

Wydział Elektroniki,
Politechnika Wrocławska

PRACA DOKTORSKA

Metody planowania ruchu układów
bezdryfowych bazujące na algorytmie
Lafferriera-Sussmanna

Jacek Jagodziński

Promotor: prof. dr hab. inż. Ignacy Dulęba

słowa kluczowe:

- algorytm Lafferriera-Sussmanna
- metody Lie-algebraiczne
- planowanie ruchu
- układ nieholonomiczny
- układ nilpotentny
- układ bezdryfowy

Wrocław 2011

Jestem bardzo wdzięczny Panu Profesorowi dr. hab. inż. Ignacemu Dulębie za zaangażowanie, wszelkie wnikliwe uwagi i pomoc w przygotowaniu dysertacji.

Dziękuję mojej żonie Aleksandrze za miłość, wsparcie oraz cierpliwość do niezliczonych godzin spędzonych przy komputerze.

Dziękuję Rodzicom, którzy umożliwili mi tę intelektualną przygodę.



„Nikt nie rodzi się mądry.”

„Maksymy”, Ptahhotep, Prolog, wiersz 33

(...) लोकसंस्थां निर्ममे संस्थया स्वया

(o istocie najwyższej) „(...) wszechświat stworzył przez inteligencję swoją”

Śrīmad Bhagavatam, canto trzecie, część druga, (20.17)

Spis treści

Oznaczenia	3
1. Wstęp	5
2. Preliminaria matematyczne	9
2.1. Grupa Liego	9
2.2. Algebra Liego	10
2.3. Bazy wolnej algebry Liego	10
2.3.1. Baza Ph. Halla	11
2.3.2. Kombinatoryczne bazy algebry Liego	12
2.3.3. Podsumowanie	15
2.3.4. Formuła Witt'a	16
2.4. Algebra Liego a grupa Liego	17
2.4.1. Przykłady macierzowych grup Liego i ich algebr	19
2.4.2. Formalna algebra Liego	19
2.4.3. Formuła Campbella-Bakera-Hausdorffa-Dynkina	22
2.5. Uogólniona formuła Campbella-Bakera-Hausdorffa-Dynkina	22
2.6. Zadanie planowania ruchu	23
3. Algorytm Lafferriera-Sussmanna	25
3.1. Układ rozszerzony – trajektoria referencyjna	25
3.2. Równanie Chena-Fliessa-Sussmanna (CFS)	26
3.3. Generacja sterowań	28
3.3.1. Sterowania kawałkami stałe (CBHD)	28
3.3.2. Sterowania kawałkami ciągłe (gCBHD)	30
3.4. Podsumowanie	34
4. Reprezentacje ruchu	36
4.1. Algorytmizacja równań CFS	40
4.1.1. CFS w dziedzinie pól wektorowych	41
4.1.2. Równanie CFS otrzymane z formalnej grupy Liego	46
4.2. Wpływ reprezentacji ruchu na przebieg trajektorii	53
4.2.1. Miary oceny trajektorii dla zadanej reprezentacji ruchu	53
4.2.2. Wyniki symulacji	55
4.2.3. Podsumowanie	60
5. Metoda sterowań sinusoidalnych jako szczególna postać algorytmu LS	62
Podsumowanie	66
6. Trajektoria referencyjna	67

6.1. Badania symulacyjne	68
6.1.1. Trajektoria symulacyjna	70
6.1.2. Podsumowanie	71
6.2. Niezależności współczynników Halla od trajektorii referencyjnej	71
7. Realizacja sterowań za pomocą formuły gCBHD	73
7.1. Generacja optymalnych energetycznie sterowań w kierunkach ruchu wyznaczonych przez bazę Halla	73
7.2. Generacja sterowań algorytmem Newtona	76
8. Aproksymacja nilpotentna układów bezdryfowych	77
8.1. Algorytm nilpotentnej aproksymacji	77
8.2. Miary dokładności nilpotentnej aproksymacji	83
8.3. Symulacje	84
8.4. Nilpotentne sprzężenie zwrotne	91
9. Algorytmy dla układów nienilpotentnych	94
9.1. Iteracyjny algorytm LS wspomagany nilpotentną aproksymacją	94
9.2. Metoda Lie algebraiczna z iteracyjnym doбором podcelów	96
9.3. Symulacje	98
9.4. Podsumowanie	104
10. Podsumowanie	106
Bibliografia	107
A. Modele	113
A.1. Integrator Brocketta	113
A.2. Czterowymiarowy układ łańcuchowy	113
A.3. Jednokołowy robot mobilny	113
A.4. Trójkołowy robot mobilny	114
B. Podstawowe struktury algebraiczne	116

Oznaczenia

Opisy obiektów matematycznych podlegają następującym regułom: wielkości skalarne pisane są kursywą (a), wielkości wielowymiarowe (wektory, macierze) oraz ich przekształcenia są oznaczane pogrubionymi symbolami; zwykle duże litery oznaczają macierze (\mathbf{A}), a małe – wektory (\mathbf{a}). Domyślnie wszystkie wektory są kolumnowe. Przestrzenie, zbiory oraz inne specjalne obiekty, takie jak struktury algebraiczne są oznaczane przy pomocy stylizowanych dużych liter (\mathbb{A} , \mathcal{A} , \mathbf{A} \mathfrak{A} itp.). Niekomutujące mnożenie oznaczane symbolem $*$ z reguły jest w pracy pomijane, pojawia się głównie w celu zaznaczenia nieprzemienności.

Wykorzystywane w pracy oznaczenia w porządku alfabetycznym:

\leftarrow	– operacja powiększenia zmiennej i o jeden ($i \leftarrow i + 1$)
$\mathbf{0}$	– wektor (macierz) o zerowych elementach
$\binom{n}{k}$	– symbol Newtona
$\lceil \cdot \rceil$	– sufit, $\lceil x \rceil = \min\{k \in \mathbb{Z} : k \geq x\}$
$\lfloor \cdot \rfloor$	– podłoga, $\lfloor x \rfloor = \max\{k \in \mathbb{Z} : k \leq x\}$
$\langle \cdot, \cdot \rangle$, $\mathbf{a} \circ \mathbf{b}$	– iloczyn skalarny
$*$	– nieprzemienne mnożenie
$\mathbf{a} \times \mathbf{b}$	– iloczyn wektorowy
\mathbf{a}^T	– transpozycja
$\ \mathbf{a}\ $	– norma euklidesowa wektora
\mathbf{A}^{-1}	– odwrotność macierzy kwadratowej
$\mathbf{A}^\#$	– pseudoodwrotność Moore’a-Penrose’a macierzy: $\mathbf{A}^\# = \mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1}$
$\mathbf{A}_{a \times b}$	– macierz o a wierszach i b kolumnach
$[\mathbf{A}, \mathbf{B}]$	– nawias Liego
$\text{ad}_{\mathbf{A}^k} \mathbf{B}$	– iterowany nawias Liego: $\text{ad}_{\mathbf{A}^0} \mathbf{B} = \mathbf{B}$, $\text{ad}_{\mathbf{A}^k} \mathbf{B} = [\mathbf{A}, \text{ad}_{\mathbf{A}^{k-1}} \mathbf{B}]$
\mathbb{C}^k	– klasa funkcji ciągłych rzędu k , gdy $k = \infty$ – gładkich
$\det(\mathbf{A})$	– wyznacznik macierzy kwadratowej
$\text{diag}(\mathbf{A})$	– macierz diagonalna
$\dim(\mathbf{a})$	– wymiar (przestrzeni, wektora)
$\mathbf{G}(\mathbf{q})_{n \times m}$	– macierz generatorów
\mathbf{I}	– kwadratowa macierz jednostkowa, operator tożsamościowy
\mathbf{K}	– ciało
\mathbb{N}	– zbiór liczb naturalnych

$\text{rank}(\mathbf{A})$	- rząd macierzy
\mathbb{R}^k	- k -wymiarowa przestrzeń liczb rzeczywistych
$\text{SE}(3)$	- specjalna grupa euklidesowa, (zawiera macierz obrotu i wektor translacji), $\text{SE}(3) \simeq \text{SO}(3) \times \mathbb{R}^3$
$\text{sgn}(a)$	- funkcja przyjmująca wartość 1 dla $a > 0$, wartość 0 dla $a = 0$, oraz -1 dla $a < 0$
$\text{SO}(3)$	- specjalna grupa obrotów wymiaru 3 (macierze $\mathbf{R}_{3 \times 3}$ o własnościach $\mathbf{R}^{-1} = \mathbf{R}^T$, $\det(\mathbf{R}) = 1$)
$\text{span}(\mathbf{A})$	- przestrzeń rozpięta przez \mathbf{A}
$\text{trace}(\mathbf{A})$	- ślad macierzy, suma elementów na głównej diagonali
\mathbf{V}	- przestrzeń wektorowa
\mathbb{Z}	- zbiór liczb całkowitych

1. Wstęp

Zastosowanie równań różniczkowych w sterowaniu sięga roku 1840, gdy brytyjski astronom w Greenwich, G. B. Airy, wynalazł sterowanie ze sprzężeniem zwrotnym do nakierowywania teleskopu. Jakkolwiek pętla sprzężenia powodowała duże oscylacje, to Airy był pierwszym badającym niestabilność układów z zamkniętą pętlą i pierwszym, który wykorzystał równania różniczkowe do analizy systemów [2]. Rachunek różniczkowy był już wówczas dobrze rozwinięty dzięki pracom Newtona, Leibniza, braci Bernoullich, Riccatiego i innych. Markow z kolei był pierwszym, który napisał pracę dotyczącą stricte planowania ruchu [64], a poświęconą zagadnieniu optymalnego układania torów kolejowych. W latach 50-tych ubiegłego wieku Dubins [16] zajął się optymalnym planowaniem ruchu pojazdu (sterowaniami typu bang-bang). Dalszy rozwój teorii sterowania, (m.in. Hermes, Brockett, Sussmann) doprowadził do powstania modelu w postaci układu afinicznego [8, 96, 101], którego specjalną podklasą są nieholonomiczne układy bezdryfowe rozpatrywane w niniejszej dysertacji.

Układy nieholonomiczne charakteryzowane są przez niecałkowalne równania ograniczeń zawierające zmienne stanu i ich pochodne. W literaturze robotycznej można znaleźć liczne przykłady układów nieholonomicznych. Są wśród nich kołowe roboty mobilne (także z przyczepami) [40, 56, 57, 81] oraz roboty typu trójkątny smok (także z wieloma ramionami) [41, 42] dla których ograniczenia wynikają z braku poślizgów, a także roboty podwodne [108] czy szybujące [76], których nieholonomiczność wynika z prawa zachowania momentu pędu. Innymi przykładami robotów nieholonomicznych są: robot skaczący w fazie lotu [103], manipulatory nieholonomiczne [75, 99] oraz niektóre roboty kroczące (heksapody) [33, 34].

Trudność zadań planowania ruchu, czyli przemieszczania układu między konfiguracjami brzegowymi, dla układów nieholonomicznych wynika głównie z mniejszej liczby sterowań niż wymiarowość przestrzeni stanu. Od lat 90-tych XX wieku nastąpił ogromny rozwój metod planowania ruchu układów bezdryfowych. Można je podzielić na metody ogólnego przeznaczenia, które nie wymagają od układu sterowanego żadnych szczególnych cech i metody specjalizowane, działające dla pewnej podklasy układów. Wśród metod ogólnych są, między innymi, metoda bazująca na zasadzie maksimum Pontriagina [79], metoda Newtona [15], metoda przestrzeni endogenicznej [100], metoda funkcji transwersalnych [63], metoda wykorzystująca technikę uśredniania Sussmanna i Liu [61], metoda Lie algebraiczna [17], metoda typu lapunowskiego [76], metoda gradientów [90]. Wymienione metody są zwykle dość trudne, często złożone obliczeniowo i najczęściej wymagające obliczeń numerycznych.

By pozbyć się części lub wszystkich wad metod ogólnego przeznaczenia rozwinął się drugi nurt badań nad specjalizowanymi metodami planowania ruchu. Wśród metod specjalizowanych jest metoda sterowań sinusoidalnych dedykowana układom łańcuchowym [71] dekomponująca zadanie planowania na etapy, w których kreśląc pętle w przestrzeni sterowań kolejne współrzędne wektora stanu uzyskują swe docelowe wartości (wariant tej metody wykorzystuje sterowania kawałkami stałe [67]). Kolejną metodą specjalizowaną jest sterowanie we współrzędnych biegunowych zaproponowane w pracach [3, 11], które stały się załączkiem metody orientowanych pól wektorowych, VFO [66] dedykowanej układom dwuwęjsciowym, w których jeden z generatorów jest postaci $(1, 0, \dots, 0)^T$, a drugi spełnia dodatkowe warunki (regularność, ograniczoność i orientowalność). Podział zmiennych wektora stanu na dwie grupy współrzędnych sterowanych bezpośrednio i pośrednio wykorzystuje metoda bazująca na twierdzeniu Stockesa [68]. Algorytm Murraya wykorzystuje postać trójkątna niektórych układów nieholonomicznych [72], a metoda Fliessa i współpracowników jest dedykowana układom płaskim [81]. Klasyczna metoda Dubinsa znalazła swe rozwinięcie u Reeda-Sheepa [78]. Obszerny przegląd innych metody planowania ruchu można znaleźć w książkach [59, 60, 70, 101].

W nurt metod specjalizowanych wpisuje się rozważana w rozprawie metoda Lafferriera i Sussmanna (LS) [53, 54, 55], która wyrosła na bazie analizy matematycznej i algebry skojarzonych z geometrią różniczkową, a jest przeznaczona do planowania ruchu układów nilpotentnych. Jest to metoda w dużej mierze analityczna, o ciekawych matematycznie podstawach teoretycznych i praktycznie użyteczna. Ponieważ metoda nie ogranicza projektanta algorytmu planowania ruchu, bowiem stanowi zapis idei kolejnych kroków, a nie ich jednoznaczną wykładnię, zatem tezą niniejszej dysertacji jest:

rozszerzony algorytm Lafferriera-Sussmanna może być efektywniejszy niż jego pierwotna wersja w planowaniu ruchu układów nilpotentnych oraz bezdryfowych.

Praca zorganizowana jest w osiem rozdziałów uzupełnionych spisem literatury i dwoma dodatkami, a jej zawartość jest następująca: we wstępnym rozdziale 2. wprowadzono niezbędną terminologię, przedstawiono definicje najważniejszych pojęć i zaprezentowano użyteczne formuły matematyczne. Wychodząc od pojęcia grupy Liego (której elementy mają związek z lokalną postacią trajektorii układu) przedstawiono algebrę Liego (której elementy w terminologii robotycznej odpowiadają prędkościom układu w jego bieżącej konfiguracji) i pokazano związek między tymi pojęciami. Sporo uwagi poświęcono bazie algebry Liego, gdyż gwarantuje ona nieredundantny sposób opisu możliwości ruchowych układu. W praktyce robotycznej powszechnie stosowana jest baza Halla. W niniejszej dysertacji przedstawiono równoważne jej, a zdecydowanie mniej popularne, bazy Lyndona, Shirshova i Chibrikowa. Dla układów nilpotentnych baza algebry Liego jest skończeniowymiarowa. W swej wersji oryginalnej algorytm LS zakładał realizację sterowań kawałkami ciągłych, których postawą teoretyczną jest formuła Campbella-Bakera-Hausdorffa-Dynkina. W rozdziale przedstawiono tę formułę wraz z jej uogólnioną wersją umożliwiającą generację sterowań ciągłych. Rozdział 2. koń-

czy się postawieniem zadania planowania ruchu i sformułowaniem twierdzenia Chow, wymaganym do założonej sterowalności układu. W rozdziale 3. przedstawiono kolejne kroki oryginalnego algorytmu LS wraz z ich dyskusją i przykładami ilustrującymi. Sporo miejsca poświęcono sposobowi generacji sterowań dla zadanego ruchu, formułując to zadanie jako analogon zadania kinematyki odwrotnej manipulatorów. Rozdział podsumowano szkicami dwóch algorytmów planowania ruchu układów nilpotentnych różnicowanych sposobem generacji sterowań oraz przedstawiono proponowany schemat zapisu i modyfikacji (uzmienniania składowych) algorytmu LS. Obszerny rozdział 4. poświęcono najważniejszemu elementowi podlegającemu wyborowi – reprezentacji ruchu. W oryginalnym algorytmie LS pokazano tylko dwie możliwe reprezentacje znane jako „w przód” i „wstecz”. W niniejszej dysertacji znacznie rozszerzono liczbę dopuszczalnych reprezentacji. Każdej reprezentacji przyporządkowane jest pewne równanie różniczkowe znane jako równanie Chena-Fliessa-Sussmanna (CFS). Dla klasy reprezentacji dopuszczalnych udowodniono charakterystyczną wielomianowo-trójkątną postać tego równania. Następnie pokazano metody wyprowadzania równania CFS. Zaprezentowano literaturowy sposób bazujący na podejściu Sussmanna, który intensywnie wykorzystuje własności pól wektorowych i ich strumieni. Z kolei pokazano również literaturowy sposób otrzymywania równań CFS korzystając z formalnej grupy Liego. Obydwa wymienione sposoby są efektywne jedynie w przypadku niskowymiarowych zadań, gdy obliczenia nie są zbyt trudne. Dla przypadków bardziej skomplikowanych zaproponowano autorskie sposoby modyfikacji drugiej z metod. Pierwsza z opisanych modyfikacji polega na przeniesieniu obliczeń do przestrzeni jednomianów, a ich wynikiem jest konieczność rozwiązania niskowymiarowych układów równań liniowych w celu selekcji równań niezależnych. Druga z modyfikacji idzie jeszcze dalej i korzysta dodatkowo ze struktury generowanych jednomianów, która umożliwia znaczne uproszczenie procesu selekcji równań. Każdy ze sposobów generacji równań CFS zapisano jako algorytm (Algorytmy 3–6), a ich działanie zilustrowano przykładami (dla algorytmów autorskich odpowiednio trudnymi dla literaturowych). Dalszą część rozdziału stanowi badanie przebiegu trajektorii odpowiadających zadanym reprezentacjom ruchu na wybranych układach nilpotentnych. W tym celu wprowadzono miary określające jak generowana trajektoria jest usytuowana względem trajektorii prostoliniowej łączącej zadane konfiguracje brzegowe. Przebieg trajektorii rozwiązującej zadanie planowania nie jest bardzo istotny w bezkolizyjnej przestrzeni konfiguracyjnej, a kluczowy – gdy w tej przestrzeni występują obszary zabronione (przeszkody). Przedmiotem rozdziału 5. jest zastosowanie metody LS do sterowania układów łańcuchowych, będących szczególną postacią układów nilpotentnych, i porównanie wyników z literaturową metodą sterowań sinusoidalnych Murraya-Sastriego. W rozdziale 6. zweryfikowano wpływ trajektorii referencyjnej na rozwiązanie równań CFS. Literaturowe doniesienia mówiły o potencjalnej możliwości wpływania przez trajektorie referencyjną na kształt trajektorii wynikowej. Hipotezę tę najpierw negatywnie zweryfikowano na wybranym układzie nilpotentnym dla kilku rodzin trajektorii referencyjnych, a następnie sformułowano

twierdzenie mówiące o nieistnieniu takiego wpływu. Rozdział 7 jest poświęcony generowaniu sterowań, z wykorzystaniem formuły gCBHD, realizujących ruch zadany przez składowe reprezentacji ruchu. Wyniki z tego rozdziału znajdują zastosowanie we wszystkich zadaniach testowych przedstawionych w dysertacji. Wyróżniono dwa przypadki: ruchu w kierunkach początkowych elementów bazy Halla, dla których możliwe jest otrzymanie optymalnych energetycznych sterowań w klasie funkcji harmoniczných. Dla przypadku ruchu ogólnego zaproponowano zastosowanie metody Newtona.

