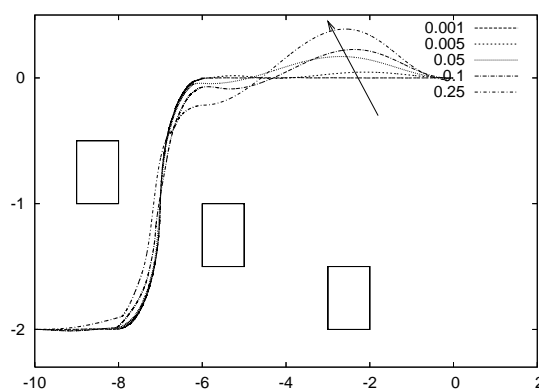
Rysunek 6.6. Ścieżka i sterowania dla nominalnych parametrów dwukołowca

Rysunek 6.7. Zbiór konfiguracji docelowych dla funkcji  $d_1(q)$  i  $d_2(q)$ Rysunek 6.8. Zbiór konfiguracji docelowych dla funkcji  $d_4(q)$ , dwa warianty

Rysunek 6.9. Zbiór konfiguracji docelowych dla manipulatora z brzegiem wewnętrznym

Tablica 6.1. Wpływ dokładności realizacji ścieżki na charakterystyki planera

dokładność realizacji	bez manipulatora		dwuwahadło		dwuwahadło z obrotem	
	mono	duo	mono	duo	mono	duo
liczba elementów ścieżki						
0.001	51	56	62	65	64	67
0.005	43	47	51	52	54	55
0.05	18	18	18	19	19	19
0.1	12	12	12	12	12	13
0.25	7	7	7	7	8	8
energia sterowań						
0.001	25.9	30.0	23.4	25.7	24.5	26.8
0.005	39.6	39.9	28.6	31.1	29.7	32.2
0.05	45.7	57.1	34.3	39.0	36.5	40.7
0.1	48.6	59.9	38.5	43.4	40.3	45.2
0.25	55.6	65.8	46.1	51.9	48.4	53.9
czas symulacji [s]						
0.001	7.2	8.3	8.3	8.9	10.4	10.7
0.005	7.4	8.2	8.1	8.9	9.1	9.0
0.05	5.3	5.8	5.3	6.8	5.8	5.9
0.1	4.8	4.8	4.7	5.6	5.1	5.2
0.25	4.7	4.7	4.7	5.2	4.9	5.0



Rysunek 6.10. Ścieżka dla monocykla przy zmiennej dokładności realizacji ścieżki

### 6.6.3. Uproszczony sposób wyznaczania dokładności realizacji

Kolejnym testem było sprawdzenie wpływu uproszczonego wyznaczania dokładności realizacji ścieżki na zachowanie algorytmu. Uproszczona metoda zaniedbuje różnice pomiędzy współrzędnymi liniowymi i kątowymi (położenie i orientacja). Programowa realizacja takiej metody jest trochę prostsza od wariantu uwzględniającego różnice. W tablicy 6.2 przedstawiono wpływ metody obliczania dokładności realizacji (normalna, uproszczona) na parametry symulacji. Wykres 6.11 przedstawia symulację monocykla, a 6.12 symulację dwukołowca.

Tablica 6.2. Wpływ uproszczonego sposobu obliczania odległości na charakterystyki planera

liczenie odległości	bez manipulatora		dwuwahadło		dwuwahadło z obrotem	
	mono	duo	mono	duo	mono	duo
liczba elementów ścieżki						
uproszczone	18	18	40	19	42	19
dokładne	18	18	18	19	19	19
energia sterowań						
uproszczone	45.7	57.1	29.6	39.0	31.2	40.7
dokładne	45.7	57.1	34.3	39.0	36.5	40.7
czas symulacji [s]						
uproszczone	5.0	5.1	8.9	5.0	7.5	5.0
dokładne	4.9	5.2	4.8	5.0	6.3	6.6

### 6.6.4. Liczba punktów ścieżki testowej

Przebadano wpływ liczby punktów dyskretyzujących ścieżkę testową na algorytm planowania ścieżki. Liczba punktów była zmieniana od 25 do 500 w pięciu krokach. Wyniki przedstawiono w tablicy 6.3. Ścieżka układu została zobrazowana na rysunku 6.13.

Zwiększanie liczby punktów dyskretyzującej ścieżki początkowej powoduje wzrost liczby punktów ścieżki wynikowej. Rezultat ten można wytłumaczyć następująco: im więcej punktów dyskretyzujących liczy ścieżka początkowa, tym dalej (do pewnego stopnia) będzie omijać przeszkody. Jeśli punktów jest niewiele, ścieżka jest wyciągnięta, mniej powyginana i do jej realizacji potrzeba mniej punktów w ścieżce wynikowej. Energia sterowań i czas symulacji również rosną wraz ze zwiększaniem długości ścieżki testowej.

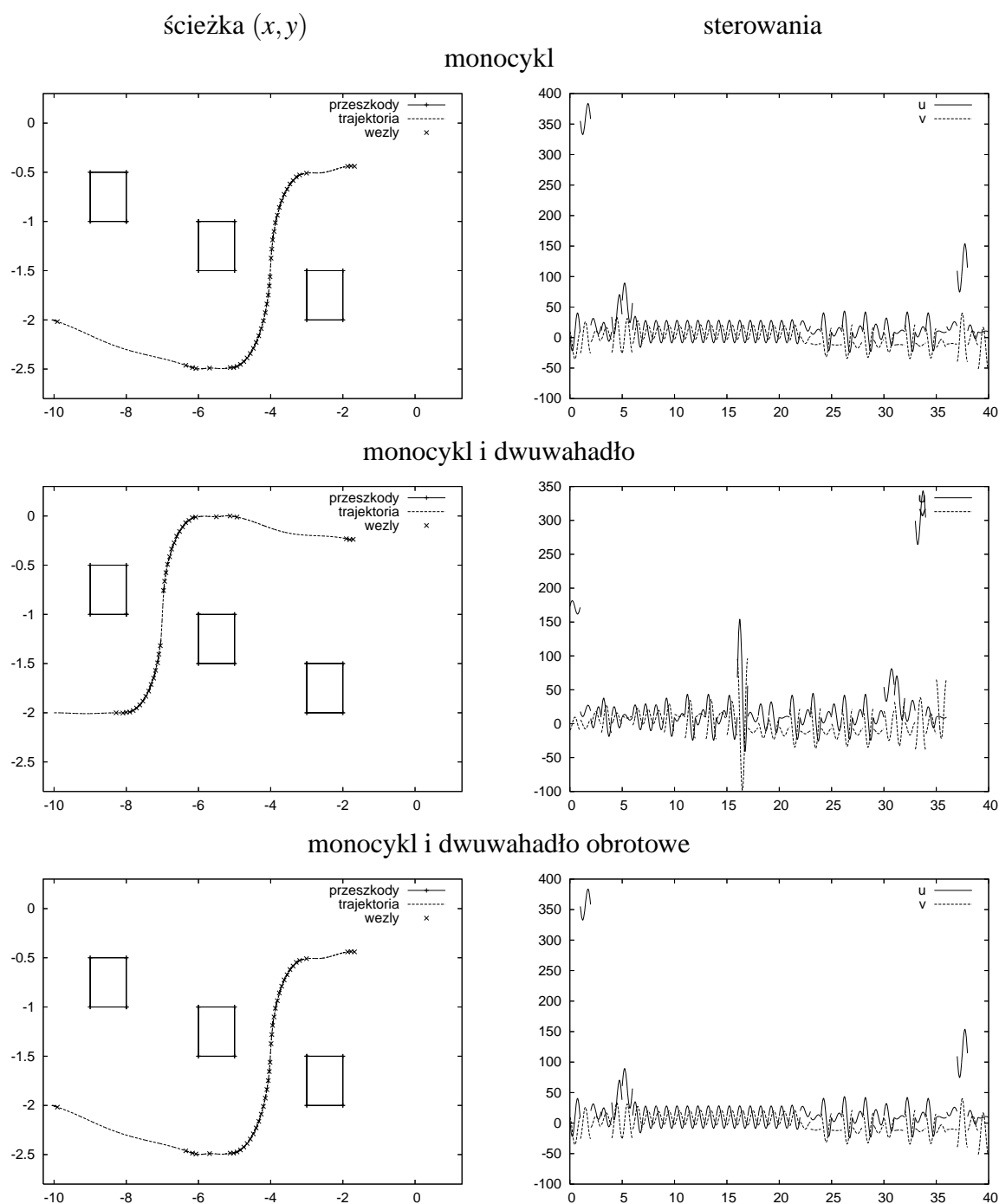
### 6.6.5. Współczynniki wykorzystywane w algorytmie elastycznej wstęgi

Przebadano wpływ współczynników algorytmu elastycznej wstęgi (współczynnika odpychania, zasięgu oddziaływania odpychającego oraz współczynnika ściągania) na zachowanie algorytmu planowania.

#### Współczynnik odpychania

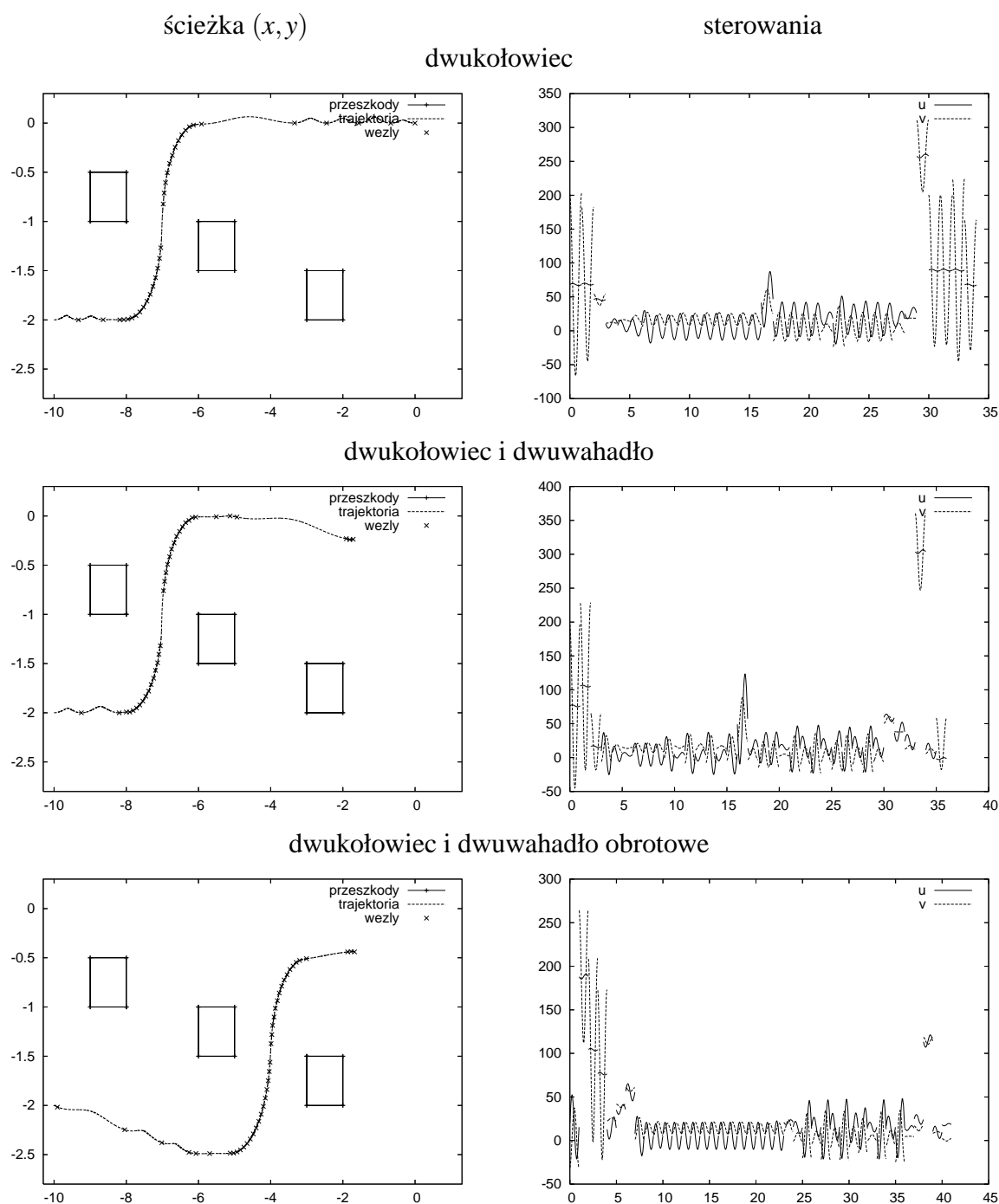
Dla celów testowych przyjęto cztery wartości współczynnika odpychania od przeszkód  $\xi_o$  od  $10^{-4}$  do  $10^{-1}$ . Wyniki przedstawiono w tablicy 6.4. Ścieżka została uwidoczniona na rysunku 6.14.