Algorytm LS realizuje zadanie planowania ruchu dla układów nilpotentnych. Dla układów bezdryfowych nie posiadających tej cechy Sussmann i Lafferriere zaproponowali iteracyjną wersję LS [97]. W literaturze znane są sposoby przybliżania ogólnego układu sterowania z zastosowaniem nilpotentnej aproksymacji [4, 38, 58, 83, 94, 105, 106]. W rozdziale 8 przypomniano metodę nilpotentnej aproksymacji Lamounda i współpracowników oraz sformułowano autorskie miary oceny tej procedury. Miary te określono dla dwóch układów bezdryfowych różniących się wymiarowością przestrzeni konfiguracyjnej. Procedurę nilpotentnej aproksymacji wykorzystano w rozdziale 9. jako wspomagającą działania algorytmu LS dla układów bezdryfowych. Dla porównania, w tym samym rozdziale, pokazano działanie metody Lie-algebraicznej zastosowanej do identycznych zadań testowych. Metody przetestowano symulacyjnie zmieniając reprezentację ruchu oraz punkty brzegowe planowania. W rozdziale 10. zebrano osiągnięcia pracy formułując wnioski końcowe. W dodatkach zamieszczono: modele układów, na których testowano różne aspekty algorytmu LS (dodatek A) oraz przypomniano podstawowe struktury algebraiczne wykorzystywane w dysertacji (dodatek B).

2. Preliminaria matematyczne

W ostatnim trzydziestoleciu XIX w. Marius Sophus Lie (1842-1899) stworzył aparat matematyczny służący jako narzędzie do rozwiązywania równań różniczkowych. Od tego czasu teoria grup i algebr Liego rozwinęła się i utworzyła odrębną dyscyplinę naukową. Dzięki swej obszerności grupy Liego połączyły algebrę z geometrią, tworząc udany mariaż geometrii różniczkowej z teorii grup, o szerokim spektrum zastosowań m.in. w analizie numerycznej, fizyce teoretycznej i kwantowej, mechanice statystycznej i teorii sterowania.

Aparat ten jest intensywnie wykorzystywany w niniejszej dysertacji. W rozdziale przytoczono podstawowe definicje: grupy Liego, algebry Liego, baz algebry Liego, pokazano związek między grupą a algebrą Liego. Zwieńczeniem rozdziału jest formuła Campbella-Bakera-Hausdorffa-Dynkina i jej uogólniona postać. Wprowadzono wiele pojęć i oznaczeń intensywnie wykorzystywanych w dalszej części dysertacji (przypomnienie podstawowych struktur algebraicznych zebrano w Dodatku B). Jakkolwiek prezentacja materiału jest nieco obszerniejsza niż wynikałoby to li tylko z praktycznej użyteczności, to po wyczerpujące potraktowanie materiału należy skonsultować matematyczne pozycje literaturowe [49, 80, 92].

2.1. Grupa Liego

Grupa Liego \mathcal{G} jest różniczkowalną rozmaitością wyposażoną w strukturę grupy dzięki odwzorowaniu

$$\mu : \mathcal{G} \times \mathcal{G} \rightarrow \mathcal{G}; \quad (g, h) \mapsto gh,$$

oraz przekształceniu odwrotnemu

$$\mathcal{I} : \mathcal{G} \rightarrow \mathcal{G}; \quad g \mapsto g^{-1},$$

będącymi funkcjami klasy \mathbb{C}^∞ [49]. Jeśli odwzorowanie μ jest przemienne – grupę nazywa się abelową.

Przykład 1. Niech \mathbf{V} będzie skończeniowymiarową przestrzenią wektorową nad \mathbb{R} . Operacją grupową jest dodawanie wektorów, natomiast elementem odwrotnym do \mathbf{x} niech będzie wektor $-\mathbf{x}$. Zarówno $\mu(\mathbf{x}, \mathbf{y}) = \mathbf{x} + \mathbf{y}$ jak i $\mathcal{I}(\mathbf{x}) = -\mathbf{x}$ są klasy \mathbb{C}^∞ . Wówczas $(\mathbf{V}, +)$ jest abelową grupą Liego (izomorficzną z \mathbb{R}^n , gdzie $n = \dim(\mathbf{V})$).

2.2. Algebra Liego

Algebrą Liego $\mathcal{L}(\mathbf{V})$ nad ciałem \mathbf{K} (o charakterystyce $\neq 2$) nazywamy strukturę algebraiczną $(\mathbf{V}, [\cdot, \cdot])$ złożoną z przestrzeni wektorowej \mathbf{V} nad \mathbf{K} oraz dwuliniowego odwzorowania $[\cdot, \cdot] : \mathbf{V} \times \mathbf{V} \rightarrow \mathbf{V}$ zwanego nawiasem Liego (komutatorem) o własnościach:

$$\forall \mathbf{X} \in \mathbf{V} : [\mathbf{X}, \mathbf{X}] = \mathbf{0}, \quad (2.1)$$

$$\forall \mathbf{X}, \mathbf{Y} \in \mathbf{V} : [\mathbf{X}, \mathbf{Y}] + [\mathbf{Y}, \mathbf{X}] = \mathbf{0} \quad (\text{antysymetria}), \quad (2.2)$$

$$\forall \mathbf{X}, \mathbf{Y}, \mathbf{Z} \in \mathbf{V} : [\mathbf{X}, [\mathbf{Y}, \mathbf{Z}]] + [\mathbf{Y}, [\mathbf{Z}, \mathbf{X}]] + [\mathbf{Z}, [\mathbf{X}, \mathbf{Y}]] = \mathbf{0} \quad (\text{tożsamość Jacobiego}). \quad (2.3)$$

Nawias Liego jest antysymetryczny (2.2) i nie jest stowarzyszony (2.3). Dzięki dwuliniowości: $\forall \epsilon, \gamma \in \mathbf{K}$ i $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \in \mathbf{V} : [\epsilon\mathbf{X} + \gamma\mathbf{Y}, \mathbf{Z}] = \epsilon[\mathbf{X}, \mathbf{Z}] + \gamma[\mathbf{Y}, \mathbf{Z}]$, $[\mathbf{X}, \epsilon\mathbf{Y} + \gamma\mathbf{Z}] = \epsilon[\mathbf{X}, \mathbf{Y}] + \gamma[\mathbf{X}, \mathbf{Z}]$ warunki (2.1), (2.2) są równoważne. Niektórzy matematycy jako aksjomat traktują warunek (2.1) [80], inni – antysymetrię [49]. Gdy rozpatrujemy algebrę Liego o dwóch generatorach, tożsamość Jacobiego posiada także inne sformułowanie, przykładowo $[[[\mathbf{X}, \mathbf{Y}], \mathbf{X}], \mathbf{Y}] = [[[\mathbf{X}, \mathbf{Y}], \mathbf{Y}], \mathbf{X}]$. Ponieważ od nawiasu Liego nie żądamy łączności, w rzadkich przypadkach komutator jest działaniem łącznym $[\mathbf{X}, \mathbf{Y}] = \mathbf{0}$, zatem algebra Liego nie jest szczególnym przypadkiem algebry (porównaj definicję algebry z Dodatku B).

2.3. Bazy wolnej algebry Liego

Algebra Liego jest rozpięta przez zbiór generatorów i wszystkich możliwych komutatorów tych elementów. Podzbiór elementów niezależnych tworzy bazę algebry Liego. Sformułowanie wolna oznacza, że żąda się niezależności pomiędzy generatorami oraz ich potomkami (wygenerowanymi nawiasem Liego), które nazywamy jednomianami Liego. Zachowane są jedynie aksjomaty antysymetrii (2.2) i tożsamości Jacobiego (2.3). Dla jednomianów Liego wprowadzamy pojęcie stopnia i warstwy. Stopień jednomianu Liego jest zdefiniowany rekurencyjnie

$$\begin{cases} \deg(\mathbf{A}) = 1 & \text{jeśli } \mathbf{A} \text{ jest generatorem,} \\ \deg([\mathbf{A}, \mathbf{B}]) = \deg(\mathbf{A}) + \deg(\mathbf{B}) & \text{dla złożonych jednomianów Liego.} \end{cases} \quad (2.4)$$

Przykład 2. Niech \mathbf{X}, \mathbf{Y} będą generatorami, wtedy $\deg([[[\mathbf{X}, \mathbf{Y}], \mathbf{X}], \mathbf{Y}]) = 4$, $\deg([[[\mathbf{X}, \mathbf{Y}], [\mathbf{Y}, \mathbf{X}]], \mathbf{X}]) = 5$.

Jednomiany Liego o identycznym stopniu tworzą warstwę

$$\mathbb{W}_i = \{\mathbf{X} \mid \deg(\mathbf{X}) = i\}. \quad (2.5)$$

Wszystkie własności uzyskane dla wolnej algebry Liego natychmiast transformują się do szczególnych postaci algebr Liego (np. algebry Liego pól wektorowych). Dla uniknięcia redundancji w operacjach na jednomianach Liego ważnym jest określenie bazy algebry Liego. Literaturowym konstrukcjom baz poświęcono kolejne podrozdziały.

2.3.1. Baza Ph. Halla

W poszukiwaniu bazy wolnej algebry Liego, własność skośnej symetrii i tożsamość Jacobiego powinny być automatycznie wykorzystane do eliminacji zależności między kolejnymi wygenerowanymi jednomianami Liego. Tę cechę zapewnia algorytm generacji bazy Ph. Halla, \mathbb{H} , który polega na wyborze jednomianów Liego zgodnie z następującymi regułami [101]:

1. $\mathbf{B}_i \in \mathbb{H}$, $i = 1, \dots, m$ (wszystkie generatory należą do bazy Halla i są arbitralnie uporządkowane),
2. jeżeli $\deg(\mathbf{B}_i) < \deg(\mathbf{B}_j)$ wtedy $\mathbf{B}_i \stackrel{\mathbb{H}}{<} \mathbf{B}_j$, czyli jeżeli stopień pewnego jednomianu Liego jest mniejszy od stopnia innego jednomianu, to w bazie ten pierwszy jednomian poprzedza (niekoniecznie bezpośrednio) drugi (jeśli w ogóle w bazie te jednomiany występują),
3. $[\mathbf{B}_i, \mathbf{B}_j] \in \mathbb{H}$ wtedy i tylko wtedy, gdy
 - a) $\mathbf{B}_i, \mathbf{B}_j \in \mathbb{H}$ i $\mathbf{B}_i \stackrel{\mathbb{H}}{<} \mathbf{B}_j$,
 - b) albo $\mathbf{B}_j = \mathbf{B}_k$ dla pewnego $k = 1, \dots, m$,

albo $\mathbf{B}_j = [\mathbf{B}_l, \mathbf{B}_r]$ dla pewnych $\mathbf{B}_l, \mathbf{B}_r \in \mathbb{H}$ i $\mathbf{B}_l \stackrel{\mathbb{H}}{<} \mathbf{B}_r$.

Dowody, że baza Halla stanowi bazę algebry Liego można znaleźć w [36, 86]. Baza zaprezentowana powyżej znana P. Hallowi, została po raz pierwszy wprowadzona przez M. Halla [36]. Schützenberger [85] uogólnił konstrukcję bazy przez osłabienie warunku 2, natomiast Viennot [107] zastąpił warunek 2 następującym:

2*. Jeśli $[\mathbf{B}_i, \mathbf{B}_j] \in \mathbb{H}$, to $\mathbf{B}_i \in \mathbb{H}$ i $\mathbf{B}_i \stackrel{\mathbb{H}}{<} [\mathbf{B}_i, \mathbf{B}_j]$.

Postać ta jest na tyle ogólna, że zawiera w sobie bazę Lyndona i Širšova, które nie są specjalnymi przypadkami oryginalnej bazy M. Halla [102]. Aby uniknąć definicji stopnia nawiasu Liego współcześnie matematycy zapisują warunek 2 w postaci [80]:

2**. $[\mathbf{B}_i, \mathbf{B}_j] \in \mathbb{H}$, gdy $\mathbf{B}_i \stackrel{\mathbb{H}}{<} [\mathbf{B}_i, \mathbf{B}_j]$ i $\mathbf{B}_j \stackrel{\mathbb{H}}{<} [\mathbf{B}_i, \mathbf{B}_j]$.

Przykłady początkowych elementów bazy Halla o trzech generatorach przytoczono niezwłocznie, a o dwóch generatorach w tab. 2.1.

Przykład 3. Baza Halla o trzech generatorach $\mathbf{A}, \mathbf{B}, \mathbf{C}$ do warstwy trzeciej włącznie jest następująca

$$\begin{aligned}
 W_1 &: \mathbf{A}, \mathbf{B}, \mathbf{C} \\
 W_2 &: [\mathbf{A}, \mathbf{B}], [\mathbf{A}, \mathbf{C}], [\mathbf{B}, \mathbf{C}] \\
 W_3 &: [\mathbf{A}, [\mathbf{A}, \mathbf{B}]], [\mathbf{A}, [\mathbf{A}, \mathbf{C}]], [\mathbf{B}, [\mathbf{A}, \mathbf{B}]], [\mathbf{B}, [\mathbf{A}, \mathbf{C}]], \\
 & \quad [\mathbf{B}, [\mathbf{B}, \mathbf{C}]], [\mathbf{C}, [\mathbf{A}, \mathbf{B}]], [\mathbf{C}, [\mathbf{A}, \mathbf{C}]], [\mathbf{C}, [\mathbf{B}, \mathbf{C}]].
 \end{aligned}$$

Algorytm generacji bazy Halla napisany w pakiecie Mathematica znajduje się w dysertacji [72].

2.3.2. Kombinatoryczne bazy algebry Liego

Pierwsza z prezentowanych baz – Ph. Halla, konstruuje kolejne elementy bazy z elementów bazowych już powstałych, stosując ewentualnie pewne kryterium eliminacji. Natomiast pozostałe konstrukcje baz wolnej algebry Liego należące do Lyndona, Shirshova, czy Chibrikova, wywodzą się z kombinatoryki. W metodach generacji tych baz najpierw definiuje się pewien alfabet (jego moc jest równa liczbie generatorów bazy), stosuje algorytm tworzenia słów nad tym alfabetem, a później stosuje przepis zamiany wybranych słów na nawiasy Liego. Do opisu kombinatorycznych baz będą wykorzystane następujące definicje.

Zbiór wszystkich uporządkowanych liter nazywamy alfabetem i oznaczamy jako \mathbb{A} . Słowo jest sekwencją liter $(\alpha_1, \dots, \alpha_n)$, zwykle zapisywanych jako $\alpha_1 \dots \alpha_n$ i może być zastąpione pojedynczym symbolem $u = \alpha_1 \dots \alpha_n$. Długość słowa u , oznaczana jako $|u|$, jest równa liczbie liter w słowie. Niech $u = \alpha_1 \dots \alpha_n$, $w = \beta_1 \dots \beta_m$. Konkatenację dwóch słów opisuje się następująco: $uv = \alpha_1 \dots \alpha_n \beta_1 \dots \beta_m$. $\mathbb{A}^{\mathbb{P}}$ oznacza zbiór wszystkich słów nad alfabetem \mathbb{A} . Struktura złożona ze zbioru $\mathbb{A}^{\mathbb{P}}$ oraz operacji konkatenacji może być traktowana jako półgrupa.