Z danych symulacyjnych wynika, że zwiększanie współczynnika odpychania powoduje wzrost liczby punktów wynikowej ścieżki, zwiększenie energii potrzebnej na realizację sterowań oraz wydłużenie symulacji. Zbyt duża wartość współczynnika daje w efekcie ścieżki omijające przeszkody jak najdalej, które niekoniecznie są najkrótsze.



Rysunek 6.11. Uprozczone wyliczanie dokładności realizacji — monocykl

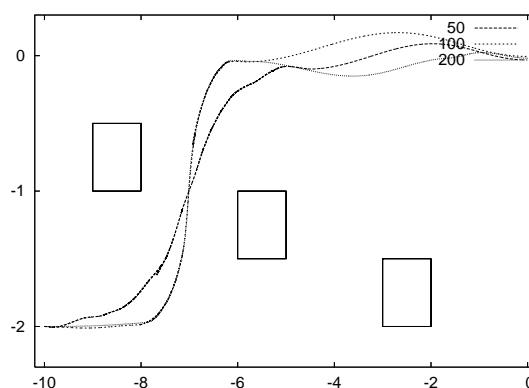




Rysunek 6.12. Uprozczone wyliczanie dokładności realizacji — dwukołowiec

Tablica 6.3. Wpływ liczby punktów dyskretyzujących ścieżkę na charakterystyki planera

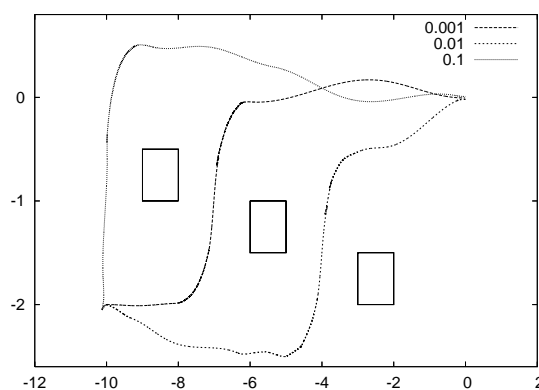
liczba punktów $N$	bez manipulatora		dwuwahadło		dwuwahadło z obrotem	
	mono	duo	mono	duo	mono	duo
końcowa liczba elementów ścieżki						
25	12	12	12	12	13	13
50	13	15	15	16	16	17
100	18	18	18	19	19	19
200	17	17	17	17	18	18
500	20	20	18	18	19	19
energia sterowań						
25	18.7	23.8	15.9	18.6	15.8	17.9
50	29.2	33.5	23.1	25.3	23.0	25.4
100	45.7	57.1	34.3	39.0	36.5	40.7
200	58.3	72.0	46.8	52.9	47.3	53.4
500	81.1	89.5	73.0	74.8	74.8	77.0
czas symulacji [s]						
25	1.4	1.4	1.4	1.4	1.7	1.6
50	3.0	3.4	2.9	3.1	3.4	3.8
100	4.9	5.2	4.8	5.0	6.3	6.6
200	9.0	9.1	8.9	8.9	10.6	11.1
500	22.5	22.3	22.4	22.3	27.6	28.7



Rysunek 6.13. Wykres ścieżki monocykla dla różnych liczb elementów ścieżki testowej

Tablica 6.4. Wpływ współczynnika odpychania na charakterystyki planera

współczynnik odpychania $\xi_o$	bez manipulatora		dwuwahadło		dwuwahadło z obrotem	
	mono	duo	mono	duo	mono	duo
liczba elementów ścieżki						
0.0001	12	12	13	13	13	13
0.001	18	18	18	19	19	19
0.01	17	18	18	18	18	18
0.1	19	20	21	22	22	23
energia sterowań						
0.0001	34.3	43.7	24.8	28.5	23.5	27.2
0.001	45.7	57.1	34.3	39.0	36.5	40.7
0.01	45.9	55.3	38.2	41.7	39.7	42.1
0.1	72.7	79.8	69.8	69.0	75.9	77.6
czas symulacji [s]						
0.0001	3.7	3.7	4.4	4.4	5.7	5.1
0.001	4.9	5.2	4.8	5.0	6.3	6.6
0.01	4.9	4.9	4.8	4.8	6.3	5.8
0.1	5.0	5.1	5.1	5.3	6.3	6.3



Rysunek 6.14. Zmiana współczynnika odpychania — ścieżka monocykla

### Zasięg oddziaływania

Test na wpływ zasięgu odpychania  $\delta_o$  ścieżki od przeszkód został wykonany dla wartości od 0.1 do 2 w czterech krokach. Wyniki przedstawione zostały w tablicy 6.5.

Tablica 6.5. Wpływ zasięgu odpychania elastycznej wstęgi na charakterystyki planera

zasięg odpychania $\delta_o$	bez manipulatora		dwuwahadło		dwuwahadło z obrotem	
	mono	duo	mono	duo	mono	duo
liczba elementów ścieżki						
0.1	5	5	4	4	4	4
0.5	8	8	7	7	8	8
1	18	18	18	19	19	19
2	23	23	27	27	29	28
energia sterowań						
0.1	59.6	72.6	43.7	57.5	46.9	62.9
0.5	43.6	57.3	31.8	39.6	36.9	45.3
1	45.7	57.1	34.3	39.0	36.5	40.7
2	36.1	41.9	38.4	41.1	37.0	39.8
czas symulacji [s]						
0.1	1.4	1.5	1.6	1.7	1.4	1.5
0.5	4.4	4.4	4.5	4.3	6.0	6.0
1	4.9	5.2	4.8	5.0	6.3	6.6
2	5.2	5.5	5.4	5.3	7.4	7.5

Zwiększanie zasięgu odpychania powoduje znaczący wzrost liczby punktów ścieżki wynikowej oraz czasu symulacji. Natomiast energia sterowań maleje. Powiększenie zasięgu oddziaływania może spowodować niezdolność algorytmu do przejścia pomiędzy gęsto ustawionymi przeszkodami (w przeprowadzonych symulacjach jest to sytuacja dla wartości 2).

### Współczynnik ściągania

Zmiany współczynnika ściągania zostały przeprowadzone dla trzech wartości od 0.05 do 0.5. Wyniki przedstawiono w tablicy 6.6

Można zauważyć, iż zwiększanie wartości współczynnika ściągania powoduje zmniejszanie długości ścieżki. Nie można jednak zwiększać wartości tego współczynnika dowolnie, gdyż dla zbyt dużych wartości (oraz zbyt małej liczby punktów tworzących ścieżkę) ścieżka może stać się kolizyjna. Wpływ współczynnika ściągania na energię oraz czas symulacji nie jest jednoznaczny.

#### 6.6.6. Współczynniki optymalizatora nieholonomicznego

Przebadano wpływ zmian współczynników optymalizatora nieholonomicznego: współczynnika wzmocnienia ścieżki nieholonomicznej w zakresie 0.01 do 0.9 — przedstawione w tablicy 6.7 i na wykresie 6.15 oraz elementarnego kroku zmian w zakresie 0.01 do 0.7 przedstawione w tablicy 6.8 i na wykresie 6.16.

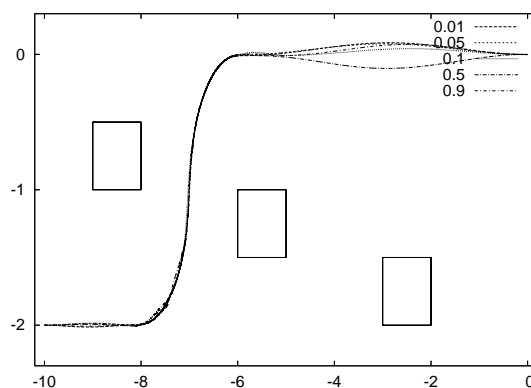
Na podstawie symulacji można wywnioskować, że współczynnik wzmocnienia posiada niewielki wpływ na parametry ścieżki. Jedynie dla kosztu nieholonomicznego daje się zauważyć pewne minimum dla wartości 0.5. Krok zmian nieholonomicznych także jest wielkością, którą można dobrać w miarę dowolnie, bez znaczących różnic na wygląd ścieżki.

Tablica 6.6. Wpływ współczynnika ściągania algorytmu elastycznej wstęgi na charakterystyki planera

współczynnik ściągania $\xi_e$	bez manipulatora		dwuwahadło		dwuwahadło z obrotem	
	mono	duo	mono	duo	mono	duo
liczba elementów ścieżki						
0.05	19	19	18	19	19	20
0.1	18	19	18	18	19	20
0.5	18	18	18	18	19	19
energia sterowań						
0.05	38.1	43.5	32.2	34.3	43.7	42.4
0.1	41.2	48.5	33.4	36.5	41.6	42.3
0.5	45.7	57.1	34.3	39.0	36.5	40.7
czas symulacji [s]						
0.05	5.1	4.9	4.9	4.8	7.0	7.3
0.1	5.0	5.1	4.8	4.9	6.9	7.4
0.5	4.9	5.2	4.8	5.0	6.3	6.6

Tablica 6.7. Wpływ współczynnika wzmocnienia optymalizatora nieholonomicznego na charakterystyki planera

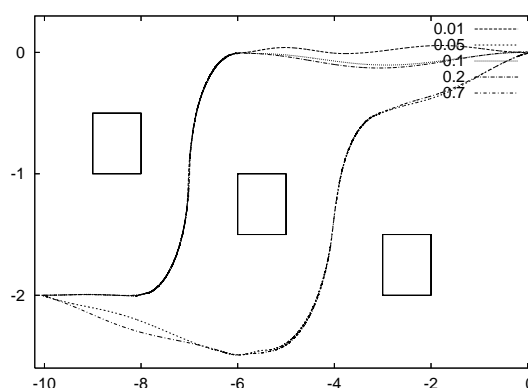
współczynnik wzmocnienia $\xi_{nh}$	bez manipulatora		dwuwahadło		dwuwahadło z obrotem	
	mono	duo	mono	duo	mono	duo
liczba elementów ścieżki						
0.01	38	40.0	41	42	44	45
0.05	37	40	41	42	45	45
0.1	37	40	41	42	45	45
0.5	38	39	41	42	44	46
0.9	36	39	40	43	43	46
energia sterowań						
0.01	42.2	44.2	29.1	32.9	35.6	35.0
0.05	42.7	44.0	29.0	33.1	35.5	34.9
0.1	42.6	44.2	29.1	32.9	35.8	34.9
0.5	42.2	44.2	29.0	32.1	38.1	35.0
0.9	42.1	43.1	31.9	33.9	39.4	34.5
koszt nieholonomiczny						
0.01	136.4	189.8	128.6	158.6	147.9	176.1
0.05	136.1	189.1	128.9	158.8	147.5	176.2
0.1	134.9	188.9	128.2	158.4	146.9	176.3
0.5	133.8	187.5	126.5	156.7	145.6	174.8
0.9	145.7	190.2	161.3	182.4	179.5	198.1
czas symulacji [s]						
0.01	38.8	37.7	37.6	43.1	37.6	42.4
0.05	37.4	37.7	37.5	37.5	37.6	41.4
0.1	37.5	37.8	37.4	37.8	37.9	41.4
0.5	41.6	37.6	37.6	37.9	37.9	38.1
0.9	37.4	38.2	37.4	37.7	38.2	41.6



Rysunek 6.15. Współczynnik wzmocnienia ścieżki nieholonomicznej — ścieżka wynikowa

Tablica 6.8. Wpływ współczynnika kroku zmian planera nieholonomicznego na charakterystyki planera

krok zmian $\xi_d$	bez manipulatora		dwuwahadło		dwuwahadło z obrotem	
	mono	duo	mono	duo	mono	duo
liczba elementów ścieżki						
0.01	38	40	42	43	45	46
0.05	37	39	41	42	45	46
0.1	38	39	41	42	44	46
0.2	37	40	42	42	45	46
0.7	38	40	42	43	45	45
energia sterowań						
0.01	42.1	44.5	28.5	32.1	34.8	34.0
0.05	41.9	43.8	28.5	31.6	36.2	34.4
0.1	42.2	44.2	29.0	32.1	38.1	35.0
0.2	42.6	44.5	28.9	32.3	36.2	35.2
0.7	42.0	44.1	28.8	32.4	35.2	35.1
koszt nieholonomiczny						
0.01	133.3	185.8	123.5	152.9	142.9	170.9
0.05	132.3	185.7	124.2	153.7	143.4	172.8
0.1	133.8	187.5	126.5	156.7	145.6	174.8
0.2	137.3	189.5	129.7	159.4	148.1	177.5
0.7	138.7	191.1	131.4	160.9	150.2	178.9
czas symulacji [s]						
0.01	42.3	52.9	40.6	43.6	43.0	41.0
0.05	41.1	50.3	40.3	43.4	45.9	37.6
0.1	41.1	46.5	40.4	41.5	46.4	44.3
0.2	41.2	40.6	40.6	40.6	45.8	37.9
0.7	41.7	40.3	40.6	44.2	47.9	37.9



Rysunek 6.16. Ścieżki dla różnych wartości współczynnika kroku zmian planera nieholonomicznego

### 6.6.7. Wielokrotne planowanie ścieżki

Zbadano wpływ wielokrotnego wywoływania algorytmu planowania ścieżki. Po każdym etapie planowania ścieżki, realizowany jest tylko następny punkt ścieżki (zamiast dochodzenia aż do punktu końcowego). Od tego punktu następowało kolejne wywołanie algorytmu planowania ścieżki dla pozostałego fragmentu, aż do uzyskania punktu końcowego. Wyniki symulacji zostały przedstawione w tablicy 6.9, a ścieżki na wykresach 6.17 i 6.18.

Tablica 6.9. Wpływ wielokrotnego planowania ścieżki na charakterystyki planera

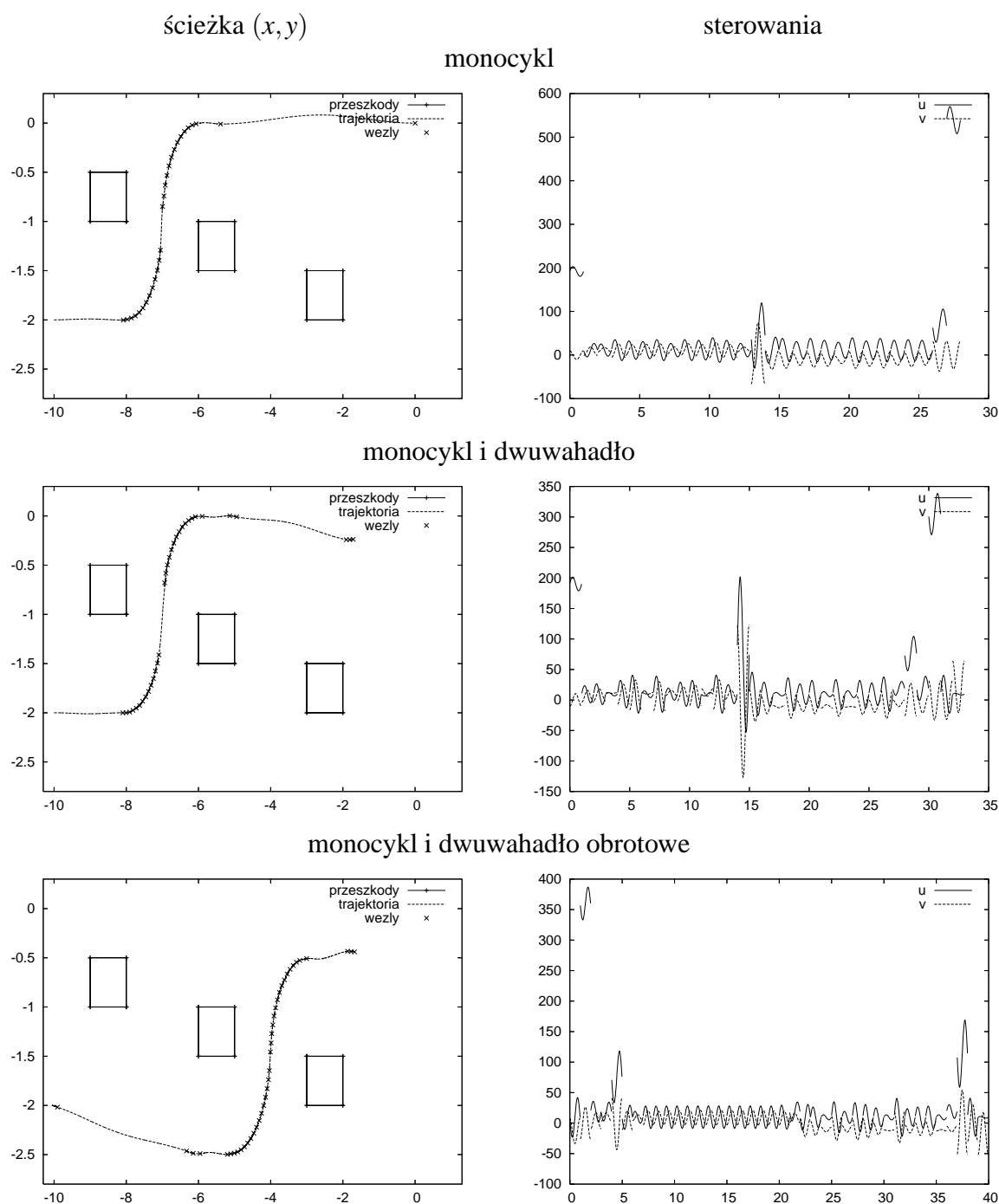
wielokrotne planowanie	bez manipulatora		dwuwahadło		dwuwahadło z obrotem	
	mono	duo	mono	duo	mono	duo
liczba elementów						
włączone	37	40	41	42	43	44
wyłączone	18	18	18	19	19	19
energia sterowań						
włączone	42.5	44.4	29.5	33.7	29.9	33.4
wyłączone	45.7	57.1	34.3	39.0	36.5	40.7
czas symulacji [s]						
włączone	133.5	105.2	100.0	111.4	97.3	93.4
wyłączone	4.9	5.2	4.8	5.0	6.3	6.6

Użycie wielokrotnego planowania ścieżki pozwala na obniżenie energii sterowań potrzebnych do realizacji ścieżki kosztem zwiększenia liczebności elementów ścieżki oraz znacznego zwiększenia czasu planowania.

### 6.6.8. Ograniczenie dopuszczalnej rodziny sterowań

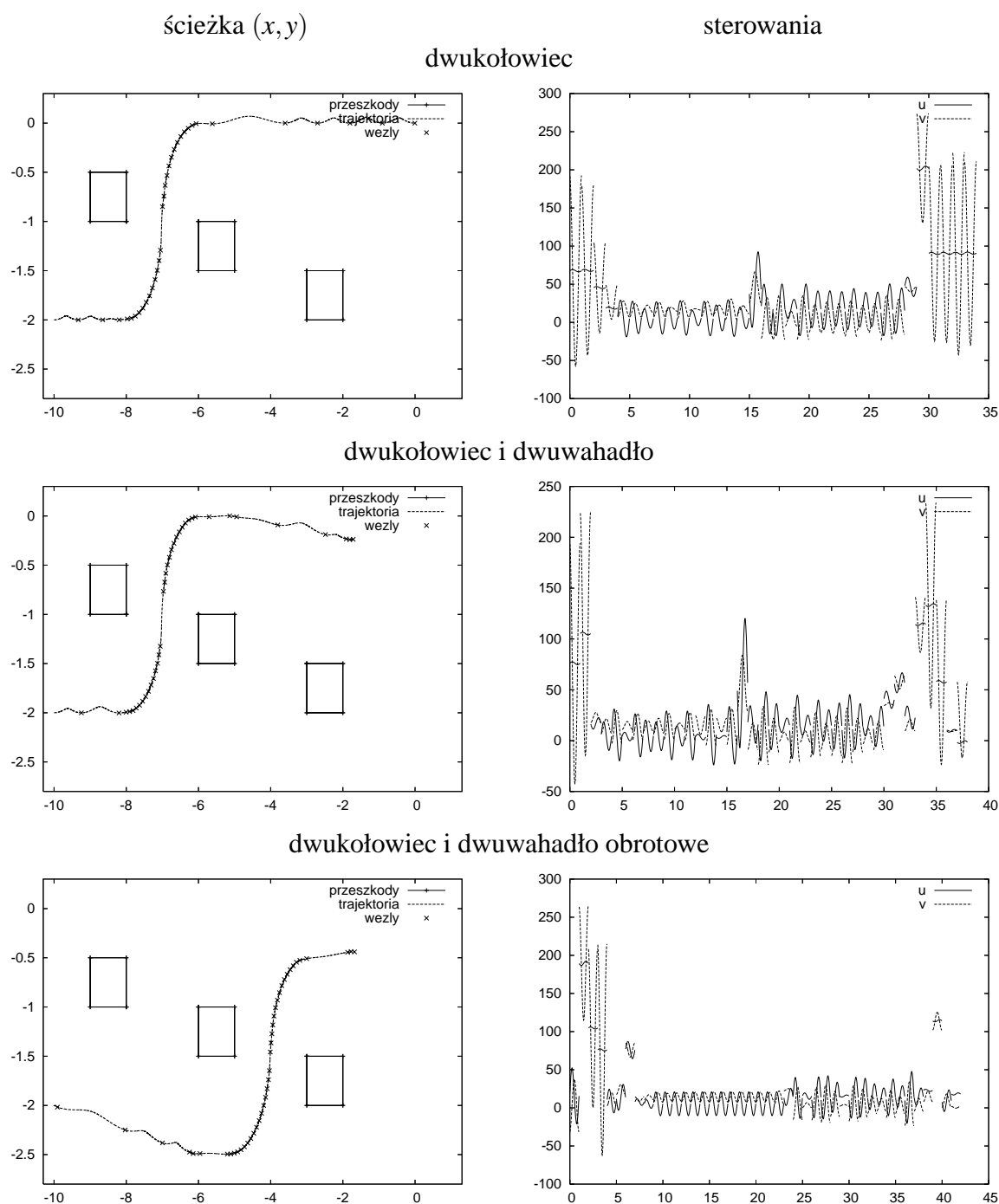
Zbadano wpływ ograniczenia dopuszczalnej rodziny sterowań (6.17), przez przyjęcie dodatkowo wartości współczynnika  $b_0 \leftarrow 0$  występującego w sterowaniu  $v$ . Wyniki zebrane są w tablicy 6.10 oraz na wykresie 6.19.