Baza Lyndona

Baza Lyndona algebry Liego została sformułowana w [43], a praktyczny algorytm pozwalający na jej wyliczenie znajduje się w pracy [84] (o liniowej złożoności obliczeniowej ze względu na długość słowa tworzącego nawias albo stopień nawiasu). Niech dany będzie uporządkowany alfabet \mathbb{A} . Odwrotny porządek leksykograficzny na zbiorze $\mathbb{A}^{\mathbb{P}}$ zdefiniowany jest następująco

$$u < v \iff \begin{cases} \text{albo (i)} & v = fu \text{ dla pewnego } f \in \mathbb{A}^{\mathbb{P}}, \\ \text{albo (ii)} & u = fxw, v = gyw, \quad x < y, \quad x, y \in \mathbb{A}, \quad f, g \in \mathbb{A}^{\mathbb{P}}. \end{cases} \quad (2.6)$$

Słowo $w \in \mathbb{A}^{\mathbb{P}}$ jest odwrotnym słowem Lyndona wtedy i tylko wtedy, gdy jest mniejsze od wszystkich swych lewych składników. Niech \mathbb{L} oznacza zbiór słów Lyndona, wtedy słowo $b \in \mathbb{L}$, gdy spełnia

$$\begin{cases} \text{albo } b \in \mathbb{A} & \text{czyli } b \text{ jest literą} \\ \text{albo } b = ml & l < m, \quad m, l \in \mathbb{L}. \end{cases} \quad (2.7)$$

Dla najdłuższego słowa m spełniającego własność (2.7) para $(m, l) = sf(b)$ to standardowa faktoryzacja b . Każdemu odwrotnemu słowu $b \in \mathbb{L}$ przypisuje się odpowiadający mu jednomian Liego \mathbf{A}_b

$$\begin{cases} \mathbf{A}_b = b & \text{jeśli } b \in \mathbb{A}, \\ \mathbf{A}_b = [\mathbf{A}_m, \mathbf{A}_l] & \text{jeśli } sf(b) = (m, l). \end{cases} \quad (2.8)$$

Rodzina $\{\mathbf{A}_b \mid b \in \mathbb{L}\}$ jest bazą (odwrotna baza Lyndona) w przestrzeni jednomianów Liego.

Przykład 4. Słowa Lyndona alfabetu $\mathbb{A} = \{a, b\}$ w porządku (2.6) do warstwy 4. są następujące: $a, baaa, baa, bbaa, ba, bba, bbba, b$.

Baza Širšova (Shirshova) [107]

Niech \mathbb{A} będzie uporządkowanym (relacją \leq) alfabetem. Rozszerzenie porządku dla wszystkich słów jest następujące:

$$uxv < uyw \quad \text{ i } \quad u > uv \quad \text{ dla wszystkich } \quad u, v, w \in \mathbb{A}^{\mathbb{P}}, \quad x, y \in \mathbb{A} \quad \text{ takich, że } \quad x < y. \quad (2.9)$$

Niech teraz \mathbb{F} będzie zbiorem słów $w = \alpha_1 \dots \alpha_n$ ściśle większych (zgodnie z relacją \leq) od wszystkich swych cyklicznych przesunięć, które są zdefiniowane w następującej postaci $\alpha_{i+1} \dots \alpha_n \alpha_1 \dots \alpha_i$, $i = 1, \dots, n - 1$. Lemat Shirshova [88] stanowi, że każde słowo w jest niemalejącym iloczynem (konkatenacją) słów z \mathbb{F}

$$w = f_1 \dots f_n, \quad \text{ z } \quad f_1, \dots, f_n \in \mathbb{F} \quad \text{ i } \quad f_1 \leq \dots \leq f_n. \quad (2.10)$$

Podobnie jak dla słów Lyndona, słowa z \mathbb{F} również prowadzą do wolnej algebry Liego (nad \mathbb{A}), a funkcja przekształcająca słowa w nawiasy Liego $\pi: \mathbb{F} \rightarrow \mathcal{L}(\mathbb{A})$ jest następująca: $\pi(\alpha) = \alpha$, gdy $\alpha \in \mathbb{A}$. W przeciwnym przypadku słowo $w \in \mathbb{F}$ może być zapisane jako

$$w = f_1 \dots f_n \alpha \quad \text{ z } \quad \alpha \in \mathbb{A}, \quad f_1 \dots f_n \in \mathbb{F} \quad \text{ i } \quad f_1 \leq \dots \leq f_n,$$

a wtedy

$$\pi(w) = [\pi(f_1), [\pi(f_2), [\dots [\pi(f_n), \alpha]]]].$$

Zbiór $\{\pi(f) : f \in \mathbb{F}\}$ tworzy bazę Shirshova wolnej algebry Liego nad \mathbb{A} .

Jakkolwiek sposób definicji baz Shirshova i Lyndona są różne, to elementy bazy algebry Liego powstałe z zastosowania algorytmów Shirshova i Lyndona są identyczne. Z tego powodu wymienione bazy są czasem traktowane jako jedna i nazywane wspólnie bazą Lyndona-Shirshova, baza Shirshova nazywana jest także czasem bazą Chena-Foxa-Lyndona [107]. W tab. 2.1 zebrano początkowe elementy bazy Shirshova oraz Lyndona dla układu dwuwjęściowego (czyli alfabetu o dwóch literach).

Prawounormowana baza (Chibrikova)

Prawounormowana baza algebry Liego (czyli złożona z elementów o następującej postaci $[a_1, [a_2, [\dots [a_{t-1}, a_t] \dots]]]$) została zdefiniowana w pracy [12], natomiast godne polecenia wprowadzenie do zagadnień kombinatorycznych i notacji pojawiającej się w tym podrozdziale są prace Bokuta [6, 7].

Niech $\mathbb{A}^{\mathbb{M}}$ będzie wolnym monoidem wszystkich łącznych słów z \mathbb{A} (również słowo puste oznaczane jako 1), (na przykład: 1, a_1 , $a_6 a_2 a_5$). Porządek leksykograficzny (zadany relacją $<$) dla $u, v \in \mathbb{A}^{\mathbb{M}}$, opisany jest następującymi zależnościami $\forall u \neq 0$, $u < 1$ oraz

$$a_i u' = u < v = a_j v' \iff \begin{cases} \text{albo } a_i < a_j, \\ \text{albo } a_i = a_j, \quad u' < v'. \end{cases} \quad (2.11)$$

Odwrotnością słowa $u = a_{i1} a_{i2} \dots a_{it} \in \mathbb{A}^{\mathbb{M}}$ jest słowo $u^* = a_{it} \dots a_{i2} a_{i1}$. Wiodącą literą słowa u jest największa litera ze zbioru \mathbb{A} występująca w słowie u . Łączne słowo

$u \in \mathbb{A}^M$ jest regularnym, jeśli $u^* > u$ dla parzystych długości $|u|$ i $u^* \geq u$ dla nieparzystych $|u|$. Każde słowo regularne w może być zapisane w postaci $w = u_1 v^*$, gdzie $u_1 < v$, takie że $u_1 = \hat{u}_1 a_1$ dla pewnego słowa $\hat{u}_1 \geq v$ i dowolnej litery $a_1 \in \mathbb{A}$.

Wykorzystując indukcję ze względu na liczbę wystąpień wiodącej litery w słowie definiujemy podzbiór $\mathbb{T}_{\mathbb{A}} \subset \mathbb{A}^M$ w następujący sposób: jeśli w słowie $w \in \mathbb{A}^M$ wiodąca litera a występuje tylko raz i $w = ua$ dla pewnego słowa u , wtedy zakładamy, że w należy do $\mathbb{T}_{\mathbb{A}}$. Jeśli w jest słowem z \mathbb{A}^M w którym wiodąca litera pojawia się więcej niż raz i słowo w można przedstawić w następującej postaci

$$w = w_{t+1} (av)^{n_t} a u_t u_t' \dots (av)^{n_2} a u_2 u_2' (av)^{n_1} a u_1 v^* a, \quad (2.12)$$

gdzie w_{t+1}, v, u_j, u_j' nie zawierają wiodącej litery a słowa w , $n_j \geq 0$, $u_j < v$, takie że $u_j = \hat{u}_j a_j$ dla pewnych słów $\hat{u}_j \geq v$ i pewnych liter a_j dla wszystkich $1 \leq j \leq t$, to $w \in \mathbb{T}_{\mathbb{A}}$.

Warunek na u_j determinuje, że reprezentacja (2.12) słowa w , o ile istnieje, to jest jednoznaczna. Wszystkie słowa, których nie można przedstawić w postaci (2.12) nie należą do $\mathbb{T}_{\mathbb{A}}$.

Przypuśćmy, że dla pewnego alfabetu \mathbb{Z} zdefiniowano wszystkie słowa $\mathbb{T}_{\mathbb{Z}}$ z liczbą wystąpień wiodącej litery mniejszą niż w słowie w . Niech

$$\tilde{w} = w_{t+1} (av)^{n_t} a u_t u_t' \dots (av)^{n_2} a u_2 u_2' (av)^{n_1+1} a u_1.$$

Przepiszmy słowo \tilde{w} zamieniając wszystkie podsłowa postaci $(av)^{n_j} a u_j$ na $A_{(av)^{n_j} a u_j}$ (przy czym $n_1 := n_1 + 1$ dla wszystkich niepustych słów $u_j' = a_{j_1} \dots a_{j_k}$ i $w_{t+1} = a_{t+1,1} \dots a_{t+1,p}$, gdzie $a_{i_s} \in \mathbb{A}$, zamieńmy a_{i_s} na $A_{a_{i_s}}$). Wtedy otrzymamy nowe słowo

$$w^{(1)} = w_{t+1}^{(1)} A_{(av)^{n_t} a u_t} u_t'^{(1)} \dots A_{(av)^{n_2} a u_2} u_2'^{(1)} A_{(av)^{n_1+1} a u_1},$$

z alfabetem $Y = \{A_Z | z \in \{(av)^{n_j} a u_j, a_{i_s}\}\}$. Porządkujemy ten zbiór w następujący sposób:

$$A_{Z_1} \geq A_{Z_2} \iff Z_1 \geq Z_2.$$

Ponadto, liczba wystąpień wiodącej litery $w^{(1)}$ jest mniejsza niż w słowie w . Definiujemy

$$w \in \mathbb{T}_{\mathbb{A}} \iff w^{(1)} \in \mathbb{T}_Y.$$

Niech słowo (postaci (2.12)) $w_2 (av)^{n_1} a u_1 v^* a \in \mathbb{T}_{\mathbb{A}}$, oraz

$$\bar{w} = w_3 (av)^{n_2} a u_2 u_2' (av)^{n_1} a u_1 v^* a$$

będzie słowem postaci (2.12) dla $t = 2$. Wtedy

$$\bar{w} \in \mathbb{T}_{\mathbb{A}} \iff (av)^{n_1+1} a u_1 \geq (av)^{n_2} a u_2, \quad \text{oraz} \quad u_2' \text{ jest regularnym słowem.}$$

Niech $\mathcal{L}(\mathbb{A})$ będzie wolną algebrą Liego generowaną przez alfabet \mathbb{A} , natomiast $\mathbb{T}_{\mathbb{A}}$ będzie zbiorem słów zdefiniowanych uprzednio. Wtedy słowa postaci

$$[a_{i_1} [a_{i_2} [\dots [a_{i_{t-1}} a_{i_t}] \dots]]]$$

Tablica 2.1. Elementy bazy Halla, Lyndona, Shirshova i Chibrikova do warstwy szóstej włącznie generowane przez X, Y

war.	baza		
	Halla	Lyndona, Shirshova	Chibrikova
1	X Y	X Y	X Y
2	$[X, Y]$	$[Y, X]$	$[X, Y]$
3	$[X, [X, Y]]$ $[Y, [X, Y]]$	$[[Y, X], X]$ $[Y, [Y, X]]$	$[X, [X, Y]]$ $[Y, [X, Y]]$
4	$[X, [X, [X, Y]]]$ $[Y, [X, [X, Y]]]$ $[Y, [Y, [X, Y]]]$	$[[[Y, X], X], X]$ $[[Y, [Y, X]], X]$ $[Y, [Y, [Y, X]]]$	$[X, [X, [X, Y]]]$ $[X, [Y, [X, Y]]]$ $[Y, [Y, [X, Y]]]$
5	$[X, [X, [X, [X, Y]]]]$ $[Y, [X, [X, [X, Y]]]]$ $[Y, [Y, [X, [X, Y]]]]$ $[Y, [Y, [Y, [X, Y]]]]$ $[[X, Y], [X, [X, Y]]]$ $[[X, Y], [Y, [X, Y]]]$	$[[[[Y, X], X], X], X]$ $[[[Y, [Y, X]], X], X]$ $[[Y, X], [[Y, X], X]]$ $[[Y, [Y, [Y, X]]], X]$ $[[Y, [Y, X]], [Y, X]]$ $[Y, [Y, [Y, [Y, X]]]]$	$[X, [X, [X, [X, Y]]]]$ $[X, [X, [Y, [X, Y]]]]$ $[X, [Y, [Y, [X, Y]]]]$ $[Y, [X, [X, [X, Y]]]]$ $[Y, [X, [Y, [X, Y]]]]$ $[Y, [Y, [Y, [X, Y]]]]$
6	$[X, [X, [X, [X, [X, Y]]]]]$ $[Y, [X, [X, [X, [X, Y]]]]]$ $[Y, [Y, [X, [X, [X, Y]]]]]$ $[Y, [Y, [Y, [X, [X, Y]]]]]$ $[Y, [Y, [Y, [Y, [X, Y]]]]]$ $[[X, Y], [X, [X, [X, Y]]]]]$ $[[X, Y], [Y, [X, [X, Y]]]]]$ $[[X, Y], [Y, [Y, [X, Y]]]]]$ $[[X, [X, Y]], [Y, [X, Y]]]$	$[[[[[Y, X], X], X], X], X]$ $[[[[Y, [Y, X]], X], X], X]$ $[[[Y, X], [[Y, X], X]], X]$ $[[[Y, [Y, [Y, X]]], X], X]$ $[[[Y, [Y, X]], [Y, X]], X]$ $[[Y, X], [[Y, [Y, X]], X]]$ $[[Y, [Y, [Y, [Y, X]]]], X]$ $[[Y, [Y, [Y, X]], [Y, X]]]$ $[Y, [Y, [Y, [Y, [Y, X]]]]]$	$[X, [X, [X, [X, [X, Y]]]]]$ $[X, [X, [X, [Y, [X, Y]]]]]$ $[X, [X, [Y, [Y, [X, Y]]]]]$ $[X, [Y, [X, [X, [X, Y]]]]]$ $[X, [Y, [X, [Y, [X, Y]]]]]$ $[X, [Y, [Y, [Y, [X, Y]]]]]$ $[Y, [X, [Y, [Y, [X, Y]]]]]$ $[Y, [Y, [X, [Y, [X, Y]]]]]$ $[Y, [Y, [Y, [Y, [X, Y]]]]]$

gdzie $a_{i_1}, a_{i_2}, \dots, a_{i_t}$ należą do $\mathbb{T}_{\mathbb{A}}$, generują bazę $\mathcal{L}(\mathbb{A})$ [12].

Tabela 2.1 zawiera początkowe elementy bazy Chibrikova dla układu dwuwęściowego (alfabet dwuliterowy).

2.3.3. Podsumowanie

Pierwsza baza wolnej algebry Liego pojawiła się w pracach Ph. Halla z 1930 roku, następnie powstały bazy Lyndona, Shirshova. Viennot [46] pokazał, że wszystkie są specjalnymi przypadkami ogólnej bazy Halla. Nowszą prawounormowaną bazę zaprezentował Chibrikov. Wybór bazy Halla przez robotyków jest zatem raczej restrykcyjny i uwarunkowany historycznie. Jest to bowiem pierwsza baza algebry Liego, a zarazem najczęściej rozważana i wykorzystywana w literaturze. Niewielu bada warunek LARC za pomocą bazy Lyndona, chociaż jest oczywiste, że wynik byłby identyczny (zgod-

Tablica 2.2. Skumulowana liczba elementów baz algebry Liego (zgodnie z formułą Witt'a) o dwóch generatorach do warstwy 20. włącznie

warstwa	1	2	3	4	5	6	7	8	9	10
il. elem.	2	3	5	8	14	23	41	71	127	226
warstwa	11	12	13	14	15	16	17	18	19	20
il. elem.	412	747	1377	2538	4720	8800	16510	31042	58636	111013

nie z Tw. 5.1 [80] o równoważności baz Lyndona i Halla). Wybór definicji algebry także wpływa na sposób jej konstrukcji i implementację. Nawet wybór konkretnej bazy (w obrębie tej samej rodziny baz np. Halla) może mieć zalety implementacyjne, gdyż algorytmy je generujące i wykorzystujące mogą różnić się złożonością obliczeniową i poziomem skomplikowania. Oprócz zalet implementacyjnych, wybór innej bazy niż Halla może oferować dodatkowe atuty. Słowa Lyndona reprezentowane są jako zbiór liter z pewnego alfabetu, a zamiana sekwencji słów na nawiasy Liego odbywa się procedurą nawiasowania (ang. *bracketing*). Zatem z jednej strony, operacja na słowach (ciągach liter) wymaga mniej pamięci komputera, z drugiej – operacje na nawiasach Liego (o strukturze drzew binarnych) mogą być mniej kosztowne obliczeniowo. Zaletą słów Lyndona-Shirshova jest także fakt, że największy mnożnik po operacji nawiasowania jest natychmiast nawiasem Liego z zachowaniem kolejności występujących symboli. W bazie Lyndona zatem od razu otrzymujemy największy jednomian Liego tworzący dany nawias, natomiast jego wyliczenie w bazie Halla jest kosztowne.