Wyniki uzasadniają wniosek, że zmniejszenie wymiarowości przestrzeni współczynników sterowań daje w rezultacie wyższą energię realizacji ścieżki. Ograniczenie nie ma znaczącego wpływu na czas wykonywania algorytmu ani na liczbę elementów ścieżki.

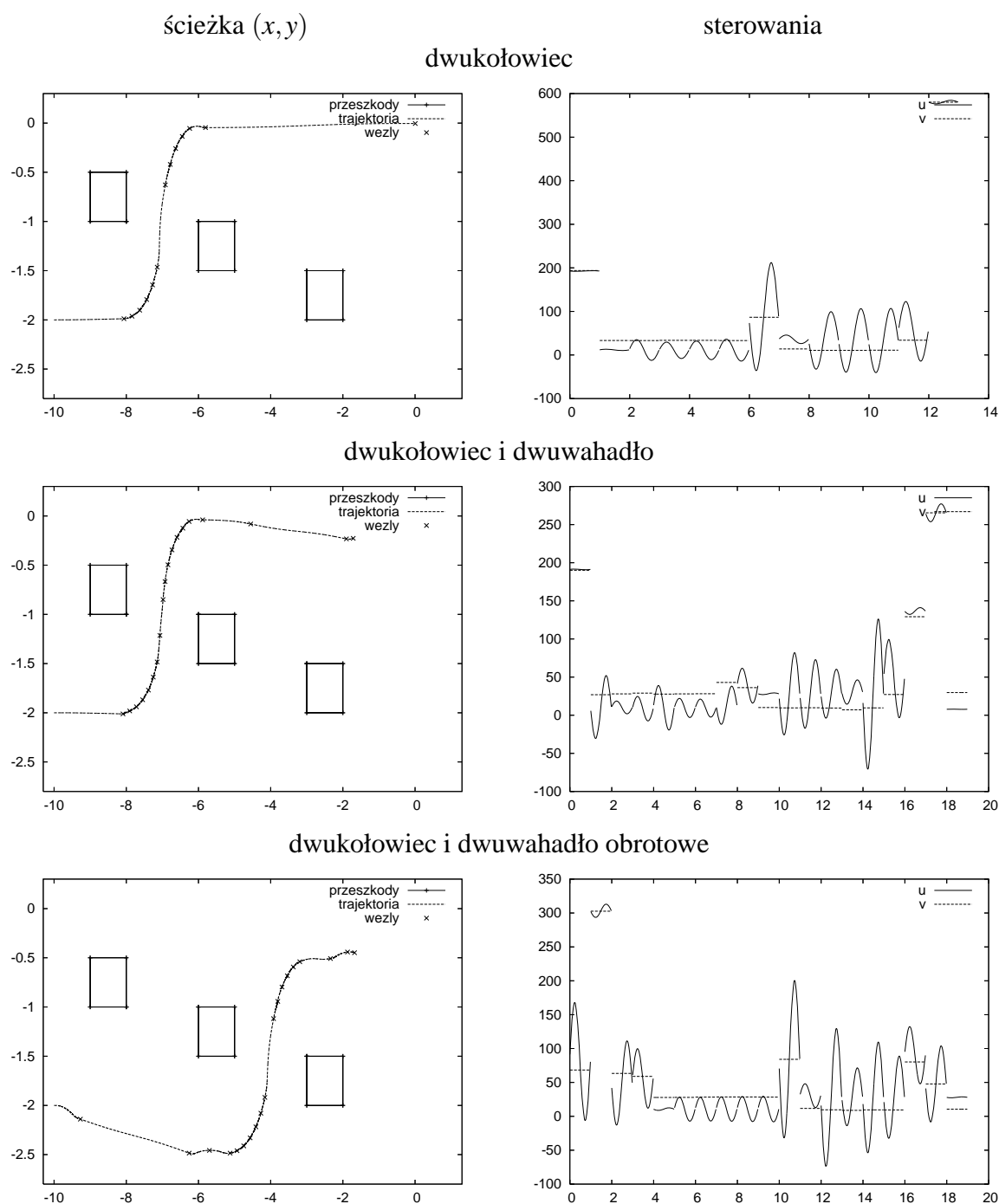


Rysunek 6.17. Wielokrotne planowanie ścieżki — ścieżka i sterowania monocykla





Rysunek 6.18. Wielokrotne planowanie ścieżki — ścieżka i sterowania dwukołowca

Rysunek 6.19. Ograniczone sterowanie  $v$  — ścieżka i sterowania dwukołowca

Tablica 6.10. Wpływ ograniczenia harmonicznego sterowania  $v$  na charakterystyki planera

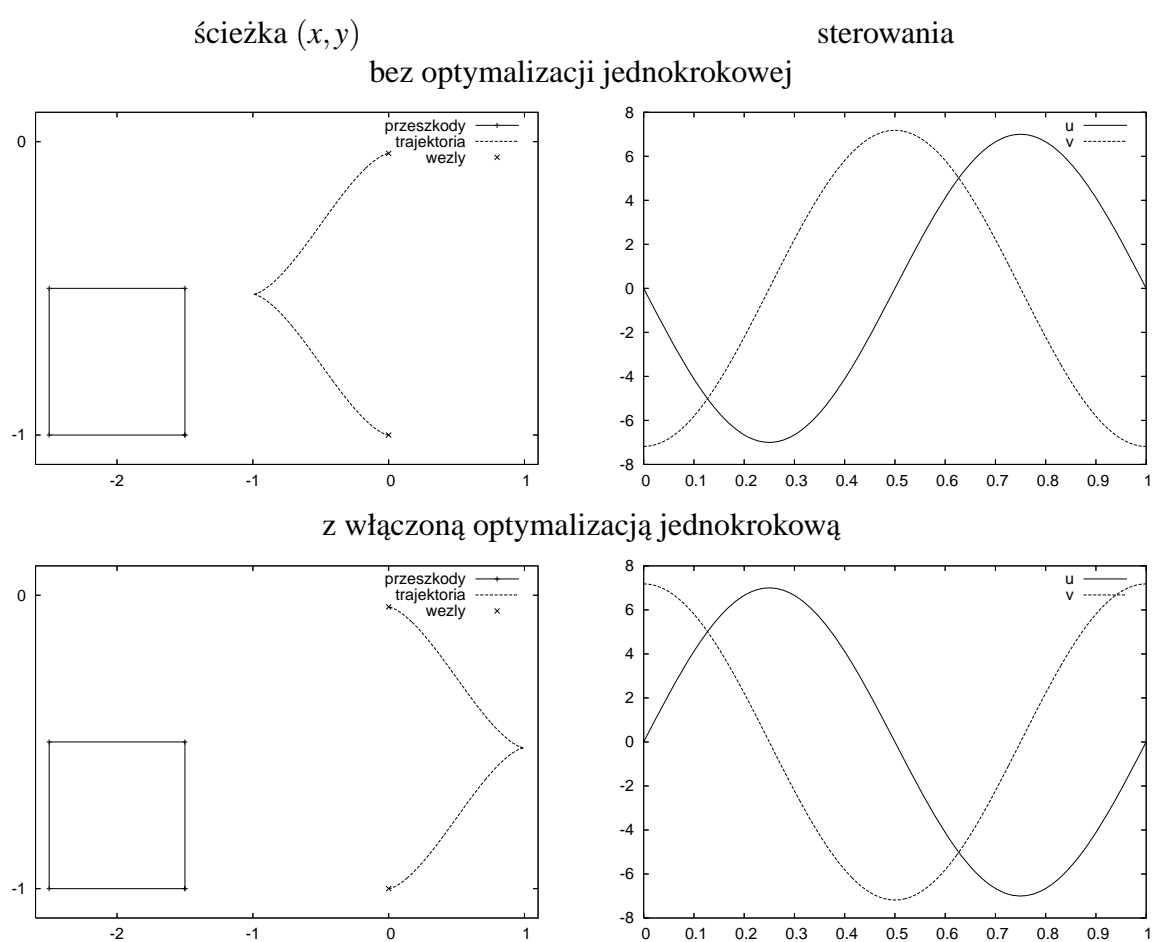
ograniczenie harmonicznego $v$	bez manipulatora		dwuwahadło		dwuwahadło z obrotem	
	mono	duo	mono	duo	mono	duo
liczba elementów ścieżki						
ograniczony	17	18	18	18	18	19
bez ogr.	18	18	18	19	19	19
energia sterowań						
ograniczony	12115.8	61.5	9062.5	56.8	12435.3	77.4
bez ogr.	45.7	57.1	34.3	39.0	36.5	40.7
czas symulacji [s]						
ograniczony	4.9	4.9	4.8	4.8	7.4	5.0
bez ogr.	4.9	5.2	4.8	5.0	6.3	6.6

### 6.6.9. Algorytm optymalizacji odległości od przeszkód

Aby przetestować wpływ algorytmu doboru kąta wstępnego przesunięcia fazowego sterowań na kształt ścieżki, przeprowadzono eksperyment polegający na celowym wymuszeniu planowania ścieżki w niekorzystnym kierunku  $[X, Y]$ . Scena składa się z pojedynczej prostokątnej przeszkody o wierzchołkach w punktach  $(-1.5 - 1.0)$ ,  $(-1.5 - 0.5)$ ,  $(-2.5 - 0.5)$ ,  $(-2.5 - 1.0)$ . Stan początkowy ruchu to  $(0, -1, 0^\circ)$  a stan końcowy to  $(0, 0, 0^\circ)$ . Ścieżka zawiera tylko te dwa punkty, bez stanów pośrednich. Na wykresie 6.20 przedstawiono symulacje dla wariantu bez i z optymalizacją. Symulacje wykonano dla monocykla.

Z wykresu wynika, że przez dobór początkowego przesunięcia fazowego sterowań można znacząco zmieniać kształt ścieżki platformy. Wpływ jest tym bardziej widoczny, im większą rolę w manewrze ruchu platformy odgrywa pole  $[X, Y]$ .

Rysunek 6.20. Wpływ optymalizacji jednokrokowej na charakterystyki planera



## 7. Zakończenie

W dysertacji przedstawiono metody planowania ruchu nieholonomicznego manipulatora mobilnego w środowisku kolizyjnym. Na tle metod literaturowych, dotyczących zarówno poszczególnych komponent układu złożonego z holonomicznego manipulatora umieszczonego na nieholonomicznej platformie jak i ich połączeniu, zaproponowano adaptację lokalnych metod Lie-algebraicznych do rozwiązania zadania planowania. Rozważania przeprowadzono na poziomie kinematycznym, tzn. manipulator w pełni jest zadany przez transformację kinematyki prostej, natomiast platforma mobilna – przez bezdryfowy układ sterowania. Metaforą zaproponowanego podejścia jest koncepcja robota-dźwigu, w myśl której następuje naturalna dekompozycja zadania planowania ruchu układu złożonego na trzy etapy: osiągnięcia bezpiecznej konfiguracji transportowej manipulatora, etap transportowy i etap manipulacyjny w okolicy zadanego punktu docelowego w przestrzeni zadaniowej. W każdym z etapów ruch wykonuje jeden podukład, podczas gdy drugi pozostaje w bezruchu. Wymieniono zalety i wady takiego podejścia. Dla każdego z etapów zaproponowano algorytmy planowania ruchu. Dla etapu pierwszego i końcowego adoptowano klasyczne algorytmy rozwiązania zadania kinematyki odwrotnej z ograniczeniami wynikającymi z istnienia przeszkód w przestrzeni konfiguracyjnej. Podczas wyznaczania konfiguracji transportowej manipulatora zaproponowano, korzystając z przesłanek fizycznych, kilka kryteriów oceny rozkładu siły nacisku wywieranej przez manipulator na punkty styku kół platformy z podłożem oraz postawiono zadanie takiego kształtowania konfiguracji manipulatora, by nacisk spełniał wybrane kryterium jakości optymalnie.

Główne i oryginalne osiągnięcia rozprawy dotyczą etapu przemieszczania platformy. Dla mobilnego etapu planowania ruchu zaproponowano nieholonomiczną miarę jakości ścieżek platformy, adaptację metod Lie-algebraicznych do zadania docelową konfiguracją zadaną jako zbiór pól docelowych oraz miary bezkolizyjności lokalnego manewru wyrażone w abstrakcyjnej przestrzeni pól wektorowych rozpinających przestrzeń stanu wokół bieżącej pozycji platformy. Po etapie planowania bezkolizyjnej ścieżki nieholonomicznej, realizowanej przy pomocy klasycznej metody elastycznej wstęgi, następuje etap poprawy ścieżki z wykorzystaniem wprowadzonej miary nieholonomicznej. Zaproponowano dwa algorytmy optymalizacyjne: lokalnych wariacji oraz lokalnego zaburzenia ścieżki. Działanie obydwu algorytmów polega na zaburzaniu bezkolizyjnej ścieżki bieżącej, ocenie potencjalnych ścieżek i wyborze najlepszej, która staje się bieżącą dla kolejnej iteracji algorytmów. Różnica między algorytmami polega na zakresie dopuszczalnych zaburzeń: w algorytmie lokalnych wariacji zaburzana jest ścieżka na całej swej długości, dla algorytmu lokalnych wariacji – tylko wokół wybranego punktu ścieżki. Etapy planowania ruchu manipulatora mobilnego przetestowano na wybranych układach platform i manipulatorów na różnym poziomie szczegółowości. Etapy dla których istnieją klasyczne metody planowania (osiągnięcie zadanej konfiguracji transportowej manipulatora i osiągnięcie zadanego punktu przestrzeni zadaniowej w etapie trzecim planowania) praktycznie pominięto w testach poświęcając etapowi przemieszczania platformy najwięcej uwagi. W szerokich badaniach symulacyjnych zbadano wpływ parametrów warunkujących działanie zaproponowanych algorytmów.

Wyniki dysertacji mogą być naturalnie rozszerzane w kierunku uwzględnienia dynamiki zarówno manipulatora jak i platformy mobilnej. Kolejnym etapem jest planowanie koordynacji ruchu manipulatora i platformy w fazach przejściowych, gdy, z powodu środowiska z dużą liczbą przeszkód, niemożliwe jest uniknięcie kolizji przez ruch jedynie jednego podukładu manipulatora mobilnego.

## A. Dodatki

Materiał uzupełniający przedstawiono w następującym porządku: w podrozdziale A.1 zaprezentowano klasyczny algorytm Newtona służący do wyznaczania rozwiązań zadania kinematyki odwrotnej, wraz z wersjami umożliwiającymi optymalizację w przestrzeni zerowej. W podrozdziale A.2 przedstawiono algorytmiczne aspekty sprawdzania kolizyjności robota (punktu materialnego) z przeszkodami w przestrzeni wielokątów. W podrozdziale A.3 zaproponowano kilka wskaźników obszerności lokalnego manewru platformy mobilnej w abstrakcyjnej przestrzeni pól wektorowych zlużających do sprawdzania bezkolizyjności ruchu.

### A.1. Algorytm Newtona

Klasyczny algorytm Newtona [59] pozwala na wyznaczenie dla funkcji  $\mathbf{f}(\mathbf{z}) = \mathbf{y}$ , o wymiarze przestrzeni wewnętrznej  $\mathbf{z} \in \mathbf{Z}$  nie mniejszym od wymiaru przestrzeni zadaniowej  $\mathbf{y} \in \mathbf{Y}$ , zadaną wartością  $\mathbf{y}_d$ , jednego z rozwiązań  $\mathbf{z}^*$ , czyli  $\mathbf{f}(\mathbf{z}^*) = \mathbf{y}_d$ , gdy poszukiwania zapoczątkowane są w zadanym punkcie  $\mathbf{z}_0$ . Klasyczna postać algorytmu Newtona zapisana w wersji dyskretnej, gotowej do implementacji komputerowej, jest następująca:

$$\mathbf{z}_{i+1} = \mathbf{z}_i + \xi \cdot \mathbf{J}^\#(\mathbf{z}_i)(\mathbf{y}_d - \mathbf{f}(\mathbf{z}_i)), \quad (\text{A.1})$$

gdzie  $i$  jest licznikiem iteracji zainicjowanym wartością zerową,  $\xi$  jest dodatnim parametrem rzeczywistym warunkującym szybkość zbieżności procesu iteracyjnego,  $\mathbf{J}(\mathbf{z}) = \partial \mathbf{f}(\mathbf{z}) / \partial \mathbf{z}$  jest jacobianem, a  $\mathbf{J}^\# = \mathbf{J}^T (\mathbf{J} \cdot \mathbf{J}^T)^{-1}$  pseudoodwrotnością Moore'a-Penrose'a, wybierającą z możliwych rozwiązań – rozwiązanie o najmniejszej długości euklidesowej. Wersja algorytmu Newtona pozwalająca dodatkowo na optymalizację w przestrzeni zerowej zadana jest zależnością:

$$\mathbf{z}_{i+1} = \mathbf{z}_{i+1} + \xi \cdot \mathbf{J}^\#(\mathbf{z}_i)(\mathbf{y}_d - \mathbf{f}(\mathbf{z}_i)) + (\mathbf{I} - \mathbf{J}^\#(\mathbf{z}_i)\mathbf{J}(\mathbf{z}_i))\mathbf{v}, \quad (\text{A.2})$$

gdzie  $\mathbf{v} \in \mathbb{R}^{\dim \mathbf{Z}}$  jest dowolnym wektorem. Optymalizację w przestrzeni zerowej przeprowadza się w celu minimalizacji dodatkowej funkcji kryterialnej  $d(\mathbf{z})$ . Wtedy  $\mathbf{v}$  (A.2) przyjmuje postać antygradientu:

$$\mathbf{v}(\mathbf{z}) = -\frac{\partial d(\mathbf{z})}{\partial \mathbf{z}}. \quad (\text{A.3})$$

W literaturze robotycznej [59] znane są także wersje algorytmu Newtona z drugorzędną funkcją kryterialną  $\mathbf{f}_2(\mathbf{z})$ , która powinna osiągnąć wartość  $\mathbf{y}_{d2}$ :

$$\begin{aligned} \mathbf{r} &= \mathbf{y}_d - \mathbf{f}(\mathbf{z}) \\ \mathbf{r}_2 &= \mathbf{y}_{d2} - \mathbf{f}_2(\mathbf{z}) \\ \hat{\mathbf{J}}_2 &= \mathbf{J}_2(\mathbf{I} - \mathbf{J}^\# \mathbf{J}) \\ \mathbf{z}_{i+1} &= \mathbf{z}_i + \xi \cdot \mathbf{J}^\# \mathbf{r} + \hat{\mathbf{J}}_2^\# (\mathbf{r}_2 - \mathbf{J}_2 \mathbf{J}^\# \mathbf{r}). \end{aligned} \quad (\text{A.4})$$

## A.2. Sprawdzanie kolizyjności robota z przeszkodami

Przeszkody są reprezentowane dowolnymi wielokątami. W implementacji komputerowej przeszkoda przedstawiona jest jako lista krawędzi. Przeszkoda musi zawierać przynajmniej dwa wierzchołki. Przyjmuje się, że robot reprezentowany jest jako punkt materialny. Podczas wyznaczania wzajemnej relacji robota i przeszkód istotne są tylko współrzędne położeniowe, oznaczone  $\mathbf{q}'$ . Do sprawdzania kolizyjności robota z przeszkodami zaimplementowano funkcje pomocnicze:

- wyznaczenie odległości punktu trajektorii  $\mathbf{q}'_i$  od krawędzi przeszkody  $\mathbf{o}_{j,k}, \mathbf{o}_{j,k+1}$  — jako odległość pomiędzy punktem a odcinkiem:

- (1)  $\vec{\mathbf{v}} \leftarrow \mathbf{o}_{j,k+1} - \mathbf{o}_{j,k}$
- (2)  $\vec{\mathbf{w}} \leftarrow \mathbf{q}'_i - \mathbf{o}_{j,k}$
- (3) jeśli  $\vec{\mathbf{w}} \circ \vec{\mathbf{v}} \leq 0$ :  $odl \leftarrow d(\mathbf{o}_{j,k}, \mathbf{q}'_i)$ , koniec
- (4) jeśli  $|\vec{\mathbf{v}}|^2 \leq \vec{\mathbf{w}} \circ \vec{\mathbf{v}}$ :  $odl \leftarrow d(\mathbf{o}_{j,k+1}, \mathbf{q}'_i)$ , koniec
- (5)  $odl \leftarrow d(\mathbf{o}_{j,k} + \frac{\vec{\mathbf{w}} \circ \vec{\mathbf{v}}}{|\vec{\mathbf{v}}|^2} \vec{\mathbf{v}}, \mathbf{q}'_i)$ ,

gdzie  $d(\mathbf{p}, \mathbf{q}')$  — oznacza odległość pomiędzy punktem  $\mathbf{p}$  a  $\mathbf{q}'$ .