2.3.4. Formuła Witt'a

Pierwsza baza wolnej algebry Liego została odkryta przez M. Halla [35], który pokazał, że proces gromadzenia stworzony przez Ph. Halla [37] zachowuje się jak baza. Od tego czasu wiadomo, że każdy element wolnej algebry Liego jest liniową kombinacją pewnych bazowych komutatorów. A.I. Shirshov [87] i R.C. Lyndon [62] skonstruowali swoje bazy algebry Liego, które zawierają tzw. słowa Lyndona-Shirshova. W. Magnus i E. Witt udowodnili, że algebra Liego otrzymana z wolnej łącznej algebry jest dalej wolna [5], prostszy dowód pokazał również Shirshov [87]. Dzięki tym результатам Witt podał wzór, który pozwala na wyliczenie liczby elementów bazy algebry Liego w kolejnych warstwach. Dla n -tej warstwy liczba ta wynosi [69]

$$v(n, s) = \frac{1}{n} \sum_{d|n} \mu(d) s^{n/d} \quad (2.13)$$

gdzie s oznacza liczbę generatorów, sumowanie następuje po wszystkich całkowitych dzielnikach d liczby n , a funkcja $\mu : \mathbb{N} \rightarrow \{-1, 0, 1\}$ jest zdefiniowana następująco: jeżeli

rozkład d na czynniki ma postać $d = p_1^{n_1} p_2^{n_2} \dots p_q^{n_q}$, $n_i > 0$, gdzie p_i są liczbami pierwszymi, to

$$\mu(d) = \begin{cases} 1 & \text{dla } d = 1 \\ (-1)^q & \text{gdy wszystkie } n_i = 1, \\ 0 & \text{w przeciwnym przypadku.} \end{cases} \quad (2.14)$$

Tabela 2.2 zawiera skumulowaną liczbę elementów kolejnych warstw wolnej algebry Liego dla dwóch generatorów (wg formuły Witt'a). Ich liczba w kolejnych warstwach rośnie bardzo szybko.

2.4. Algebra Liego a grupa Liego

Mając zadane elementy $g, h \in \mathcal{G}$ definiuje się lewostronne przesunięcie (lewą translację) $L_g : \mathcal{G} \rightarrow \mathcal{G}; h \rightarrow gh$, będące dyfeomorfizmem. Przekształcenie L_g indukuje liniowe przekształcenie przestrzeni stycznych $T_h L_g : T_h \mathcal{G} \rightarrow T_{gh} \mathcal{G}$ ($T_{\kappa} \mathcal{G}$ to przestrzeń styczna do \mathcal{G} w κ) będące izomorfizmem. Można teraz skonstruować gładkie pole wektorowe \mathbf{V} na \mathcal{G} przez wzięcie wektora $\boldsymbol{\xi} = \mathbf{V}(e)$ z przestrzeni stycznej $T_e \mathcal{G}$ w elemencie neutralnym e grupy \mathcal{G} i zaczepienie go w $T_e L_g$ następująco

$$\forall g \in \mathcal{G}: \quad \mathbf{V}_{\boldsymbol{\xi}}(g) = (T_e L_g) \mathbf{V}(e). \quad (2.15)$$

Pole wektorowe \mathbf{V} spełniające własność

$$(T_h L_g) \mathbf{V}(h) = \mathbf{V}(gh), \forall g, h \in \mathcal{G} \quad (2.16)$$

jest nazywane lewoniemienniczym. Zupełnie analogicznie przeprowadza się rozumowanie dla prawej translacji $R_g : \mathcal{G} \rightarrow \mathcal{G}; h \rightarrow hg$. Obydwa przekształcenia (lewa i prawa translacja) są ściśle ze sobą związane.

Niech $\mathcal{V}_L(\mathcal{G})$ oznacza zbiór lewoniemiennicznych pól wektorowych na grupie Liego \mathcal{G} . Wtedy $\mathcal{V}_L(\mathcal{G})$ wraz z lewoniemiennicznymi działaniami na $\mathcal{G} \cdot : \mathbb{R} \cdot \mathcal{V}_L(\mathcal{G}) \rightarrow \mathcal{V}_L(\mathcal{G})$, $+$: $\mathcal{V}_L(\mathcal{G}) + \mathcal{V}_L(\mathcal{G}) \rightarrow \mathcal{V}_L(\mathcal{G})$ oraz nawias Liego $[\cdot, \cdot] : [\mathcal{V}_L(\mathcal{G}), \mathcal{V}_L(\mathcal{G})] \rightarrow \mathcal{V}_L(\mathcal{G})$ tworzy algebrę Liego. Przestrzeń styczna $T_e \mathcal{G}$ w elemencie neutralnym grupy $e \in \mathcal{G}$ oraz algebra Liego $\mathcal{V}_L(\mathcal{G})$ są związane dwoma odwzorowaniami, pierwsza to transformacja (ang. *transplanatation map*)

$$t : T_e \mathcal{G} \rightarrow \mathcal{V}_L(\mathcal{G}), \quad \boldsymbol{\xi} \mapsto \mathbf{V}_{\boldsymbol{\xi}},$$

oraz przekształcenie nadające wartość (ang. *evaluation map*)

$$e : \mathcal{V}_L(\mathcal{G}) \rightarrow T_e \mathcal{G}, \quad \mathbf{V} \mapsto \mathbf{V}(e).$$

Ponadto, przestrzeń styczna $T_e \mathcal{G}$ (nazywana algebrą Liego \mathcal{L} grupy \mathcal{G}) dziedziczy z $\mathcal{V}_L(\mathcal{G})$ strukturę algebry Liego przez indukowany nawias

$$[\boldsymbol{\psi}, \boldsymbol{\eta}] = [\mathbf{V}_{\boldsymbol{\psi}}, \mathbf{V}_{\boldsymbol{\eta}}](e).$$

Bijekcja między $T_e\mathcal{G}$ a $\mathcal{V}_1(\mathcal{G})$, rozszerzona do jednoparametrowej podgrupy $\phi : \mathbb{R} \rightarrow \mathcal{G}$, $t \mapsto \phi(t)$, i lewonezmiennicza \mathbb{R} -akcja $\Phi : \mathbb{R} \times \mathcal{G} \rightarrow \mathcal{G}$, $(t, g) \mapsto g\phi(t)$ są uważane za jedne z największych osiągnięć teorii Liego. Wiele własności grupy Liego \mathcal{G} jest odzwierciedlonych w algebraicznej strukturze łącznej algebry Liego \mathcal{L} . Dla spójnych grup Liego zgodność rozszerza się tak dalece, że grupa Liego może być odzyskana z algebry Liego za pomocą izomorfizmu. Rozważmy odwzorowanie łączące algebrę Liego z identycznościowym elementem odpowiadającej grupy Liego. Niech $\phi_\xi : \mathbb{R} \rightarrow \mathcal{G}$ będzie krzywą całkową lewonezmienniczego pola wektorowego \mathbf{V}_ξ , $\xi \in \mathcal{L}$, która przechodzi w chwili $t = 0$ przez identyczność $e \in \mathcal{G}$. Przekształcenie ϕ_ξ będące homomorfizmem z \mathbb{R} w \mathcal{G} dla wszystkich $\xi \in \mathcal{G}$ stanowi jednoparametrową podgrupę \mathcal{G} . Przekształcenie wykładnicze (ang. *exponential map*) jest zdefiniowane jako

$$\exp : \mathcal{L} \rightarrow \mathcal{G}; \quad \exp \xi = \phi_\xi(1). \quad (2.17)$$

Łatwo sprawdzić, że $\exp(t\xi) = \phi_\xi(t)$, czyli linie $t\xi$, $t \in \mathbb{R}$, $\xi \in \mathcal{L}$ przemieszczające się ze źródła \mathcal{L} , są przekształcane na jednoparametrowe podgrupy \mathcal{G} . I odwrotnie, każda jednoparametrowa podgrupa \mathcal{G} może być wyrażona jako $\exp(t\xi)$ dla pewnego $\xi \in \mathcal{L}$. Ponieważ różniczka $d(\exp) : T_e\mathcal{G} \rightarrow T_e\mathcal{G}$ jest identycznością na $T_e\mathcal{G}$, skąd można wyciągnąć wniosek (dzięki tw. o funkcji odwrotnej i gładkości \exp), że \exp jest lokalnym dyfeomorfizmem z otoczenia $U \ni \mathbf{0}$ w \mathcal{L} na otoczenie V identyczności $e \in \mathcal{G}$. Odwrotne przekształcenie do \exp z V do U oznaczone jest przez $\log : \mathcal{G} \rightarrow \mathcal{L}$.

Zwykle odwzorowanie \exp nawet dla spójnych grup Liego nie jest ani surjekcją, ani injekcją. Istnieje wynik pokazujący odpowiednio duże otoczenie $\mathbf{0} \in \mathcal{L}$, gdzie przekształcenie wykładnicze jest bijekcją [104]. Dlatego dla zadanej bazy ξ_1, \dots, ξ_n w \mathcal{L} istnieje rozsądnie duże otoczenie $\mathbf{0} \in U \subset \mathbb{R}^n$ w którym przekształcenie

$$\phi : U \rightarrow \mathcal{G}; \quad (a_1, \dots, a_n) \mapsto \exp(a_1\xi_1 + \dots + a_n\xi_n) \quad (2.18)$$

jest gładkim dyfeomorfizmem na otwarty podzbiór $e \in \phi(U) \subset \mathcal{G}$. Wtedy gładkie funkcje x_1, \dots, x_n na V spełniające

$$x_i \exp(a_1\xi_1 + \dots + a_n\xi_n) = a_i, \quad i = 1, \dots, n; \quad \forall \mathbf{a} = (a_1, \dots, a_n)^T \in U \quad (2.19)$$

są nazywane kanonicznymi współrzędnymi pierwszego rodzaju wokół $e \in \mathcal{G}$ w bazie ξ_1, \dots, ξ_n . Inne kanoniczne współrzędne bazują na fakcie, że istnieje otoczenie $U \in \mathbb{R}^n$ w którym przekształcenie

$$\psi : U \rightarrow \mathcal{G}; \quad (a_1, \dots, a_n) \mapsto \exp(a_1\xi_1) \cdots \exp(a_n\xi_n) \quad (2.20)$$

jest dyfeomorfizmem na sąsiedztwo $\psi(U)$ identyczności \mathcal{G} . Ponieważ każde $g \in \mathcal{G}$ może być zapisane jako skończony iloczyn $g_1 g_2 \cdots g_k$ elementów grupy \mathcal{G} w arbitralnym otoczeniu $e \in V \subset \mathcal{G}$, to ψ jest surjekcją, jeśli $U = \mathbb{R}^n$. Gładkie funkcje x_1, \dots, x_n na $\psi(U)$ spełniające

$$x_i \exp(a_1\xi_1) \cdots \exp(a_n\xi_n) = a_i, \quad i = 1, \dots, n, \quad \forall \mathbf{a} = (a_1, \dots, a_n)^T \in U \quad (2.21)$$

są nazywane kanonicznymi współrzędnymi drugiego rodzaju wokół $e \in \mathcal{G}$ w bazie ξ_1, \dots, ξ_n [95].

2.4.1. Przykłady macierzowych grup Liego i ich algebr

Najbardziej ogólną macierzową grupą Liego jest $GL(n, \mathbb{R})$ (ang. *general linear group*) ($GL(n, \mathbb{C})$) odwracalnych macierzy rozmiaru $(n \times n)$ z rzeczywistymi (zespólnymi) elementami, w skrócie zapisywana jako $GL(n)$. Jako przeciwobraz $\mathbb{R} \setminus \{0\}$ ciągłego przekształcenia $\mathbf{X} \mapsto \det(\mathbf{X})$, $GL(n)$ jest otwartym podzbiorem $M_{n,n} \cong \mathbb{R}^{n^2}$, a zatem można jej nadać strukturę gładkiej rozmaitości. Grupowe mnożenie w $GL(n)$ jest mnożeniem macierzy, a przekształcenie odwrotne zamienia macierz $\mathbf{X} \in GL(n)$ na jej odwrotność \mathbf{X}^{-1} . Elementem neutralnym $e \in GL(n)$ jest macierz jednostkowa rozmiaru $(n \times n)$. Ponieważ $GL(n)$ jest otwartym podzbiorem wszystkich macierzy $(n \times n)$, $M_{n,n}$, algebrą Liego $\mathfrak{gl}(n)$ grupy $GL(n)$ będzie $M_{n,n}$ z nawiasem Liego

$$[\mathbf{X}, \mathbf{Y}] = \mathbf{X}\mathbf{Y} - \mathbf{Y}\mathbf{X}. \quad (2.22)$$

Inne macierzowe grupy Liego są podgrupami $GL(n)$, z operacjami dziedzicznymi z $GL(n)$. Znane podgrupy $GL(n)$ i ich algebry są następujące:

1. specjalna grupa liniowa $SL(n) = \{\mathbf{X} \in GL(n) | \det(\mathbf{X}) = 1\}$ o algebrze Liego $\mathfrak{sl}(n)$ zadanej przez $\mathfrak{sl}(n) = \{\mathbf{A} \in \mathfrak{gl}(n) | \text{trace}(\mathbf{A}) = 0\}$,
2. grupa macierzy ortogonalnych $O(n)$ opisuje zależność $O(n) = \{\mathbf{X} \in GL(n) | \mathbf{X}^T \mathbf{X} = \mathbf{I}\}$, a algebrę Liego $\mathfrak{o}(n) = \{\mathbf{A} \in \mathfrak{gl}(n) | \mathbf{A}^T = -\mathbf{A}\}$,
3. specjalna grupa obrotów reprezentująca obroty w n -wymiarowej przestrzeni euklidesowej $SO(n) = \{\mathbf{X} \in GL(n) | \mathbf{X} \in O(n) \cap SL(n)\}$ Ponieważ $SO(n)$ jest komponentą identyczności w $O(n)$, więc algebra Liego $\mathfrak{so}(n)$ jest równa $\mathfrak{o}(n)$,
4. specjalna grupa euklidesowa ruchu ciała sztywnego

$$\begin{aligned} SE(n) &= \left\{ \begin{pmatrix} \mathbf{X} & \mathbf{p} \\ 0^{1 \times n} & 1 \end{pmatrix} \in GL(n+1) | \mathbf{X} \in SO(n), \mathbf{p} \in \mathbb{R}^n \right\}, \text{ o algebrze Liego zdefinio-} \\ &\text{wanej przez } \mathfrak{se}(n) = \left\{ \begin{pmatrix} \mathbf{A} & \mathbf{p} \\ 0^{1 \times n} & 0 \end{pmatrix} \in \mathfrak{gl}(n+1) | \mathbf{A} \in \mathfrak{so}(n), \mathbf{p} \in \mathbb{R}^n \right\}. \end{aligned}$$

2.4.2. Formalna algebra Liego

Na użytek dysertacji niezbędne są następujące definicje (notacja pochodzi z pracy [54]). Niech $\mathbf{X}_1, \dots, \mathbf{X}_m$ oznaczają symbole formalne – generatory, natomiast ich zbiór oznaczony jest przez $\mathbb{X}_m = \{\mathbf{X}_1, \dots, \mathbf{X}_m\}$. Używamy zapisu $\mathcal{A}(\mathbb{X}_m)$ do oznaczenia algebry niekomutujących wielomianów z \mathbb{X}_m . Dla uproszczenia notacji definiujemy potęgę jednomianu w naturalny sposób $\mathbf{A}^k = \mathbf{A} * \mathbf{A}^{k-1}$, $\mathbf{A}^1 = \mathbf{A}$.

Przykład 5. Niech $m = 2$ z generatorami formalnymi $\mathbf{X}_1 = \mathbf{X}$, $\mathbf{X}_2 = \mathbf{Y}$. Wtedy niekomutującymi jednomianami (tworzącymi nieskończoną bazę) $\mathcal{A}(\mathbb{X}_2)$ są \mathbf{I} , \mathbf{X} , \mathbf{Y} , \mathbf{X}^2 , \mathbf{XY} , \mathbf{YX} , \mathbf{Y}^2 , \mathbf{X}^3 , $\mathbf{X}^2\mathbf{Y}$, \mathbf{XYX} , \mathbf{XY}^2 , \mathbf{YX}^2 , ...

Generatory są symbolami formalnymi, więc nie przypisuje się im żadnej interpretacji fizycznej, czy to macierzy czy pól wektorowych. Nawet jeśli utożsamia się generatory formalne z generatorami pewnego układu sterowania i postać układu wskazuje

na zależności pomiędzy generatorami układu, to ich formalne odpowiedniki są zawsze niezależne.

Przykład 6. Niech pola wektorowe – generatory \mathbf{g}_i , $i = 1, \dots, m$ mają odpowiedniki w symbolach formalnych \mathbf{X}_i , $i = 1, \dots, m$. Jeśli $\mathbf{g}_2 = 2\mathbf{g}_1$, to $\mathbf{X}_1, \mathbf{X}_2$ są niezależne.

Oznaczmy przez $\mathcal{L}(\mathbb{X}_m)$ podalgebrę $\mathcal{A}(\mathbb{X}_m)$ generowaną przez \mathbb{X}_m . Jeśli zdefiniuje się formalny nawias Liego zależnością

$$[\mathbf{X}, \mathbf{Y}] = \mathbf{X} * \mathbf{Y} - \mathbf{X} * \mathbf{Y} = \mathbf{X}\mathbf{Y} - \mathbf{Y}\mathbf{X}, \quad (2.23)$$

wtedy $\mathcal{L}(\mathbb{X}_m)$ tworzy algebrę Liego, której elementy są zwane wielomianami Liego. Jednomiany Liego są generowane rekurencyjnie. Są nimi generatory formalne i wszystkie obiekty utworzone z jednomianów Liego przy pomocy formalnego nawiasu Liego (2.23). Nawias Liego może być wyspecyfikowany: w macierzowej grupie Liego przyjmuje identyczną postać (2.22), gdzie $*$ oznacza mnożenie macierzy. Dla algebry Liego pól wektorowych, nawias Liego pól \mathbf{f}, \mathbf{h} , we współrzędnych \mathbf{q} , definiuje się wzorem

$$[\mathbf{f}, \mathbf{h}] = \frac{\partial \mathbf{h}}{\partial \mathbf{q}} \mathbf{f} - \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \mathbf{h}. \quad (2.24)$$

Dla układów nilpotentnych nawiasy Liego od pewnej warstwy zerują się tożsamościowo

$$\mathbb{W}_p \neq \mathbb{W}_{p+1} = \{\mathbf{0}\}. \quad (2.25)$$

gdzie p to rząd nilpotentności. Nilpotentne wersje przedstawionych struktur, zapisywane jako $\mathcal{A}_k(\mathbb{X}_m)$ (wolna nilpotentna łączna algebra rzędu k), $\mathcal{L}_k(\mathbb{X}_m)$ (wolna nilpotentna algebra Liego rzędu k), są zdefiniowane przez wyzerowanie wszystkich elementów stopnia $k + 1$ i wyższych.