- sprawdzenie, czy odcinek trajektorii  $\mathbf{q}'_i, \mathbf{q}'_{i+1}$  przecina się z krawędzią przeszkody  $\mathbf{o}_{j,k}, \mathbf{o}_{j,k+1}$ . Test polega na sprawdzeniu przecięcia się dwóch odcinków:

- (1)  $\mathbf{a} \leftarrow \mathbf{q}'_i, \mathbf{b} \leftarrow \mathbf{q}'_{i+1}, \mathbf{c} \leftarrow \mathbf{o}_{j,k}, \mathbf{d} \leftarrow \mathbf{o}_{j,k+1}$
- (2)  $m \leftarrow -c_x a_y + d_x a_y + c_x b_y - d_x b_y + a_x c_y - b_x c_y - a_x d_y + b_x d_y$
- (3)  $t_1 \leftarrow \frac{1}{m} (-c_x a_y + d_x a_y + a_x c_y - d_x c_y - a_x d_y + c_x d_y)$
- (4)  $t_2 \leftarrow \frac{1}{m} (b_x a_y - c_x a_y - a_x b_y + c_x b_y + a_x c_y - b_x c_y)$
- (5) jeśli  $t_1 \in [0, 1] \wedge t_2 \in [0, 1]$  wtedy odcinki się przecinają

gdzie indeksy dolne  $x, y$  oznaczają odpowiednio współrzędne położeniowe wektorów.

Wyznaczanie odległości punktu od przeszkody jest wykorzystywane w momencie wybierania najlepszego wariantu sterowań dla realizacji określonego odcinka ścieżki  $\mathbf{q}_i, \mathbf{q}_{i+1}$

Sprawdzanie przecięcia trajektorii z przeszkodami używane jest do weryfikacji, czy ścieżka otrzymana z planera holonomicznego jest bezkolizyjna.

## A.3. Obszerność manewru

Zależności (6.19) i (2.6) opisują zmianę stanu układu po zastosowaniu sterowania na horyzoncie czasu  $[0, T]$ . Z punktu widzenia zbieżności metody Lie-algebraicznej istotny jest tylko stan  $\mathbf{q}(T)$ . Nie jest to charakterystyka wystarczająca w środowiskach kolizyjnych, gdzie jest konieczna informacja o przebiegu pełnej trajektorii dla  $t \in [0, T]$ . Dzięki znajomości trajektorii możliwy jest dobór parametrów sterowań zapewniający odpowiednio bezpieczny manewr. Aby rozważyć trajektorie układu należy uzmiennić górną granicę całkowania w (2.8) (podstawić  $\tau$  za  $T$ ) i określić współczynniki  $\alpha$  dla

$\tau \in [0, T]$ : Przyjmując  $\tau = v \cdot T, v \in [0, 1]$ , współczynniki są bardziej skomplikowane niż (6.19)

$$\begin{aligned}\alpha_X(v) &= \frac{\sqrt{T}}{\pi} \left( a_0 v \pi + \sqrt{2} A \sin(v\pi) \sin(v\pi + \varphi) \right), \\ \alpha_Y(v) &= \frac{\sqrt{T}}{\pi} \left( b_0 v \pi + \sqrt{2} B \sin(v\pi) \sin(v\pi + \psi) \right), \\ \alpha_{[X,Y]}(v) &= \frac{T}{8\pi^2} \left( 2\sqrt{2} b_0 A v \pi \cos \varphi + 2\sqrt{2} b_0 A v \pi \cos(\varphi + 2v\pi) \right. \\ &\quad - AB \cos(\varphi - 2v\pi - \psi) + AB \cos(\varphi + 2v\pi - \psi) - 2\sqrt{2} a_0 B v \pi \cos \psi \\ &\quad - 2\sqrt{2} a_0 B v \pi \cos(2v\pi + \psi) + 2\sqrt{2} A b_0 \sin \varphi - 2\sqrt{2} A b_0 \sin(\varphi + 2v\pi) \\ &\quad \left. + 4AB v \pi \sin(\varphi - \psi) - 2\sqrt{2} a_0 B \sin \psi + 2\sqrt{2} a_0 B \sin(2v\pi + \psi) \right). \quad (\text{A.5})\end{aligned}$$

W zależności od planowanego manewru, czyli pożądanego stanu końcowego  $q(T)$ , wyróżniono trzy charakterystyczne typy ruchów:

1. tylko w kierunkach pierwszej warstwy

$$\alpha_{[X,Y]} \equiv 0 \implies A = B = 0 \quad \text{i} \quad \alpha_X = \sqrt{T} a_0, \quad \alpha_Y = \sqrt{T} b_0,$$

2. tylko w kierunkach drugiej warstwy

$$\alpha_X \equiv 0 \wedge \alpha_Y \equiv 0 \implies a_0 = b_0 = 0,$$

$$\alpha_{[X,Y]} = \frac{T}{2\pi} AB \sin(\varphi - \psi),$$

3. ruch mieszany w kierunku jednego pola pierwszej warstwy i pola drugiej warstwy.

$$\alpha_Y \equiv 0 \implies b_0 = 0, \quad \alpha_X = \sqrt{T} a_0,$$

$$\alpha_{[X,Y]} = \frac{T}{2\pi} B \left( -\sqrt{2} a_0 \cos \psi + A \sin(\varphi - \psi) \right).$$

Analiza przytoczonych wariantów prowadzi do wniosku, że istotne jest jedynie zachowanie stałej wartości  $\varphi - \psi$ . Aby zmaksymalizować długość ruchu w kierunku  $[X, Y]$  należy przyjąć  $|\sin(\varphi - \psi)| = 1$ .

Dla ustalonego  $T$  (dla uproszczenia zapisu konsekwentnie ten parametr jest dalej pomijany), grupując parametry  $\alpha(v) = (\alpha_X(v), \alpha_Y(v), \alpha_{[X,Y]}(v))^T$ , wyodrębniamy dwa stany parametrów  $\alpha$  — przed i po zastosowaniu sterowania  $u(\cdot), v(\cdot)$ :  $\alpha_0 = \alpha(0) \equiv \mathbf{0}, \alpha_1 = \alpha(1)$ . W celu ułatwienia analizy kształtu trajektorii układu między stanami  $\alpha_0$  i  $\alpha_1$  rzutujemy trajektorię z abstrakcyjnej przestrzeni pól wektorowych  $X, Y, [X, Y]$  na płaszczyznę. Parametry rzutowania są tak dobrane, by rzutem punktu  $\alpha_1$  był  $\alpha_0$  (por. rys. A.1). Rzut trajektorii staje się krzywą zamkniętą i możliwa jest obserwacja obszerności manewru.

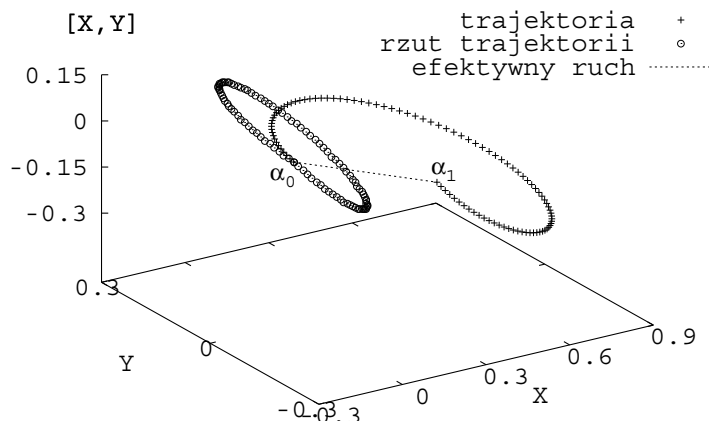
Lokalna trajektoria we współrzędnych  $\alpha$  kreśli na płaszczyźnie  $\pi$  krzywą  $(\eta(\cdot), \xi(\cdot))$ . Jej analiza daje pogląd na kształt trajektorii, a przede wszystkim pozwala na znalezienie regularnych brył zawierających trajektorię lokalną (prostokąty, walec).

### A.3.1. Wskaźniki obszerności manewru

Do analizy obszerności manewru proponujemy kilka wskaźników:

- prostokąt ograniczający o przeciwległych wierzchołkach w punktach  $(\min_{v \in [0,1]} \eta(v), \min_{v \in [0,1]} \xi(v))$  i  $(\max_{v \in [0,1]} \eta(v), \max_{v \in [0,1]} \xi(v))$ , określa podstawę prostopadłościanu (o wysokości  $\alpha_1$ ) zawierającego trajektorię lokalną;





Rysunek A.1. Idea rzutowania

— promień walca o osi wzdłuż normalnej  $\vec{n}$ , w którym zawarta jest trajektoria lokalna

$$R = \max_{v \in [0,1]} \sqrt{\eta(v)^2 + \xi(v)^2},$$

— obliczając “odległość” rzutu punktów  $(\xi(v), \eta(v))$  od prostej przechodzącej przez punkt  $(0,0)$  i kącie nachylenia  $\theta$  (względem wektora zwróconego w kierunku  $\xi()$ ),

$$D(v, \theta) = \eta(v) \sin \theta - \xi(v) \cos \theta \quad (\text{A.6})$$

można wyznaczyć szerokość rzutu

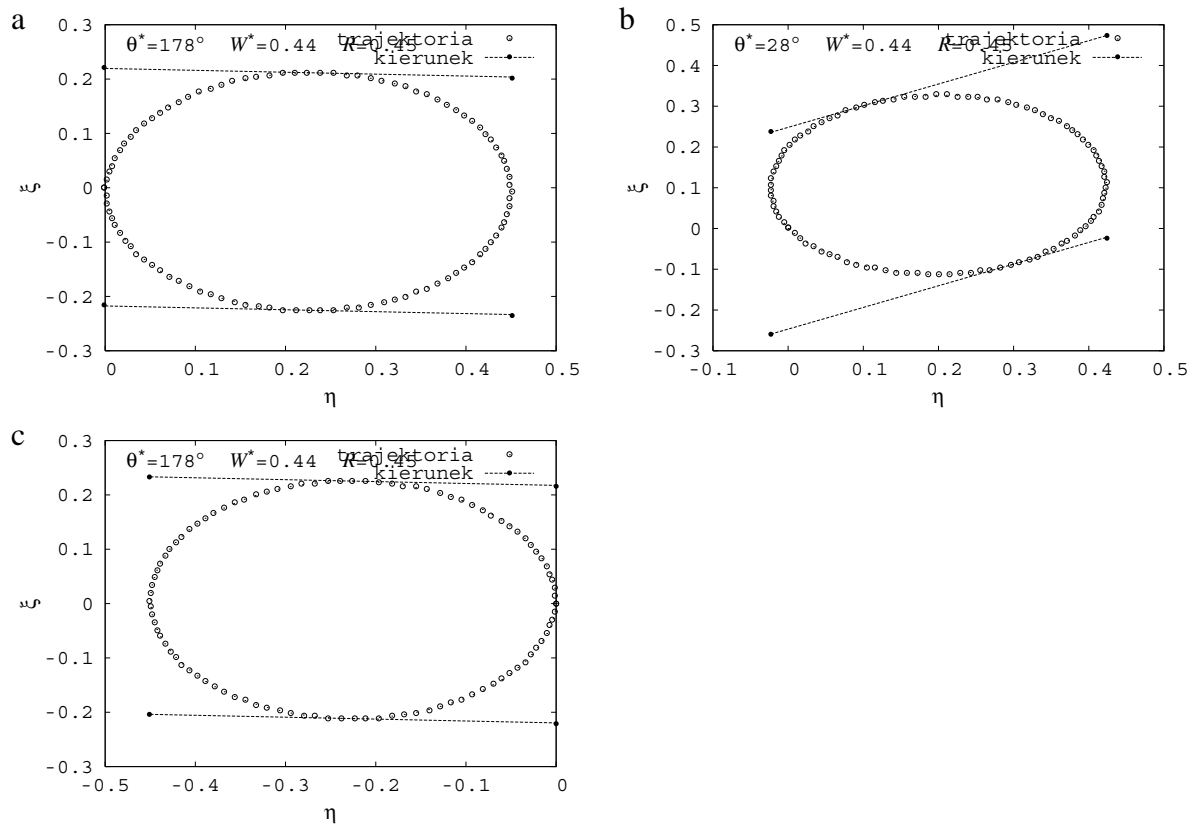
$$W(\theta) = \left| \max_{v \in [0,1]} D(v, \theta) - \min_{v \in [0,1]} D(v, \theta) \right|,$$

oraz kąt  $\theta^*$  wzdłuż którego rozłożone są punkty realizujące minimalną szerokość rzutu  $W^*$ :

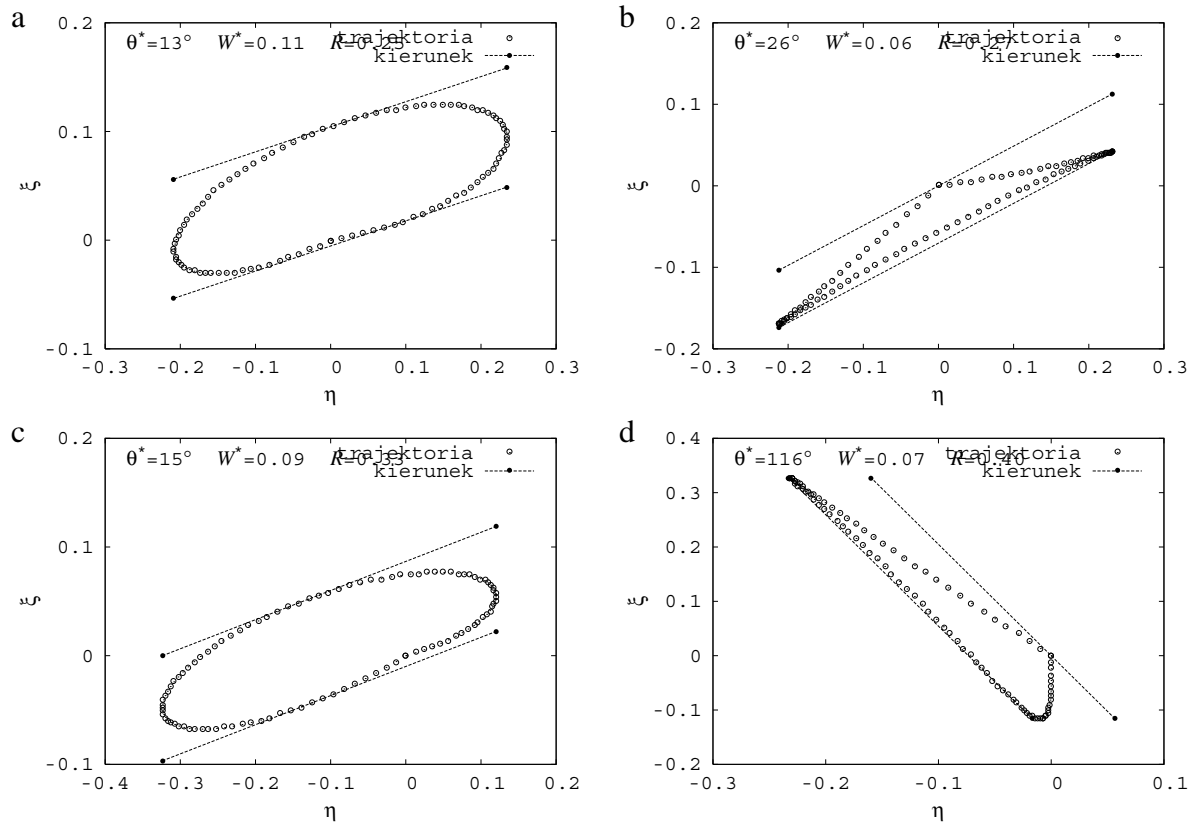
$$\theta^* = \arg \min_{\theta} W(\theta), \quad W^* = W(\theta^*).$$

Niewielka wartość wskaźnika  $W^*$  w porównaniu do  $R$  świadczy o wąskim i wydłużonym rzucie. Wielkość  $D(p, \theta)$  ze wzoru (A.6) jest dodatnia lub ujemna, w zależności od strony, po której leży punkt względem prostej.

Na rysunkach A.2 i A.3 przedstawiono ilustracje ścieżki oraz wartości liczbowe wskaźników dla przypadku ruchu w kierunku pola  $[X, Y]$  i  $X$  wraz z  $[X, Y]$ .



Rysunek A.2. Ruch w kierunku  $[X, Y]$ : a –  $A = 1, B = 1, \varphi = 0^\circ, \psi = 90^\circ$ ,  
b –  $A = 1, B = 1, \varphi = 30^\circ, \psi = 120^\circ$ , c –  $A = 1, B = 1, \varphi = 90^\circ, \psi = 180^\circ$



Rysunek A.3. Ruch w kierunku  $X$  i  $[X, Y]$ : a –  $a_0 = 0.5, A = 1, B = 1, \varphi = 0^\circ, \psi = 90^\circ$ ,  
b –  $a_0 = 1.5, A = 1, B = 1, \varphi = 0^\circ, \psi = 90^\circ$ , c –  $a_0 = 0.5, A = 1, B = 1, \varphi = 30^\circ, \psi = 120^\circ$ ,  
d –  $a_0 = 1.5, A = 1, B = 1, \varphi = 30^\circ, \psi = 120^\circ$

## Bibliografia

- [1] B. Bayle, J.-Y. Fourquet, M. Renaud. Manipulability of wheeled mobile manipulators: Application to motion generation. *Int. Jour. Rob. Res.*, 22(7–8):565–581, 2003.
- [2] O. Brock, O. Khatib. Elastic strips: A framework for motion generation in human environments. *Int. Jour. of Robotics Research*, 21(12):1031–1052, 2002.
- [3] R. Brockett. Asymptotic stability and feedback stabilization. *Differential geometry control theory*, ss. 181–191, 1983.
- [4] W. Chow. Über Systeme von linearen partiellen Differentialgleichungen erster Ordnung. *Math. Ann.*, 117(1):98–105, 1939.
- [5] C. Connolly, J. Burns. Path planning using laplace’s equation. *Int. Conf. on Robotics and Automation*, tom 3, ss. 2102–2106, Cincinnati, 1990.
- [6] B. Curto, V. Moreno, F. Blanco. A general method for c-space evaluation and its application to articulated robots. *IEEE Trans. on Rob. and Aut.*, 18(1):24–31, 2002.
- [7] A. De Luca, G. Oriolo. Local incremental planning for nonholonomic mobile robots. *Int. Conf. Robotics and Automation*, ss. 104–110, 1994.
- [8] P. Deufhard. *Newton Methods for Nonlinear Problems*. Springer-Verlag, Berlin, 2004.
- [9] A. Divelbiss, J. Wen. Nonholonomic path planning with inequality constraints. *IEEE Int. Conf. on Robotics and Automation*, tom 1, ss. 52–57, 1994.
- [10] L. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journ. of Mathematics*, 79:497–516, 1957.
- [11] I. Duleba. Checking controllability of nonholonomic systems via optimal generation of Ph. Hall basis. *IFAC Symp. on Robot Control*, ss. 485–490, Nantes, 1997.
- [12] I. Duleba. *Algorithms of motion planning for nonholonomic robots*. Oficyna Wyd. Polit. Wrocławskiej, 1998.
- [13] I. Duleba. On a computationally simple form of the generalized Campbell-Baker-Hausdorff-Dynkin formula. *System and Control Letters*, 34:191–202, 1998.
- [14] I. Duleba. On use of Campbell-Baker-Hausdorff-Dynkin formulas in nonholonomic motion planning. *Proc. of the Robot Motion Control Conf.*, ss. 177–182, Kiekrz, 1999.
- [15] I. Duleba. Lie-bracket based dexterity measure for driftless nonholonomic systems. *Proc. 14th MTNS Symposium*, Perpingnan, 2000. Cd-rom.
- [16] I. Duleba. Small radius attainability spheres for driftless non-holonomic systems. *Proc. IEEE Conf. Robotics and Autom.*, ss. 1836–1841, 2000.
- [17] I. Duleba. Uciąganie sterowań generowanych przez lokalne metody lie-algebraiczne. *Postępy robotyki*, ss. 111–118, Piechowice, 2006.
- [18] I. Duleba, W. Khefifi. Pre-control form of the generalized Campbell-Baker-Hausdorff-Dynkin formula for affine nonholonomic systems. *Syst. Contr. Lett.*, 55(2):146–157, 2006.
- [19] I. Duleba, P. Ludwików. Bases for local nonholonomic motion planning. K. Kozłowski, redaktor, *Robot Motion and Control*, Lecture Notes in Control and Information Sciences, ss. 73–82. Springer, 2006.
- [20] I. Duleba, P. Ludwików. Cheap Paths for Nonholonomic Motion Planning. *SyRoCo*, Bolonia, 2006. Cd-rom.
- [21] I. Duleba, P. Ludwików. Lie-algebraic Evaluation of Paths for Nonholonomic Motion Planning. *Int. Conf. on Artificial Intelligence and Soft Computing*, ss. 383–388, Zakopane, 2006.
- [22] I. Duleba, P. Ludwików. Local Variation Method to Determine Cheap Paths for Nonholonomic Systems. *RoManSy*, ss. 139–146, Warszawa, 2006.