Przykład 7. Skończoną bazę $\mathcal{A}_2(\mathbb{X}_2)$ tworzą następujące elementy: $1, \mathbf{X}, \mathbf{Y}, \mathbf{XX}, \mathbf{XY}, \mathbf{YX}, \mathbf{YY}$. Oczywiście, $\mathbf{XXX} = \mathbf{XXY} = \mathbf{XYX} = \mathbf{XYY} = \mathbf{YXX} = \dots = 0$.

Nie wszystkie jednomiany Liego są niezależne, gdyż zachodzą własności (2.1)-(2.3). Zbiór niezależnych jednomianów Liego tworzy bazę.

Przykład 8. Przykładowa baza $\mathcal{L}_2(\mathbb{X}_2)$ składa się z $\mathbf{X}, \mathbf{Y}, [\mathbf{X}, \mathbf{Y}]$. Wykluczony jest element $[\mathbf{Y}, \mathbf{X}] = -[\mathbf{X}, \mathbf{Y}]$.

Indeks jednomianu Liego \mathbf{A} , oznaczany jako $\text{ind}(\mathbf{A})$, jest wektorem o liczbie składowych równej liczbie generatorów formalnych. Kolejne składowe indeksu wskazują liczbę generatorów występujących w \mathbf{A} .

Przykład 9. Dla dwóch generatorów \mathbf{X}, \mathbf{Y} , $\text{ind}([\mathbf{Y}, \mathbf{X}], [[\mathbf{X}, \mathbf{Y}], \mathbf{Y}]) = (2, 3)$ oraz dla trzech generatorów $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$, $\text{ind}([\mathbf{X}, [\mathbf{Z}, [[\mathbf{Z}, \mathbf{Y}], \mathbf{Y}]]]) = (1, 2, 2)$, a także $\text{ind}([\mathbf{Z}, [\mathbf{X}, \mathbf{Z}]]]) = (1, 0, 2)$.

Zachodzi oczywista równość

$$\deg(\mathbf{A}) = \sum_i \text{ind}(\mathbf{A})_i, \quad (2.26)$$

gdzie oznacza $\sum_i \text{ind}(\mathbf{A})_i$ sumę składowych wektora indeksu.

Dla niekomutujących jednomianów ich stopień zdefiniowany jest jako $\deg(\mathbf{A} * \mathbf{B}) = \deg(\mathbf{A}) + \deg(\mathbf{B})$, natomiast indeks definiujemy analogicznie do indeksu jednomianu Liego.

Przykład 10. Dla generatorów \mathbf{X}, \mathbf{Y} , jednomian $\mathbf{A} = \mathbf{X}^2 * \mathbf{Y} = \mathbf{X} * \mathbf{X} * \mathbf{Y} = \mathbf{X}\mathbf{X}\mathbf{Y} = \mathbf{X}^2\mathbf{Y}$ posiada $\deg(\mathbf{A}) = 3$ oraz $\text{ind}(\mathbf{A}) = (2, 1)$.

Stopień może być przypisany nie tylko jednomianom Liego, ale także obiektom, których są one częściami: $\deg(a(t)\mathbf{A}) = \deg(\dot{a}(t)\mathbf{A}) = \deg(\mathbf{A})$, gdzie $a(t), \dot{a}(t)$ jest funkcją i jej pochodną, a \mathbf{A} jest jednomianem Liego. Funkcja dziedziczy stopień od jednomianu Liego, który mnoży. Stopień jednomianu będącego iloczynem elementarnych zmiennych definiujemy naturalnie jako sumę stopni poszczególnych składowych uwzględniając ich krotność.

Przykład 11. Jeśli, $\deg(a_1) = 3$, $\deg(a_2) = 5$, $\deg(a_3) = 1$, więc $\deg(a_1^2 a_2 a_3^4) = 15$.

Przekształcenie wykładnicze e (oznaczane także jako \exp) przekształca element algebry Liego w element grupy Liego $e : \mathbf{A} \rightarrow e^{\mathbf{A}}$. Dla układów sterowania odwzorowanie e ma fizyczną interpretację, bowiem w stanie \mathbf{q} chwilowa prędkość \mathbf{A} (element algebry Liego) determinuje kolejne położenie układu (element grupy Liego) $\mathbf{q}e^{t\mathbf{A}}$ (strumień pola wektorowego \mathbf{A}). Najczęściej \mathbf{A} jest skalowane zależną od sterowań (a zatem czasu) funkcją $a(t)$, wtedy otrzymujemy jednoparametrową podgrupę Liego $e^{a(t)\mathbf{A}}$, którą można rozwinąć w szereg Taylora

$$e^{a(t)\mathbf{A}} = \sum_{i=0}^{\infty} \frac{1}{i!} (a(t)\mathbf{A})^i = \mathbf{I} + a(t)\mathbf{A} + \frac{1}{2}a^2(t)\mathbf{A}^2 + \frac{1}{6}a^3(t)\mathbf{A}^3 + \dots \quad (2.27)$$

Dla nilpotentnych grup Liego, szereg (2.27) ma skończoną liczbę elementów. Dla przekształcenia wykładniczego zdefiniowane jest także różniczkowanie

$$\frac{d}{dt} e^{a(t)\mathbf{A}} = \dot{a} e^{a(t)\mathbf{A}} \mathbf{A} = \dot{a} \mathbf{A} e^{a(t)\mathbf{A}}, \quad (2.28)$$

oraz (korzystając z własności grupowych) odwrotność $e^{a(t)\mathbf{A}}$ i odwrotność iloczynu strumieni

$$(e^{a(t)\mathbf{A}})^{-1} = e^{-a(t)\mathbf{A}}, \quad (e^{a(t)\mathbf{A}} e^{b(t)\mathbf{B}})^{-1} = e^{-b(t)\mathbf{B}} e^{-a(t)\mathbf{A}}. \quad (2.29)$$

2.4.3. Formuła Campbella-Bakera-Hausdorffa-Dynkina

Dla każdego formalnego wielomianu Liego Z jest zdefiniowany odpowiadający mu formalny szereg potęgowy Liego (2.27). Dzięki результатам Campbella, Bakera i Hausdorffa, wiadomo, że zbiór szeregów potęgowych jest zamknięty ze względu na operację mnożenia oraz $e^Z e^{-Z} = \mathbf{I} = e$, zatem tworzy grupę. Można więc zapisać iloczyn

$$e^{Z_1} e^{Z_2} = e^{Z_3}$$

jednoznacznie określający Z_3 formułą (której ostateczną postać nadał Dynkin [30])

$$Z_3 = \sum_{m=1}^{\infty} \sum \frac{(-1)^{m-1} (\text{ad}_{Z_2})^{q_m} (\text{ad}_{Z_1})^{p_m} \dots (\text{ad}_{Z_2})^{q_1} (\text{ad}_{Z_1})^{p_1}}{m \sum_{i=1}^m (p_i + q_i) \prod_{i=1}^m (p_i! q_i!)}, \quad (2.30)$$

gdzie wewnętrzna suma jest wyliczana po wszystkich m -krotkach par nieujemnych liczb całkowitych (p_i, q_i) takich, że $p_i + q_i > 0$; symbol ad_X wprowadza przekształcenie $\text{ad}_X : Y \rightarrow [X, Y]$. Ostatni element we wzorze (2.30) jest rozumiany inaczej, a wyliczany następująco: $\text{ad}_Z^n = 1$ gdy $n = 0$, $\text{ad}_Z^n = Z$ gdy $n = 1$, oraz $\text{ad}_Z^n = 0$ gdy $n > 1$.

Wzór (2.30) jest określany jako (dyskretna) formuła Campbella-Bakera-Hausdorffa-Dynkina, w skrócie CBHD. Można pokazać, że zgodnie z (2.30) jest

$$e^p e^q = e^{p+q + \frac{1}{2}[p, q] + \frac{1}{12}[[p, q], q] - \frac{1}{12}[[p, q], p] + \frac{1}{24}[p, [q, [p, q]]] + \dots}. \quad (2.31)$$

Jak łatwo zauważyć, zwykle $e^p e^q \neq e^{p+q}$. Dla nilpotentnych grup Liego formuła CBHD generuje szereg skończony. Regułę CBHD zwykle przedstawia się także w postaci [17]:

$$e^p e^q e^{-p} = \exp \left(\sum_{i=0}^{\infty} \frac{\text{ad}_p^i q}{i!} \right), \quad (2.32)$$

gdzie $\exp(x) = e^x$, $\text{ad}_p^0 q = q$ oraz $\text{ad}_p^i q = [p, \text{ad}_p^{i-1} q]$.

2.5. Uogólniona formuła Campbella-Bakera-Hausdorffa-Dynkina

Uogólniona formuła Campbella-Bakera-Hausdorffa-Dynkina (gCBHD) opisuje (lokalnie) trajektorię nieautonomicznego układu równań różniczkowych z zadanyim warunkiem początkowym [93]

$$\dot{\mathbf{q}} = \mathbf{H}(t)(\mathbf{q}(t)) \quad \mathbf{q}(0) = \mathbf{q}_0 \quad (2.33)$$

gdzie $\mathbf{H}(t)(\cdot)$ jest rodziną analitycznych pól wektorowych zależnych od czasu. Rozwiązanie (2.33) postaci

$$\mathbf{q}(t) = \exp(\text{sz}(t))(\mathbf{q}_0)|_{s=1} = e^{\text{sz}(t)}(\mathbf{q}_0)|_{s=1} \quad (2.34)$$

dla czasów bliskich zeru wyraża się szeregiem

$$\mathbf{z}(t)(\mathbf{q}) = \sum_{r=1}^{\infty} \sum_{\sigma \in \mathbb{P}_r} \frac{(-1)^{\text{err}(\sigma)}}{r^2 \binom{r-1}{\text{err}(\sigma)}} \int_{\mathbb{T}_r(t)} [\dots [[\mathbf{H}(s_{\sigma(1)}), \mathbf{H}(s_{\sigma(2)}), \dots], \mathbf{H}(s_{\sigma(r)})] ds, \quad (2.35)$$

gdzie:

wartości $\mathbf{H}(s)$ wyliczamy w punkcie \mathbf{q} ,

\mathbb{P}_r jest zbiorem permutacji elementów $1, \dots, r$, przykładowo: $\mathbb{P}_2 = \{(1, 2), (2, 1)\}$, $\mathbb{P}_3 = \{(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1)\}$

$\text{err}(\sigma)$, $\sigma \in \mathbb{P}_r$ jest liczbą nieporządków w permutacji $\sigma = \{\sigma(1), \sigma(2), \dots, \sigma(r)\}$. Liczba nieporządków rośnie o jeden, gdy następny element permutacji jest mniejszy od bieżącego. Przykładowo: $\text{err}((1, 2, 3, 4)) = 0$, $\text{err}((1, 2, 4, 3)) = 1$, $\text{err}((4, 2, 3, 1)) = 2$,

$\mathbb{T}_r(t)$ to r -wymiarowy sympleks $\mathbb{T}_r(t) = \{s \in \mathbb{R}^r : 0 \leq s_1 \leq s_2 \leq s_3 \leq \dots \leq s_r \leq t\}$.

Uogólniona formuła CBHD może być interpretowana jako operator zmiany stanu, który dla krótkich czasów t przemieszcza stan bieżący \mathbf{q}_0 układu (2.33) do stanu $\mathbf{q}_0 + \mathbf{z}(t)(\mathbf{q}_0)$, a więc $\mathbf{z}(t)(\mathbf{q}_0)$ jest kierunkiem ruchu w punkcie \mathbf{q}_0 . Jeśli $\mathbf{H}(t)(\mathbf{q}(t))$ we wzorze (2.33) zależy od sterowań, to formułę (2.35) można wykorzystać do generacji takich sterowań, które wymuszają (lokalnie) ruch w zadanym kierunku. Ten sposób wykorzystania gCBHD jest preferowany w zadaniach planowania ruchu.

2.6. Zadanie planowania ruchu

Rozważmy bezdryfowy układ sterowania

$$\dot{\mathbf{q}} = \mathbf{g}_1(\mathbf{q})u_1 + \mathbf{g}_2(\mathbf{q})u_2 + \dots + \mathbf{g}_m(\mathbf{q})u_m = \sum_{i=1}^m \mathbf{g}_i(\mathbf{q})u_i, \quad (2.36)$$

gdzie:

m – liczba sterowań,

n – liczba zmiennych wektora konfiguracji, $m < n$,

\mathbf{q} – wektor konfiguracji $\mathbf{q} = (q_1 \dots q_n)^T \in \mathbf{Q} \subset \mathbb{R}^n$,

\mathbf{Q} – oznacza przestrzeń konfiguracyjną,

u_i – i -te sterowanie,

$\mathbf{g} = \{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_m\}$ – pola wektorowe klasy C^∞ stowarzyszone z układem (2.36), zwane generatorami. Pole wektorowe jest funkcją przypisującą każdej konfiguracji wektor. Zadanie planowania ruchu (ang. *Motion Planning Problem*) polega na znalezieniu algorytmu generującego sterowania $\mathbf{u}(t) = (u_1(t), \dots, u_m(t))^T$ przemieszczające układ między dowolną parą konfiguracji brzegowych $\mathbf{q}_0, \mathbf{q}_f$. By zadanie było rozwiązywalne, układ (2.36) musi być sterowalny, a warunek na tę własność podał Chow [13]. Jeżeli w każdym punkcie przestrzeni stanu algebra Liego układu (2.36) rozpina przestrzeń stanu, to układ jest sterowalny w krótkim czasie. Zastosowanie twierdzenia Chow spro-

wadza się do określenia rzędu algebry sterowalności układu. Jeżeli jest pełny, to układ jest sterowalny w krótkim czasie (jest to warunek mocniejszy od sterowalności):

$$\forall \mathbf{q} \quad \text{rank}(\mathcal{L}(\mathbf{G})(\mathbf{q})) = n. \quad (2.37)$$

Dlatego też twierdzenie Chow jest często nazywane warunkiem rzędu (LARC - ang. *Lie algebra rank condition*). W zależności (2.37) algebrę Liego można zamienić przez każdą z jej baz.

Przykład 12. *Algebra Liego (sterowalności) dla jednokołowego robota mobilnego o modelu (A.3), jest rozpięta przez pola $\mathbf{g}_1, \mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2] = (\sin q_3, -\cos q_3, 0)^\top$*

$$\mathcal{L}(\mathbf{G})(\mathbf{q}) = \text{span}\{(\cos q_3, \sin q_3, 0)^\top, (0, 0, 1)^\top, (\sin q_3, -\cos q_3, 0)^\top\}. \quad (2.38)$$

Wyznacznik macierzy utworzonej z pól rozpinających wynosi 1, a zatem pola rozpinają przestrzeń konfiguracyjną w każdym punkcie.

W dalszej części dysertacji zakładamy, że układ (2.36) spełnia warunek LARC, więc jest sterowalny w krótkim czasie.

3. Algorytm Lafferriera-Sussmanna

Algorytm Lafferriera-Sussmanna (LS) [53, 54, 55] jest Lie algebraiczną metodą planowania ruchu między zadanymi konfiguracjami \mathbf{q}_0 i \mathbf{q}_f , przeznaczoną dla bezdryfowych, nilpotentnych, sterowalnych układów sterowania (2.36). Szczegółowy opis metody LS zamieszczono także w książce [60], a jej implementacji w pakiecie Mathematica poświęcono pracę [52]. W niniejszym rozdziale przedstawimy, zorganizowane w podrozdziały, kolejne kroki metody opatrując je komentarzami i przykładami.

3.1. Układ rozszerzony – trajektoria referencyjna

Dla układu podstawowego (2.36) wyznaczany jest układ rozszerzony postaci

$$\dot{\mathbf{q}} = \mathbf{g}_1(\mathbf{q})v_1 + \cdots + \mathbf{g}_m(\mathbf{q})v_m + \mathbf{g}_{m+1}(\mathbf{q})v_{m+1} + \cdots + \mathbf{g}_r(\mathbf{q})v_r, \quad (3.1)$$

gdzie: v_i oznacza i -te sterowanie rozszerzone (wirtualne), $\mathbf{g} = \{\mathbf{g}_{m+1}, \dots, \mathbf{g}_r\}$ są polami wektorowymi wyższych stopni (większych od 1), utworzonymi z generatorów $\{\mathbf{g}_1, \dots, \mathbf{g}_m\}$ za pomocą nawiasu Liego (2.24). Z założenia o sterowalności układu (2.36), pola \mathbf{g}_i , $i = 1, \dots, r$ rozpinają przestrzeń stanu \mathbb{R}^n w każdym punkcie, czyli spełniają warunek rzędu LARC (2.37). Minimalna liczba pól spełniających warunek rzędu wynosi $r = n$. Jeżeli pola niskich stopni stają się zależne w pewnym punkcie, to $r > n$. Jeśli pewne pole k -tego stopnia wchodzi w skład pól spełniających warunek rzędu LARC, to także wszystkie pola k -tej warstwy muszą wystąpić w układzie (3.1), jakkolwiek nie są one niezbędne do spełnienia warunku rzędu. Dla układów nilpotentnych warunkiem determinującym liczbę warstw pól wektorowych występujących w układzie rozszerzonym (a zatem także liczbę r wyznaczaną za pomocą formuły Witt'a (2.13)) jest stopień nilpotentności układu (2.36).

Następnie wybierana jest dowolna, przynajmniej raz różniczkowalna, krzywa w przestrzeni konfiguracyjnej $\boldsymbol{\lambda}(t)$, $t \in [0, T]$ łącząca zadany punkt początkowy $\mathbf{q}_0 = \boldsymbol{\lambda}(0)$ z punktem końcowym $\mathbf{q}_f = \boldsymbol{\lambda}(T)$, która zwana jest *trajektoria referencyjną*. Najczęściej $\boldsymbol{\lambda}(\cdot)$ jest wybierana w postaci łatwo parametryzowalnej krzywej (np. funkcja liniowa, lub wielomianowa niskiego stopnia). Trajektoria referencyjna nie musi, i najczęściej nie jest, krzywą po której ewoluuje układ w przestrzeni konfiguracyjnej. Podstawiając trajektorię referencyjną $\boldsymbol{\lambda}(t) = \mathbf{q}(t)$ do (3.1) powstaje równanie

$$\dot{\boldsymbol{\lambda}}(t) = [\mathbf{g}_1(\boldsymbol{\lambda}(t)), \dots, \mathbf{g}_r(\boldsymbol{\lambda}(t))](v_1(t), \dots, v_r(t))^T = \mathbf{G}_{\text{ext}}(\boldsymbol{\lambda}(t))\mathbf{v}(t), \quad (3.2)$$

z którego wyznaczone są sterowania rozszerzone $\mathbf{v}(\cdot)$. Gdy $r > n$, z wykorzystaniem pseudo-odwrotności Moore'a-Penrose'a macierzy \mathbf{G}_{ext} , gdy $r = n$ pseudo-odwrotność staje się odwrotnością macierzy

$$\mathbf{v}(t) = \mathbf{G}_{\text{ext}}^{\#}(\boldsymbol{\lambda}(t))\dot{\boldsymbol{\lambda}}(t), \quad \text{gdzie} \quad \mathbf{G}_{\text{ext}}^{\#}(\boldsymbol{\lambda}(t)) = \mathbf{G}_{\text{ext}}^{\text{T}}(\boldsymbol{\lambda}(t)) \left(\mathbf{G}_{\text{ext}}(\boldsymbol{\lambda}(t))\mathbf{G}_{\text{ext}}^{\text{T}}(\boldsymbol{\lambda}(t)) \right)^{-1}. \quad (3.3)$$

Dzięki założonej sterowalności układu, zawsze istnieje (pseudo-)odwrotność macierz \mathbf{G}_{ext} . Rozwiązanie (3.3) opisuje chwilowe wartości sterowań wirtualnych odpowiadających krzywej $\boldsymbol{\lambda}(t)$ na przedziale czasu $t \in [0, T]$.

3.2. Równanie Chena-Fliessa-Sussmanna (CFS)

Drugim etapem algorytmu LS jest formalizacja układu rozszerzonego (3.1). Pola wektorowe \mathbf{g}_i , $i = 1, \dots, r$ (elementy algebry Liego sterowalności) są reprezentowane przez ich formalne odpowiedniki \mathbf{B}_i (elementy bazy wolnej algebry Liego). Polom – generatorom \mathbf{g}_i , $i = 1, \dots, m$ odpowiadają generatory bazy wolnej algebry Liego. Ewolucję w czasie elementów grupy Liego $\mathbf{S}(t)$ pod wpływem sterowań rozszerzonych opisuje równanie różniczkowe

$$\dot{\mathbf{S}}(t) = \mathbf{S}(t)(v_1(t)\mathbf{B}_1 + v_2(t)\mathbf{B}_2 + \dots + v_{r-1}(t)\mathbf{B}_{r-1} + v_r(t)\mathbf{B}_r), \quad (3.4)$$

z warunkiem początkowym $\mathbf{S}(0) = \mathbf{I}$, który oznacza, że w chwili $t = 0$ układ jest w zadanej konfiguracji początkowej. Rozwiązanie równania (3.4) dla dowolnej chwili $t > 0$ może być zapisane za pomocą szeregu Chena-Fliessa i literaturowych reprezentacji ruchu „wstecz”

$$\mathbf{S}(t) = e^{h_r(t)\mathbf{B}_r} e^{h_{r-1}(t)\mathbf{B}_{r-1}} \dots e^{h_2(t)\mathbf{B}_2} e^{h_1(t)\mathbf{B}_1}, \quad (3.5)$$

lub „w przód”

$$\mathbf{S}(t) = e^{h_1(t)\mathbf{B}_1} e^{h_2(t)\mathbf{B}_2} \dots e^{h_{r-1}(t)\mathbf{B}_{r-1}} e^{h_r(t)\mathbf{B}_r}, \quad (3.6)$$

gdzie $e^{h_i(t)\mathbf{B}_i}$ jest strumieniem bazowego jednomianu Liego \mathbf{B}_i działającym przez czas $h_i(t)$. Reprezentacja „wstecz” $\mathbf{S}(t)$ opisuje konkatenację ruchów elementarnych: najpierw zachodzi ruch wzdłuż \mathbf{B}_r przez czas $h_r(t)$, następnie wzdłuż \mathbf{B}_{r-1} przez czas $h_{r-1}(t)$, by zakończyć ruchem wzdłuż \mathbf{B}_1 . Dla znanych już sterowań wirtualnych $\mathbf{v}(\cdot)$ i przyjętej reprezentacji ruchu należy wyznaczyć z równania (3.4) wartość $\mathbf{S}(T)$, czyli de facto współczynniki $h_i(T)$, $i = 1, \dots, r$, odpowiadającą osiągnięciu konfiguracji docelowej \mathbf{q}_f . Do wyznaczenia współczynników $\mathbf{h}(T) = (h_1(T), \dots, h_r(T))^{\text{T}}$, równanie (3.4) przekształca się do postaci

$$\mathbf{S}^{-1}(t)\dot{\mathbf{S}}(t) = v_1(t)\mathbf{B}_1 + v_2(t)\mathbf{B}_2 + \dots + v_{r-1}(t)\mathbf{B}_{r-1} + v_r(t)\mathbf{B}_r = \sum_{i=1}^r \mathbf{B}_i(t)v_i(t). \quad (3.7)$$

Następnie różniczkuje się wybraną reprezentację (3.5) lub (3.6), wykorzystuje własność grupy $\mathbf{S}(t)\mathbf{S}^{-1}(t) = \mathbf{I}$ i rozwija elementy $e^{h_i(t)\mathbf{B}_i}$ w skończony (ze względu na założoną

nilpotentność układu) szereg Taylora (2.27), aby wyliczyć lewą stronę równania (3.7) w postaci kombinacji liniowej elementów bazowych \mathbf{B}_i . Porównując współczynniki przy odpowiednich elementach bazowych \mathbf{B}_i lewej i prawej strony równania (3.7) powstaje zależność zwana równaniem Chena-Fliessa-Sussmanna (CFS)

$$\dot{\mathbf{h}}(t) = \mathbf{M}(\mathbf{h}(t))\mathbf{v}(t), \quad \mathbf{h}(0) = \mathbf{0}, \quad (3.8)$$

gdzie $\mathbf{M}(\mathbf{h})$ jest pewną macierzą rozmiaru $(r \times r)$. Całkowanie numeryczne zależności (3.8) umożliwia wyznaczenie $\mathbf{h}(T)$.

Przykład 13. Równanie (3.7) dla układu o dwóch generatorach, $m = 2$, stopnia nilpotentności 2, którego przykładem jest integrator Brocketta (A.1), jest następujące ($r = 3$)

$$\mathbf{S}^{-1}(t)\dot{\mathbf{S}}(t) = v_1(t)\mathbf{B}_1 + v_2(t)\mathbf{B}_2 + v_3(t)\mathbf{B}_3. \quad (3.9)$$

Lewą stronę równania (3.9) wyliczamy dla reprezentacja „wstecz” w bazie Halla ($\mathbf{B}_3 = [\mathbf{B}_1, \mathbf{B}_2]$)

$$\mathbf{S}(t) = e^{h_3(t)\mathbf{B}_3} e^{h_2(t)\mathbf{B}_2} e^{h_1(t)\mathbf{B}_1}. \quad (3.10)$$

Zróżniczkowanie zależności (3.10) prowadzi do

$$\begin{aligned} \dot{\mathbf{S}}(t) &= \dot{h}_3(t)\mathbf{B}_3 e^{h_3(t)\mathbf{B}_3} e^{h_2(t)\mathbf{B}_2} e^{h_1(t)\mathbf{B}_1} \\ &+ e^{h_3(t)\mathbf{B}_3} \dot{h}_2(t)\mathbf{B}_2 e^{h_2(t)\mathbf{B}_2} e^{h_1(t)\mathbf{B}_1} + e^{h_3(t)\mathbf{B}_3} e^{h_2(t)\mathbf{B}_2} \dot{h}_1(t)\mathbf{B}_1 e^{h_1(t)\mathbf{B}_1}, \end{aligned} \quad (3.11)$$

natomiast odwrotność

$$\mathbf{S}^{-1}(t) = e^{-h_1(t)\mathbf{B}_1} e^{-h_2(t)\mathbf{B}_2} e^{-h_3(t)\mathbf{B}_3}. \quad (3.12)$$

Podstawiając (3.11), (3.12) do (3.9) i, dla uproszczenia zapisu, pomijając zależności od czasu otrzymujemy

$$\begin{aligned} \mathbf{S}^{-1}\dot{\mathbf{S}} &= v_1\mathbf{B}_1 + v_2\mathbf{B}_2 + v_3\mathbf{B}_3 = e^{-h_1\mathbf{B}_1} e^{-h_2\mathbf{B}_2} e^{-h_3\mathbf{B}_3} \dot{h}_3\mathbf{B}_3 e^{h_3\mathbf{B}_3} e^{h_2\mathbf{B}_2} e^{h_1\mathbf{B}_1} + \\ &+ e^{-h_1\mathbf{B}_1} e^{-h_2\mathbf{B}_2} \dot{h}_2\mathbf{B}_2 e^{h_2\mathbf{B}_2} e^{h_1\mathbf{B}_1} + e^{-h_1\mathbf{B}_1} \dot{h}_1\mathbf{B}_1 e^{h_1\mathbf{B}_1}. \end{aligned} \quad (3.13)$$

Z nilpotentności układu (A.1), wszystkie pola wektorowe stopnia większego niż drugi zerują się tożsamościowo. Zatem rozwinięcia eksponent w szereg Taylora są następujące

$$e^{\pm h_i\mathbf{B}_i} = \mathbf{I} \pm h_i\mathbf{B}_i + \frac{1}{2}h_i^2\mathbf{B}_i^2 \quad \text{dla } i \in \{1, 2\}, \quad e^{\pm h_3\mathbf{B}_3} = \mathbf{I} \pm h_3\mathbf{B}_3. \quad (3.14)$$

\mathbf{B}_3 jest stopnia 2, zatem \mathbf{B}_3^2 jest stopnia 4, więc zeruje się. Następnie podstawiając (3.14) do (3.13), pamiętając o nieprzemiennej mnożeniu, oraz wykorzystując definicję nawiasu Liego (2.23) otrzymujemy

$$\dot{h}_1\mathbf{B}_1 + \dot{h}_2(\mathbf{B}_2 - h_1\mathbf{B}_3) + \dot{h}_3\mathbf{B}_3 = v_1\mathbf{B}_1 + v_2\mathbf{B}_2 + v_3\mathbf{B}_3, \quad (3.15)$$

a po porównaniu odpowiednich współczynników przy \mathbf{B}_i , równanie CFS przyjmuje postać

$$\begin{cases} \dot{h}_1(t) = v_1(t) \\ \dot{h}_2(t) = v_2(t) \\ \dot{h}_3(t) = v_3(t) + h_1(t)v_2(t), \end{cases} \quad (3.16)$$

z warunkiem początkowym $\mathbf{h}(0) = \mathbf{0}$.

3.3. Generacja sterowań

Ostatnim krokiem algorytmu LS jest wygenerowanie sterowań na podstawie współczynników $\mathbf{h}(T)$ opisujących sekwencje ruchów wynikającą z wybranej reprezentacji. W oryginalnej wersji algorytmu LS do generacji sterowań autorzy wykorzystywali formułę CBHD (2.32), o kawałkami stałych sygnałach sterujących. W pracach [17, 48] zaproponowano wykorzystanie uogólnionej formuły CBHD (2.35) dopuszczającą sterowania kawałkami ciągłe.

3.3.1. Sterowania kawałkami stałe (CBHD)

Ideę wykorzystania formuły CBHD podczas generacji sterowań zilustrujemy przykładem.

Przykład 14. Niech \mathbf{X} , \mathbf{Y} oznaczają formalne generatory – odpowiedniki generatorów pewnego układu (2.36), a celem będzie wygenerowanie ruchu w kierunku $[\mathbf{X}, \mathbf{Y}]$. Łącząc cztery operatory przemieszczenia oraz wykorzystując (2.31) i (2.32) można układ przemieścić w następujący sposób [17, 55]:

$$\begin{aligned} \exp(h(t)\mathbf{X})\exp(h(t)\mathbf{Y})\exp(-h(t)\mathbf{X})\exp(-h(t)\mathbf{Y}) &= \\ &= \exp\left(h(t)\mathbf{Y} + h^2(t)[\mathbf{X}, \mathbf{Y}] + \frac{h^3(t)}{2}[\mathbf{X}, [\mathbf{X}, \mathbf{Y}]] + o(t^3)\right)\exp(-h(t)\mathbf{Y}) \\ &= \exp(h(t)\mathbf{Y} + h^2(t)[\mathbf{X}, \mathbf{Y}] + \frac{h^3(t)}{2}[\mathbf{X}, [\mathbf{X}, \mathbf{Y}]] - h(t)\mathbf{Y} + \\ &\quad + \frac{h^2(t)}{2}[\mathbf{Y} + h(t)[\mathbf{X}, \mathbf{Y}] + \frac{h^2(t)}{2}[\mathbf{X}, [\mathbf{X}, \mathbf{Y}]], -\mathbf{Y}] + o(t^3)) \\ &= \exp(h^2(t)[\mathbf{X}, \mathbf{Y}] + \frac{h^3(t)}{2}[\mathbf{X}, [\mathbf{X}, \mathbf{Y}]] + \frac{h^3(t)}{2}[\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]] + o(t^3)). \end{aligned}$$

gdzie $o(t^3)$ oznacza małą 3 rzędu, czyli $o(t^3)/t^3 \xrightarrow{t \rightarrow 0} 0$. Dla układu nilpotentnego rzędu 2 formuła powyższa przyjmuje postać

$$\exp(h(t)\mathbf{X})\exp(h(t)\mathbf{Y})\exp(-h(t)\mathbf{X})\exp(-h(t)\mathbf{Y}) = \exp(h^2(t)[\mathbf{X}, \mathbf{Y}]). \quad (3.17)$$

Jest to jeden ze sposobów generacji ruchu w kierunku $[\mathbf{X}, \mathbf{Y}]$, inne sposoby przedstawiono w pracy [44].