- [23] I. Duleba, J. Sasiadek. Nonholonomic motion planning based on Newton algorithm with energy optimization. *IEEE Trans. Contr. Syst. Technology*, 11(3):355–363, 2003.
- [24] P. Ferbach. A method of progressive constraints for nonholonomic motion planning. *IEEE Int. Conf. on Robotics and Automation*, tom 4, ss. 2949–2955, Minneapolis, 1996.
- [25] P. Ferbach, J.-F. Rit. Planning nonholonomic motions for manipulated objects. *IEEE Int. Conf. on Robotics and Automation*, ss. 2935–2942, Minneapolis, 1996.
- [26] C. Fernandes, L. Gurvits, Z. Li. Near-optimal nonholonomic motion planning for a system of coupled rigid bodies. *IEEE Trans. On Automatic Control*, 39(3):450–463, 1994.
- [27] M. Fruchard, P. Morin, C. Samson. A framework for the control of nonholonomic mobile manipulators. *Int. Journ. of Robotics Research*, 25(8):745–780, 2006.
- [28] M. Galicki. *Wybrane metody planowania optymalnych trajektorii robotów manipulacyjnych*. WNT, Warszawa, 2000.
- [29] M. Galicki. Inverse kinematics solution to mobile manipulators. *Int. Journ. of Robotic Research*, 22(12):1041–1064, 2003.
- [30] J. Hopcroft, J. Schwartz, M. Sharir. *Planning, geometry, and complexity of robot motion*. Ablex Publishing, Norwood NJ, 1987.
- [31] A. Hourtash, M. Tarokh. Manipulator path planning by decomposition: Algorithm and analysis. *IEEE Trans. on Robotics and Automation*, 17(9):842–856, 2001.
- [32] M. Iwamura, M. Yamamoto, A. Mohri. A gradient-based approach to collision-free quasi-optimal trajectory planning of nonholonomic systems. *Int. Conf. on Intelligent Robots and Systems*, tom 3, ss. 1734–1740, Takamatsu, 2000.
- [33] W. Jacak. *Roboty inteligentne. Metody planowania działań i ruchów*. Wydawnictwo Politechniki Wrocławskiej, Wrocław, 1991.
- [34] G. Jacob. Lyndon discretization and exact motion planning. *European Control Conf.*, ss. 1507–1512, Grenoble, 1991.
- [35] G. Jacob. Motion planning by piecewise constant or polynomial inputs. *IFAC Nonlinear Control Syst.*, ss. 239–244, 1992.
- [36] J. Jakubiak, K. Tchoń. Motion planning in endogenous configuration space. *Proc. 3rd Int. Workshop on Robot Motion and Control*, ss. 231–236, Bukowy Dworek, 2002.
- [37] F. Jean. Complexity of nonholonomic motion planning. *Int. Journ. of Control*, 74(8):776–782, 2001.
- [38] B. Kernighan, D. M. Ritchie. *Język ANSI C*. WNT, 1994.
- [39] J.-O. Kim, P. Khosla. Real-time obstacle avoidance using harmonic potential functions. *Trans. on Robotics and Automation*, 8(3):338–349, 1992.
- [40] D. E. Knuth. *T<sub>E</sub>X: The program, volume B of computers and Typesetting*. Addison-Wesley, USA, 1986.
- [41] F. Lafferriere, H. Sussmann. Motion Planning for Controllable Systems without Drift. *Proc. IEEE Conf. on Rob. Autom.*, ss. 1148–1153, Sacramento, 1991.
- [42] G. Lafferriere. A general strategy for computing steering controls of systems without drift. *30th Conf. on Decision and Control*, tom 2, ss. 1115–1120, Brighton, 1991.
- [43] G. Lafferriere, H. Sussmann. Motion planning for controlable systems without drift: A preliminary report, 1990. Rutgers Center for Systems and Control.
- [44] F. Lamiraud, J.-P. Laumond. Smooth motion planning for car-like vehicles. *IEEE Trans. on Rob. and Aut.*, 17(4):498–502, 2001.
- [45] L. Lamport. *L<sup>A</sup>T<sub>E</sub>X: System przygotowania dokumentów*. Ariel, Kraków, 1992.
- [46] J.-P. Laumond. Feasible trajectories for mobile robots with kinematic and environment constraints. *Int. Conf. on Intelligent Autonomous Systems*, ss. 346–354, Amsterdam, 1987.
- [47] S. LaValle. *Planning algorithms*. Cambridge University Press, 2006.
- [48] F. Lizarralde, J. Wen. Feedback stabilization of nonholonomic systems based on path space iteration. *2nd Int. Symp. on Methods and Models in Automation and Robotics, MMAR'95*, ss. 485–490, Międzyzdroje, 1995.
- [49] P. Ludwików. Statyczne oddziaływanie wybranych platform robotów na podłoże. K. Tchoń, redaktor, *Postępy robotyki. Przemysłowe i medyczne systemy robotyczne*, tom 2, ss. 339–347, Polanica Zdrój, 2005. WKŁ.

- [50] P. Ludwików. Lokalne wskaźniki obszerności manewru bezdryfowego układu nieholonomicznego. K. Tchoń, redaktor, *Postępy robotyki*, tom 1, ss. 119–128, Piechowice, 2006.
- [51] P. Ludwików, I. Duleba. Direct vs. null space approach in posture optimization while solving inverse kinematics of redundant manipulators. S. Domek, R. Kaszyński, redaktorzy, *Proc. of the 11th IEEE Int. Conf. on Methods and Models in Automation and Robotics*, ss. 561–564, Międzyzdroje, 2005.
- [52] P. Ludwików, I. Duleba. Lie algebraic approach to nonholonomic motion planning in cluttered environment. K. Kozłowski, redaktor, *Robot Motion and Control 2007*, Lecture Notes in Control and Information Sciences, ss. 249–257. Springer, 2007.
- [53] A. Maciejewski, C. Klein. The singular value decomposition: Computation and applications to robotics. *Int. Journ. of Robotics Research*, 8(6):63–79, 1989.
- [54] G. Melançon, C. Reutenauer. Lyndon words, free algebras and shuffles. *Can. Journ. Math.*, XLI(4):577–591, 1989.
- [55] R. Montgomery. Abnormal Minimizers. *SIAM J. Contr. and Optimization*, 32(6):1605–1620, 1994.
- [56] A. Mukerjee, S. Sharma, R. Agrawal. When is an obstacle a Perfect obstacle? *Int. Conf. on Robotics and Automation*, ss. 2273–2278, 1995.
- [57] R. Murray, S. Sastry. Nonholonomic motion planning: Steering using sinusoids. *IEEE Trans. on Aut. Contr.*, 38(5):700–716, 1993.
- [58] R. Murray, S. Sastry, Z. Li. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.
- [59] Y. Nakamura. *Advanced Robotics: Redundancy and Optimization*. Addison Wesley, New York, 1991.
- [60] Y. Nakamura, R. Mukherjee. Nonholonomic path planning of space robots via a bidirectional approach. *IEEE Transactions on Robotics and Automation*, 7(4):500–514, 1991.
- [61] C. Ong, E. Gilbert. Robot path planning with penetration growth distance. *Journ. of Robotic Systems*, 2(15):57–74, 1998.
- [62] R. Paul. *Robot Manipulators: Mathematics, Programming and Control*. MIT Press, Cambridge, 1981.
- [63] M. Pilgrim. *Dive Into Python*. Apress, 2004.
- [64] D. Popa, J. Wen. Nonholonomic path-planning with obstacle avoidance: a path-space approach. *Int. Conf. on Robotics and Automation*, ss. 2662–2667, Minneapolis, 1996.
- [65] W. Press, B. Flannery, S. Teukolsky, W. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992.
- [66] J. Reeds, R. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific Journ. Math.*, 2(145):367–393, 1990.
- [67] D. Reister, F. Pin. Time-optimal trajectories for mobile robots with two independently driven wheels. *Int. Journ. of Robotics Research*, 13(1):38–54, 1994.
- [68] R. Resnick, D. Halliday. *Fizyka dla studentów nauk przyrodniczych i technicznych*, tom 1. PWN, Warszawa, 1973.
- [69] S. Richter, R. DeCarlo. Continuation methods: Theory and applications. *IEEE Trans. on Circuits and Systems*, CAS-30(6):347–352, 1983.
- [70] S. Samuel, S. Keerthi. Numerical determination of optimal non-holonomic paths in the presence of obstacles. *IEEE Int. Conf. on Robotics and Automation*, tom 1, ss. 826–831, Atlanta, 1993.
- [71] S. Sekhavat, M. Chyba. Nonholonomic deformation of a potential field for motion planning. *IEEE Int. Conf. on Robotics and Automation*, ss. 817–822, Detroit, Michigan, 1999.
- [72] J. Sellen. Planning paths of minimal curvature. *IEEE Int. Conf. on Robotics and Automation*, ss. 1976–1982, 1995.
- [73] T. Siméon, S. Leroy, J. Laumond. Computing good holonomic collision-free paths to steer nonholonomic mobile robots. *IEEE Int. Conf. on Intelligent Robots and Systems*, tom 2, ss. 1004–1009, 1997.
- [74] E. Sontag. Gradient techniques for systems with no drift: A classical idea revisited. *Proc. of the 32nd Conf. on Decision and Control*, ss. 2706–2711, San Antonio, Texas, 1993.
- [75] K. Souccar, J. Coelho, C. Connolly, R. Grupen. Harmonic functions for path planning and control. K. Gupta, A. del Pobil, redaktorzy, *Practical Motion Planning in Robotics: Current Approaches and Future Directions*, rozdział 14, ss. 277–300. John Wiley & Sons, 1998.
- [76] D. Sourlas, V. Manousiouthakis. On the computation of the nonlinearity measure. *Proc. of the 37th IEEE Conf. on Decision and Control*, ss. 1434–1439, Florida, USA, 1998.

- [77] M. W. Spong, M. Vidyasagar. *Robot Dynamics and Control*. John Wiley & Sons, Inc., New York, 1989.
- [78] R. Strichartz. The Campbell-Baker-Hausdorff-Dynkin Formula and Solutions of Differential Equations. *Journ. of Funct. Anal.*, 72:320–345, 1987.
- [79] S. Sundar, Z. Shiller. Optimal obstacle avoidance based on the hamilton-jacobi-bellman equation. *Trans. on Robotics and Automation*, 13(2):305–310, 1997.
- [80] H. Sussmann. A continuation method for nonholonomic path-finding problems. *32nd Conf. on Decision and Control*, tom 3, ss. 2718–2723, San Antonio, 1993.
- [81] H. Sussmann. Local controllability and motion planning for some cases of systems with drift. *30th Conf. on Decision and Control*, ss. 1110–1114, Brighton, 1991.
- [82] H. Sussmann. Two new methods for motion planning for controllable systems without drift. *European Control Conf.*, ss. 1501–1506, Grenoble, 1991.
- [83] K. Tchoń, J. Jakubiak. An extended jacobian inverse kinematics algorithm for doubly nonholonomic mobile manipulators. *Proc. IEEE Int. Conf. on Robotics and Automation*, ss. 1560–1565, Barcelona, 2005.
- [84] K. Tchoń, J. Jakubiak. A repeatable inverse kinematics algorithm with linear invariant subspaces for mobile manipulators. *IEEE Trans. Syst. Man Cybern.*, 35(5):1051–1057, 2005.
- [85] K. Tchoń, J. Jakubiak. Extended Jacobian inverse kinematics algorithm for nonholonomic mobile robots. *Int. Journ. Control*, 79(8):895–909, 2006.
- [86] K. Tchoń, J. Jakubiak. Jacobian inverse kinematics algorithms with variable steplength for mobile manipulators. J. Lenarcic, B. Roth, redaktorzy, *Advances in robot kinematics. Mechanisms and motion*, ss. 465–472. Springer, Dodrecht, 2006.
- [87] K. Tchoń, A. Mazur, I. Dulęba, R. Hossa, R. Muszyński. *Manipulatory i roboty mobilne*. Akademicka Oficyna Wydawnicza PLJ, Warszawa, 2004.
- [88] D. Tilbury, J.-P. Laumond, R. Murray, S. Sastry, G. Walsh. Steering car-like systems with trailers using sinusoids. *IEEE Int. Conf. on Robotics and Automation*, ss. 1993–1998, Nice, 1992.
- [89] V. Varadarayan. *Lie groups, Lie algebras and their representations*. Prentice-Hall Inc., 1974.
- [90] M. Vendittelli, J. Laumond, P. Souères. Shortest paths to obstacles for a polygonal car-like robot. *38 Conf. on Decision and Control*, ss. 17–22, Phoenix, 1999.
- [91] H. Wacker. *Continuation Methods*. Academic, New York, 1987.
- [92] L. Wall, T. Christiansen, R. L. Schwartz, S. Potter. *Perl Programowanie*. O'Reilly, 1998.
- [93] S. Wolfram. *The Mathematica Book, 4th ed.* Wolfram Media, Cambridge University Press, Champaign, USA, 1999.