Ogólna metoda generowania sterowań z wykorzystaniem formuły CBHD jest następująca: niech $\mathbf{B}_1, \dots, \mathbf{B}_m$ oznaczają formalne generatory. Zadana reprezentację (3.5),

bądź (3.6) dzięki rekurencyjnemu zastosowaniu (2.32), przedstawia się za pomocą skończonego iloczynu bazowych operatorów przemieszczenia

$$\mathbf{S}(T) = \exp(a_1 \tilde{\mathbf{B}}_1) \exp(a_2 \tilde{\mathbf{B}}_2) \dots \exp(a_k \tilde{\mathbf{B}}_k), \quad (3.18)$$

gdzie $\tilde{\mathbf{B}}_i$, $i = 1, \dots, k$ są sekwencją złożoną z formalnych generatorów $\mathbf{B}_1, \dots, \mathbf{B}_m$ występujące w pewnej liczbie oraz kolejności. Współczynniki a_i opisujące czasy działania kolejnych operatorów $\exp(a_i \tilde{\mathbf{B}}_i)$, są znane na podstawie współczynników $\mathbf{h}(T)$ wyliczonych z zależności (3.8). Każdy z formalnych generatorów odpowiada polu wektorowemu układu (2.36), a na każde z nich można wpływać bezpośrednio za pomocą sterowania

$$u_i(t) = \begin{cases} \operatorname{sgn}(a_i) & \text{jeśli } \tilde{\mathbf{B}}_i = \mathbf{B}_i \\ 0 & \text{jeśli } \tilde{\mathbf{B}}_i \neq \mathbf{B}_i \end{cases} \quad \text{dla } t \in \left[\sum_{k=1}^{i-1} |a_k|, \sum_{k=1}^i |a_k| \right]. \quad (3.19)$$

Przykład 15. Rozważmy układ o dwóch generatorach formalnych $\mathbf{B}_1, \mathbf{B}_2$, $m = 2$, stopniu nilpotentności 2, z reprezentacją ruchu „wstecz” zadaną przez (3.10), $r = 3$. Celem jest wygenerowanie wektora zadanych współczynników Halla $\mathbf{h}(T) = (-1, 2, 3)^\top$ dla $T = 1$. Z zależności (3.17) zastosowanej dla reprezentacja (3.10), uwzględnieniem zadanych współczynników Halla scenariusz ruchu jest postaci

$$e^{\sqrt{3}\mathbf{B}_1} e^{\sqrt{3}\mathbf{B}_2} e^{-\sqrt{3}\mathbf{B}_1} e^{-\sqrt{3}\mathbf{B}_2} e^{2\mathbf{B}_2} e^{-1\mathbf{B}_1} = e^{\sqrt{3}\mathbf{B}_1} e^{\sqrt{3}\mathbf{B}_2} e^{-\sqrt{3}\mathbf{B}_1} e^{2-\sqrt{3}\mathbf{B}_2} e^{-1\mathbf{B}_1}. \quad (3.20)$$

Seqwencja $e^{-\sqrt{3}\mathbf{B}_2} e^{2\mathbf{B}_2}$ została skompresowana, gdyż opisuje ruch wzdłuż tego samego pola. Wykorzystując wzór (3.19) sekwencja sterowań jest następująca

$$u_1(t) = \begin{cases} 1 \\ 0 \\ -1 \\ 0 \\ -1 \end{cases} \quad u_2(t) = \begin{cases} 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{cases} \quad \text{dla } \begin{cases} t \in [0, \sqrt{3}] \\ t \in (\sqrt{3}, 2\sqrt{3}] \\ t \in (2\sqrt{3}, 3\sqrt{3}] \\ t \in (3\sqrt{3}, 2 + 2\sqrt{3}] \\ t \in (2 + 2\sqrt{3}, 3 + 2\sqrt{3}]. \end{cases} \quad (3.21)$$

Liczba elementarnych ruchów dla sterowań kawałkami stałych nie musi być jednakowa dla ustalonego rzędu nilpotentności i liczby generatorów, zależy bowiem od wybranej reprezentacji ruchu oraz sposobu generacji nawiasów Liego za pomocą ruchów elementów. W przykładzie 15, pomimo ustalenia czasu T na wartość 1, rzeczywisty czas ruchu wynosił $3 + 2\sqrt{3}$. Należy pamiętać więc, że na przedziale czasu $[0, T]$ opisana jest ewolucja równań (3.8), a nie prawdziwe przemieszczenie układu sterowanego. Rzeczywisty czas ruchu można kształtować przez odpowiednie przeskalowanie sterowań (3.19) (np. zmniejszając czas ruchu kosztem zwiększenia amplitudy (energii) sterowań).

3.3.2. Sterowania kawałkami ciągle (gCBHD)

Drugi ze sposobów generacji sterowań wykorzystuje uogólnioną formułę gCBHD. Nieautonomiczny układ równań różniczkowych (2.33) utożsamia się z bezdryfowym układem sterowania (2.36), co prowadzi do zależności

$$\begin{aligned} \mathbf{z}(t) = & \sum_{i=1}^m \int_{\mathbb{T}_r(t)} \mathbf{g}_i u_i(s_{\sigma(1)}) ds + \sum_{\sigma \in \mathbb{P}_2} \frac{(-1)^{\text{err}(\sigma)}}{2^2 \binom{2-1}{\text{err}(\sigma)}} \sum_{i,j=1}^m \int_{\mathbb{T}_2(t)} [\mathbf{g}_i, \mathbf{g}_j] u_i(s_{\sigma(1)}) u_j(s_{\sigma(2)}) ds + \\ & + \sum_{\sigma \in \mathbb{P}_3} \frac{(-1)^{\text{err}(\sigma)}}{3^2 \binom{3-1}{\text{err}(\sigma)}} \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^m \int_{\mathbb{T}_3(t)} [\mathbf{g}_i, [\mathbf{g}_j, \mathbf{g}_k]] u_i(s_{\sigma(1)}) u_j(s_{\sigma(2)}) u_k(s_{\sigma(3)}) ds + \dots \end{aligned} \quad (3.22)$$

Szereg (3.22) jest skończony, jeśli układ (2.36) jest nilpotentny. Ponieważ testowanie algorytmów planowania ruchu będzie przeprowadzane dla układów dwuwęściowych, $m = 2$, równanie (3.22) zapiszemy dla tego właśnie przypadku, z wykorzystaniem upraszczającej zapis konwencji, w myśl której $u_i(s_1)u_j(s_2) \dots u_k(s_n) = u_{ij\dots k}$

$$\begin{aligned} \mathbf{z}(t) = & \int_{\mathbb{T}_1(t)} \mathbf{g}_1 u_1 ds + \int_{\mathbb{T}_1(t)} \mathbf{g}_2 u_2 ds + \\ & + \int_{\mathbb{T}_2(t)} [\mathbf{g}_1, \mathbf{g}_2] \left(\frac{1}{4} u_{12} - \frac{1}{4} u_{21} \right) ds + \int_{\mathbb{T}_2(t)} [\mathbf{g}_2, \mathbf{g}_1] \left(-\frac{1}{4} u_{12} + \frac{1}{4} u_{21} \right) ds + \\ & + \int_{\mathbb{T}_3(t)} [\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]] \left(\frac{1}{9} u_{112} - \frac{1}{18} u_{121} - \frac{1}{18} u_{112} - \frac{1}{18} u_{211} - \frac{1}{18} u_{121} + \frac{1}{9} u_{211} \right) ds + \\ & + \int_{\mathbb{T}_3(t)} [\mathbf{g}_1, [\mathbf{g}_2, \mathbf{g}_1]] \left(-\frac{1}{9} u_{121} + \frac{1}{18} u_{112} + \frac{1}{18} u_{211} + \frac{1}{18} u_{112} + \frac{1}{18} u_{211} - \frac{1}{9} u_{121} \right) ds + \\ & + \int_{\mathbb{T}_3(t)} [\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]] \left(\frac{1}{9} u_{212} - \frac{1}{18} u_{221} - \frac{1}{18} u_{122} - \frac{1}{18} u_{221} - \frac{1}{18} u_{122} + \frac{1}{9} u_{212} \right) ds + \\ & + \int_{\mathbb{T}_3(t)} [\mathbf{g}_2, [\mathbf{g}_2, \mathbf{g}_1]] \left(-\frac{1}{9} u_{221} + \frac{1}{18} u_{212} + \frac{1}{18} u_{221} + \frac{1}{18} u_{122} + \frac{1}{18} u_{212} - \frac{1}{9} u_{122} \right) ds + \dots \end{aligned} \quad (3.23)$$

Dalsze uproszczenie (3.23) polega na zapisie $\mathbf{z}(t)$ jako kombinacji elementów bazowych (dalej stosowana jest baza Halla). Korzystając z antysymetrii mamy $[\mathbf{g}_2, \mathbf{g}_1] = -[\mathbf{g}_1, \mathbf{g}_2]$, $[\mathbf{g}_1, [\mathbf{g}_2, \mathbf{g}_1]] = -[\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]]$, $[\mathbf{g}_2, [\mathbf{g}_2, \mathbf{g}_1]] = -[\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]]$. Ostatecznie, początkowe elementy szeregu opisującego formułę gCBHD zależą od sterowań następująco

$$\begin{aligned} \mathbf{z}(t) = & \mathbf{g}_1 \int_{\mathbb{T}_1(t)} u_1 ds + \mathbf{g}_2 \int_{\mathbb{T}_1(t)} u_2 ds + \frac{1}{2!} [\mathbf{g}_1, \mathbf{g}_2] \int_{\mathbb{T}_2(t)} (u_{12} - u_{21}) ds + \\ & + \frac{1}{3!} [\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]] \int_{\mathbb{T}_3(t)} (u_{112} - 2u_{121} + u_{211}) ds + \\ & + \frac{1}{3!} [\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]] \int_{\mathbb{T}_3(t)} (-u_{112} + 2u_{121} - u_{211}) ds + \dots \end{aligned} \quad (3.24)$$

Tablica 3.1. Pre-sterowania dla elementów bazy Halla i Chibrikova do warstwy piątej włącznie generowane przez \mathbf{X}, \mathbf{Y} (η – współczynnik przez który należy przemnożyć odpowiednie pre-sterowanie)

η	element bazy Halla	pre-sterowanie	element bazy Chibrikova
1	\mathbf{X}	u_1	\mathbf{X}
1	\mathbf{Y}	u_2	\mathbf{Y}
$\frac{1}{2}$	$[\mathbf{X}, \mathbf{Y}]$	$u_{12} - u_{21}$	$[\mathbf{X}, \mathbf{Y}]$
$\frac{1}{3!}$	$[\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]$	$u_{112} - 2u_{121} + u_{211}$	$[\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]$
$\frac{1}{3!}$	$[\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]$	$-u_{122} + 2u_{212} - u_{221}$	$[\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]$
$\frac{1}{4!}$	$[\mathbf{X}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]$	$-4u_{1121} + 4u_{1211}$	$[\mathbf{X}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]$
$\frac{1}{4!}$	$[\mathbf{Y}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]$	$-4u_{1122} + 4u_{1212} - 4u_{2121} + 4u_{2211}$	$[\mathbf{X}, [\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]]$
$\frac{1}{4!}$	$[\mathbf{Y}, [\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]]$	$-4u_{2212} + 4u_{2211}$	$[\mathbf{Y}, [\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]]$
$\frac{1}{5!}$	$[\mathbf{X}, [\mathbf{X}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]]$	$-4u_{11112} - 4u_{11121} + 16u_{11211} - 4u_{12111} - 4u_{21111}$	$[\mathbf{X}, [\mathbf{X}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]]$
$\frac{1}{5!}$	$[\mathbf{Y}, [\mathbf{X}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]]$	$-8u_{11122} + 12u_{11212} + 12u_{11221} - 8u_{12112} - 8u_{12121} + 12u_{12211} - 8u_{21112} - 8u_{21121} + 12u_{21211} - 8u_{22111}$	
$\frac{1}{5!}$	$[\mathbf{Y}, [\mathbf{Y}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]]$	$8u_{11222} - 12u_{12122} + 8u_{12212} + 8u_{12221} - 12u_{21122} + 8u_{21212} + 8u_{21221} - 12u_{22112} - 12u_{22121} + 8u_{22211}$	
$\frac{1}{5!}$	$[\mathbf{Y}, [\mathbf{Y}, [\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]]]$	$4u_{12222} + 4u_{21222} - 16u_{22122} + 4u_{22212} + 4u_{22221}$	$[\mathbf{Y}, [\mathbf{Y}, [\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]]]$
$\frac{1}{5!}$	$[[\mathbf{X}, \mathbf{Y}], [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]$	$-12u_{11122} + 8u_{11212} + 8u_{11221} + 8u_{12112} - 12u_{12121} + 8u_{12211} - 12u_{21112} + 8u_{21121} + 8u_{21211} - 12u_{22111}$	$[\mathbf{X}, [\mathbf{X}, [\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]]]$
$\frac{1}{5!}$	$[[\mathbf{X}, \mathbf{Y}], [\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]]$	$-4u_{11222} - 4u_{12122} + 16u_{12212} - 4u_{12221} - 4u_{21122} - 4u_{21212} + 16u_{21221} - 4u_{22112} - 4u_{22121} - 4u_{22211}$	$[\mathbf{X}, [\mathbf{Y}, [\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]]]$
$\frac{1}{5!}$		$4u_{11122} + 4u_{11212} + 4u_{11221} - 16u_{12112} + 4u_{12121} + 4u_{12211} + 4u_{21112} - 16u_{21121} + 4u_{21211} + 4u_{22111}$	$[\mathbf{Y}, [\mathbf{X}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]]$
$\frac{1}{5!}$		$12u_{11222} - 8u_{12122} - 8u_{12212} + 12u_{12221} - 8u_{21122} + 12u_{21212} - 8u_{21221} - 8u_{22112} - 8u_{22121} + 12u_{22211}$	$[\mathbf{Y}, [\mathbf{X}, [\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]]]$

Sumy podcałkowe występujące w (3.24) zwykle nazywa się pre-sterowaniami (gdyż nie specyfikują one w jakiej klasie sterowań będą poszukiwane sterowania rzeczywiste). Dla baz Halla i Chibrikova pre-sterowania do warstwy 5-ej włącznie zebrano w tab. 3.1.

Ogólnie przedstawienie dowolnego elementu algebry Liego, za pomocą bazowych jednomianów Liego jest zadaniem trudnym (szczególnie dla wyższych warstw). Istnieje jednak algorytm [17] przedstawiający dowolny jednomian Liego jako kombinację elementów bazy Ph. Halla.

W celu wyliczenia sterowań rzeczywistych z wykorzystaniem formuły gCBHD najpierw dobiera się ich klasę. Zwykle są to sterowania w postaci funkcji harmonicznym pochodzących z bazy Fouriera lub funkcje wielomianowe [25]. Ustalona (i skończona) reprezentacja sterowań $u_i(t, \mathbf{p})$, $i \in 1, \dots, m$, działających na przedziale czasu $t \in [0, T]$ i zależnych od wektora parametrów \mathbf{p} , jest podstawiana do wzoru (3.24), w wyniku czego otrzymuje się pewne funkcje zależne od parametrów

$$\mathbf{z}(T) = \mathbf{g}_1 f_{\mathbf{g}_1}(\mathbf{p}, T) + \mathbf{g}_2 f_{\mathbf{g}_2}(\mathbf{p}, T) + \mathbf{g}_3 f_{\mathbf{g}_3}(\mathbf{p}, T) + \mathbf{g}_4 f_{\mathbf{g}_4}(\mathbf{p}, T) + \dots, \quad (3.25)$$

gdzie $\mathbf{z}(T)$ oznacza pożądaną wartość zakreślenia, $f_{\mathbf{g}_i}(\mathbf{p}, T)$ oznaczają wartości całek po sympleksach z pre-sterowań w równaniu (3.24) odpowiadających polom wektorowym \mathbf{g}_i wartościowanymi w konfiguracji \mathbf{q} , a \mathbf{g}_i są kolejnymi elementami bazy (Halla) pojawiającymi się w zależności (3.24) ($\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3 = [\mathbf{g}_1, \mathbf{g}_2]$ itd.). Wektor funkcji zależny od parametrów sterowań

$$\boldsymbol{\beta} = (f_{\mathbf{g}_1}(\mathbf{p}, T), f_{\mathbf{g}_2}(\mathbf{p}, T), f_{\mathbf{g}_3}(\mathbf{p}, T), f_{\mathbf{g}_4}(\mathbf{p}, T), \dots)^T = \mathbf{f}_{\mathbf{g}}(\mathbf{p}) \quad (3.26)$$

można traktować jako odpowiednik kinematyki prostej manipulatorów, a zadaniem najczęściej rozwiązywanym jest odwrotne zadanie kinematyki (na podstawie znanego przemieszczenia $\mathbf{z}(T)$ i wartości wektorów $\mathbf{g}_i(\mathbf{q})$, $i = 1, \dots$ wyznacza się pożądaną wartość $\boldsymbol{\beta}_d$, która z kolei służy do wyznaczenia \mathbf{p}).

Ponieważ \mathbf{B}_i są formalnymi odpowiednikami \mathbf{g}_i dla wybranej reprezentacji ruchu („w przód” (3.5), „wstecz” (3.6)), zatem dla generacji składowych elementów $e^{h_i(T)\mathbf{B}_i}$ ze znanymi $\mathbf{h}(T)$ można skorzystać z zależności

$$e^{h_i(T)\mathbf{B}_i} = e^{\mathbf{z}(T)}$$

i kolejno porównywać operatory przemieszczenia pojawiające się w reprezentacji z zależnością (3.25), co prowadzi do sekwencji układów równań

$$\begin{cases} f_{\mathbf{g}_1}(\mathbf{p}, T) & = 0 \\ \vdots & \vdots \\ f_{\mathbf{g}_{i-1}}(\mathbf{p}, T) & = 0 \\ f_{\mathbf{g}_i}(\mathbf{p}, T) & = h_i(T) \\ f_{\mathbf{g}_{i+1}}(\mathbf{p}, T) & = 0 \\ \vdots & \vdots \\ f_{\mathbf{g}_r}(\mathbf{p}, T) & = 0. \end{cases} \quad (3.27)$$

Równanie (3.27) po nadaniu postaci wektorowej $\mathbf{f}_{\mathbf{g}}(\mathbf{p}) = \boldsymbol{\beta}_{h_i} = (0, \dots, 0, h_i(T), 0, \dots, 0)^T$, należy rozwiązać ze względu na \mathbf{p} , uzyskując wartości nieznanymi parametrów sterowań.

Przykład 16. Dla danych jak w przykładzie 15 należy wygenerować ruch korzystając z bazy wielomianowej. Niech bazą funkcyjną dla sterowań będą wielomiany Czebyszewa (cztery pierwsze z nich to $1, t, 2t^2 - 1, 4t^3 - 3t$, [9]). Arbitralnie wybieramy reprezentację sterowań w klasie wielomianów Czebyszewa jako $u_1(t) = p_1 + p_2t, u_2(t) = p_3 + p_4(2t^2 - 1)$. Wyznaczamy całki po sympleksach z kolejnych pre-sterowań (3.24)

$$\begin{cases} f_{g_1}(\mathbf{p}, T) = \int_{s_1=0}^T u_1(s_1) ds_1 \\ f_{g_2}(\mathbf{p}, T) = \int_{s_1=0}^T u_2(s_1) ds_1 \\ f_{g_3}(\mathbf{p}, T) = \frac{1}{2!} \int_{s_2=0}^T \int_{s_1=0}^{s_2} (u_1(s_1)u_2(s_2) - u_2(s_1)u_1(s_2)) ds_1 ds_2, \end{cases}$$

które wynoszą

$$\begin{cases} f_{g_1}(\mathbf{p}, T) = p_1 T + \frac{p_2 T^2}{2} \\ f_{g_2}(\mathbf{p}, T) = p_3 T - p_4 T + \frac{2p_4 T^3}{3} \\ f_{g_3}(\mathbf{p}, T) = -\frac{1}{12} p_2 p_3 T^3 + \frac{1}{12} p_2 p_4 T^3 + \frac{1}{6} p_1 p_4 T^4 + \frac{1}{30} p_2 p_4 T^5. \end{cases}$$

Reprezentacja (3.10) składa się z trzech segmentów, zatem należy rozwiązać trzy układy równań, z uwzględnieniem $T = 1$, które są postaci

$$\begin{cases} p_1 + \frac{p_2}{2} = \beta_{h_i 1} \\ p_3 - \frac{p_4}{3} = \beta_{h_i 2} \\ -\frac{1}{12} p_2 p_3 + \frac{1}{6} p_1 p_4 + \frac{7}{60} p_2 p_4 = \beta_{h_i 3}, \end{cases} \quad \text{dla } i \in 3, 2, 1,$$

gdzie $\beta_{h_3} = (0, 0, 3)^T, \beta_{h_2} = (0, 2, 0)^T, \beta_{h_1} = (-1, 0, 0)^T$. Wszystkie układy równań są redundantne, a przykładowymi ich rozwiązaniami są następujące wektory parametrów $\mathbf{p}_{h_3} = (-10, 20, 9, 27)^T, \mathbf{p}_{h_2} = (0, 0, 2, 0)^T, \mathbf{p}_{h_1} = (-1, 0, 0, 0)^T$, stąd sterowania są postaci

$$u_1(t) = \begin{cases} -10 + 20t & t \in [0, 1], \\ 0 & t \in [1, 2], \\ -1 & t \in [2, 3]. \end{cases} \quad u_2(t) = \begin{cases} 9 + 27(2t^2 - 1) & t \in [0, 1], \\ 2 & t \in [1, 2], \\ 0 & t \in [2, 3]. \end{cases}$$

Zależnie od doboru klasy sterowań i liczby parametrów układ równań (3.27) może być (i najczęściej w praktyce jest) redundantny. W takim przypadku wykorzystanie formuły gCBHD do wyliczenia sterowań pozwala na optymalizację dodatkowych kryteriów jakości. Zły dobór reprezentacji dla wybranej klasy sterowań (np. zbyt mało składowych) może skutkować nieistnieniem rozwiązania układu (3.27). W przypadku doboru klasy sterowań warto zadbać, aby funkcje były niezależne od siebie, co jest szczególnie istotne podczas generacji kierunków (nawiasów Liego) wyższych warstw. Taką niezależność zapewniają funkcje ortogonalne określone na skończonym i ustalonym przedziale $[0, T]$. Przykładowo, funkcje bazowe Fouriera, Legendre'a, czy Czebyszewa.

W ogólnym przypadku do rozwiązania układu (3.27) wykorzystuje się algorytm Newtona, korzystając z analogii równania (3.27) do kinematyki prostej manipulatorów

(β jest odpowiednikiem punktu w przestrzeni zadaniowej, a \mathbf{p} – konfiguracji). Dla szczególnych przypadków wektora β można wyliczyć analitycznie sterowania optymalne (por. Rozdział 7).

3.4. Podsumowanie

Algorytm LS może być zapisany za pomocą następującej sekwencji kroków.

Algorytm 1. Podstawowy LS

Krok 1 Na podstawie generatorów układu (2.36) wyliczyć elementy bazy algebry Liego i utworzyć układ rozszerzony (3.1). Dobrać dowolną trajektorię referencyjną $\lambda(t)$, $t \in [0, T]$ łączącą zadaną konfigurację początkową $\mathbf{q}_0 = \lambda(0)$ z konfiguracją końcową $\mathbf{q}_f = \lambda(T)$ i wyliczyć sterowania rozszerzone na podstawie (3.3).

Krok 2 Wybrać jedną z reprezentacji (3.5) lub (3.6). Utworzyć odpowiednik układu rozszerzonego w formalnej grupie Liego dany wzorem (3.4). Wyliczyć równania CFS (3.8), następnie rozwiązać je uzyskując wektor współczynników $\mathbf{h}(T)$ dla wybranej reprezentacji ruchu.

Krok 3 Wykorzystując formułę CBHD, w jednej z postaci (2.31) lub (2.32), transformować wybraną reprezentację do postaci przemieszczeń generatorów (3.18) i dla tej postaci wygenerować sterowania rzeczywiste korzystając ze wzoru (3.19).

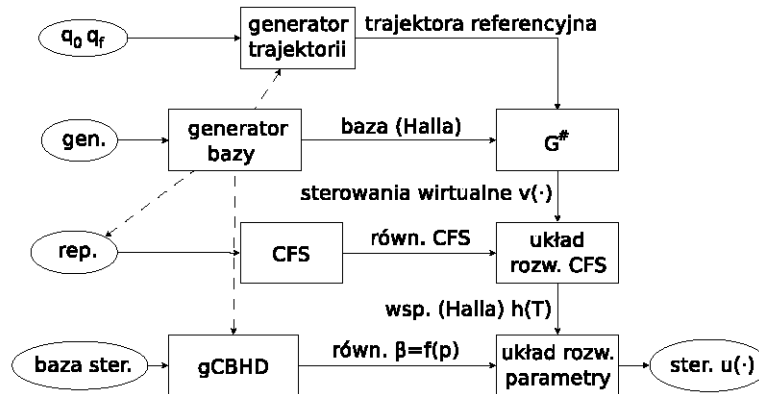
Danymi wejściowymi dla podstawowego algorytmu LS, oprócz układu sterowanego (2.36) i konfiguracji brzegowych $\mathbf{q}_0, \mathbf{q}_f$ pomiędzy którymi planowany jest ruch, jest reprezentacja ruchu ((3.5), (3.6)). Modyfikację algorytmu LS z wykorzystaniem uogólnionej formuły gCBHD opisuje Algorytm 2, o wspólnych dwóch pierwszych krokach z Algorytmem 1:

Algorytm 2. LS z wykorzystaniem gCBHD

Krok 3 Wybrać klasę funkcji sterujących i skończoną reprezentację sterowań zależną od wektora parametrów \mathbf{p} . Wyliczyć pre-sterowania zgodnie z zależnością (3.24) i całkując uzyskać funkcje zależne od \mathbf{p} (3.25). Porównać po kolei strumienie wybranej reprezentacji ruchu z zależnością (3.25) tworząc układy równań (3.27) z podlegającymi dookreśleniu wektorom parametrów \mathbf{p} . Rozwiązać układy (3.27), a na podstawie wyliczonych parametrów \mathbf{p} wyznaczyć rzeczywiste sterowania.

Rysunek 3.1 przedstawia schemat blokowy algorytmu LS zgodny z Algorytmem 2. Istnieje możliwość konfiguracji metody LS przez dobór:

- trajektorii referencyjnej – badaniu wpływu krzywej łączącej konfigurację początkową z docelową poświęcono rozdział 6. Trajektorja referencyjna jest niezbędna do wyliczenia sterowań rozszerzonych (por. Rysunek 3.1). Autorzy algorytmu LS sugerowali, że można za jej pomocą wpływać na generowane sterowania, jednak dla uproszczenia zawsze wybierali odcinek łączący konfiguracje brzegowe;



Rysunek 3.1. Rozszerzony algorytm LS

- reprezentacji ruchu – w metodzie oryginalnej dostępne są jedynie reprezentacje znane jako „w przód” i „wstecz”. Warto rozważyć inne permutacje strumieni elementów bazowych, a także bardziej skomplikowane reprezentacje ruchu. Proponowane dotychczas reprezentacje ruchu można uważać za przejaw współrzędnych kanonicznych drugiego rodzaju (2.21). Można również wykorzystać reprezentacje ruchu będące odpowiednikiem współrzędnych pierwszego rodzaju (2.19). Rozważaniom dotyczącym reprezentacji ruchu poświęcono rozdział 4. Na rysunku 3.1 wejście opisujące reprezentacje ruchu oznaczono jako *rep.*,
- klasy sterowań – realizujących zadanie planowania z wykorzystaniem formuły gCBHD. Oryginalna metoda LS oferuje jedynie sterowania kawałkami stałe, co powoduje pewne ograniczenia w realizowalności fizycznej sterowań;
- bazy algebry Liego – w oryginalnym algorytmie domyślną jest baza Halla. Warto zauważyć jednak, że w ustalonej bazie wyrażana jest reprezentacja ruchu. To baza determinuje postać układu rozszerzonego, a także wpływa na finalną postać formuły CBHD oraz gCBHD (zob. przykładowo (3.24)).

W pracy [53] Lafferriere podał własności algorytmu LS. Nie generuje rozwiązania optymalnego – dobiera się dowolną postać klasy sterowań, natomiast algorytm LS pozwala wyliczyć parametry przyjętej reprezentacji sterowań dla konkretnego zadania. Zatem to programista dobiera klasę sterowań i ich reprezentację. Wynikiem algorytmu LS jest dokładne rozwiązanie zadania planowania ruchu dla układów nilpotentnych oraz nilpotentyzowalnych przez sprzężenie zwrotne. Dopuszcza się modyfikację w postaci iteracyjnego algorytmu LS działającego dla ogólnych układów bezdryfowych, który przeprowadza układ od konfiguracji początkowej \mathbf{q}_0 do docelowej \mathbf{q}_f z dowolnie zadaną dokładnością.

4. Reprezentacje ruchu

Rozpatrzmy jeszcze raz rozszerzone równanie formalne w postaci (3.7) i rozszerzmy literaturowe reprezentacje „wstecz” i „w przód” o inne (autorskie) dopuszczalne reprezentacje. Aby $\mathbf{S}(t)$ było reprezentacją ruchu, musi zapewniać ruch we wszystkich kierunkach bazowych \mathbf{B}_i , więc liczba zmiennych opisujących ruch musi być co najmniej równa r . Dodatkowo odwzorowanie $\mathbf{h} = (h_1, \dots, h_k) \rightarrow \mathbb{R}^r$ musi być surjekcją ($k \geq r$). Aby zminimalizować złożoność obliczeniową zakładamy dodatkowo, że jest bijekcją ($k = r$). W tym przypadku elementy \mathbf{h} mogą być nazwane współrzędnymi instancji operatora $\mathbf{S}(t)$. Rozważmy zatem ogólną rodzinę reprezentacji daną wzorem

$$\mathbf{S}(t) = \prod_{i=1}^{\bar{r}} e^{\langle \boldsymbol{\rho}_i, \mathbf{h}(t), \mathbf{B} \rangle}, \quad (4.1)$$

gdzie $\mathbf{h}(t) = (h_1(t), \dots, h_r(t))^T$ jest wektorem współrzędnych Halla, Lyndona, czy Chibrikova – zależnie od wyboru bazy, $1 \leq \bar{r} \leq r$ opisuje z ilu akcji elementarnych składa się ruch, $\mathbf{B} = (\mathbf{B}_1, \dots, \mathbf{B}_r)^T$ gromadzi wszystkie elementy wybranej bazy o stopniu mniejszym lub równym stopniowi nilpotentności układu, $\boldsymbol{\rho}_i = (\rho_1, \dots, \rho_r)^T$ oznacza i -ty wektor wyboru, a binarne zmienne decyzyjne $\rho_j \in \{0, 1\}$ włączają, 1, lub wyłączają, 0, poszczególne elementy \mathbf{B} . Odwzorowanie $\mathbb{R}^n \star \mathbb{R}^n \rightarrow \mathbb{R}^n$ jest iloczynem „po współrzędnych” dwóch wektorów. Zbiór wektorów wyboru spełnia dodatkowy warunek

$$\sum_{i=1}^{\bar{r}} \boldsymbol{\rho}_i = (1, \dots, 1)^T, \quad (4.2)$$

który determinuje jednokrotny wybór każdego elementu bazy w reprezentacji ruchu. Rodzina reprezentacji (4.1) zależy od wyboru bazy i z niej dziedziczy nazwę, podobnie jak i współrzędne \mathbf{h} (Halla, Lyndona, Chibrikova). Dla przypomnienia, konkatencję eksponent w (4.1) należy czytać od lewej strony ku prawej. Przykładowo: $e^{h_1 \mathbf{B}_1 + h_2 \mathbf{B}_2} e^{h_3 \mathbf{B}_3}$ oznacza, że na stan przemieszczony przez akcję $e^{h_1 \mathbf{B}_1 + h_2 \mathbf{B}_2}$ działamy $e^{h_3 \mathbf{B}_3}$. Należy pamiętać także, że w ogólności akcje nie komutują, więc kolejność włączania poszczególnych eksponent jest istotna. Wektorami wyboru dla powyższego przykładu są $\boldsymbol{\rho}_1 = (1, 1, 0)$, $\boldsymbol{\rho}_2 = (0, 0, 1)$ ($r = 3$, $\bar{r} = 2$).

Dla reprezentacji „w przód” (3.5) wektory wyboru $\boldsymbol{\rho}_i = (0, \dots, 0, \underbrace{1}_i, 0, \dots, 0)^T$, a dla reprezentacji „wstecz” (3.6) – $\boldsymbol{\rho}_i = (0, \dots, 0, \underbrace{1}_{r-i+1}, 0, \dots, 0)^T$, $i \in \{1, \dots, r\}$. Dla

Tablica 4.1. Liczba reprezentacji $F(r)$ dla różnych długości reprezentacji r

r	1	2	3	4	5	6	7	8
$F(r)$	1	3	13	75	541	4683	47293	545836

obydwu literaturowych reprezentacji $\bar{r} = r$. Istnieje szczególna reprezentacja, którą nazywamy kanoniczną, unikająca problemu wyboru kolejności stosowania eksponent. Dla reprezentacji kanonicznej jest dokładnie jeden wektor wyboru złożony z samych jedynek, $\bar{r} = 1$. Liczbę wszystkich reprezentacji (4.1), (4.2), dla zadanego r , opisuje wzór zaczerpnięty z pracy [20]

$$F(r) = \sum_{k=1}^r \sum_{i=1}^r (-1)^{k-i} \binom{k}{i} i^r, \quad (4.3)$$

a wartości $F(r)$ dla małych argumentów r zamieszczono w tab. 4.1.

Rozważmy jak w praktyce można wyliczyć równanie CFS. Dla uogólnionej reprezentacji ruchu (4.1) lewa strona równania (3.7) wynosi

$$\begin{aligned} \mathbf{S}^{-1}(t)\dot{\mathbf{S}}(t) &= \prod_{j=1}^{\bar{r}} e^{-\langle \boldsymbol{\rho}_{\bar{r}+1-j} \star \mathbf{h}(t), \mathbf{B} \rangle} \left(\sum_{i=1}^{\bar{r}} \prod_{j=1}^i e^{\langle \boldsymbol{\rho}_j \star \mathbf{h}(t), \mathbf{B} \rangle} \langle \boldsymbol{\rho}_i \star \dot{\mathbf{h}}(t), \mathbf{B} \rangle \prod_{j=i+1}^{\bar{r}} e^{\langle \boldsymbol{\rho}_j \star \mathbf{h}(t), \mathbf{B} \rangle} \right) \\ &= \sum_{i=1}^{\bar{r}} \left(\left(\prod_{j=i+1}^{\bar{r}} e^{\langle \boldsymbol{\rho}_j \star \mathbf{h}(t), \mathbf{B} \rangle} \right)^{-1} \langle \boldsymbol{\rho}_i \star \dot{\mathbf{h}}(t), \mathbf{B} \rangle \prod_{j=i+1}^{\bar{r}} e^{\langle \boldsymbol{\rho}_j \star \mathbf{h}(t), \mathbf{B} \rangle} \right). \end{aligned} \quad (4.4)$$

Każdy składnik wynikowej sumy jest postaci $e^{-\mathbf{A}\mathbf{C}e^{\mathbf{A}}}$ pomnożony przez odpowiednią składową $\dot{\mathbf{h}}(t)$ gdzie \mathbf{A}, \mathbf{C} są kombinacjami jednomianów Liego. Biorąc pod uwagę znaną identyczność [70]

$$e^{-\mathbf{A}\mathbf{C}e^{\mathbf{A}}} = \sum_{i=0}^{\infty} \frac{(-1)^i}{i!} \text{ad}_{\mathbf{A}}^i \mathbf{C} = \mathbf{C} - [\mathbf{A}, \mathbf{C}] + \frac{1}{2}[\mathbf{A}, [\mathbf{A}, \mathbf{C}]] - \frac{1}{6}[\mathbf{A}, [\mathbf{A}, [\mathbf{A}, \mathbf{C}]]] \dots, \quad (4.5)$$

dla rozwiązania równania (3.7) wystarczy posiadać wzór na wyrażenie $e^{\mathbf{A}}e^{\mathbf{C}}$ oraz algorytm transformujący dowolną kombinację jednomianów Liego w kombinację elementów bazowych (algorytm ten opisano w pracy [18] dla bazy Halla). Klasyczne wyrażenie na

$$e^{\mathbf{A}}e^{\mathbf{C}} = e^{\mathbf{Z}}, \quad (4.6)$$

zostało zaczerpnięte z raportu [28], gdzie \mathbf{Z} do warstwy 5-tej włącznie jest zadane przez

$$\begin{aligned} \mathbf{Z} &= \mathbf{A} + \mathbf{C} + \frac{1}{2}[\mathbf{A}, \mathbf{C}] + \frac{1}{12}[\mathbf{A}, [\mathbf{A}, \mathbf{C}]] - \frac{1}{12}[\mathbf{C}, [\mathbf{A}, \mathbf{C}]] - \frac{1}{24}[\mathbf{C}, [\mathbf{A}, [\mathbf{A}, \mathbf{C}]]] + \\ &\quad - \frac{1}{720}[\mathbf{A}, [\mathbf{A}, [\mathbf{A}, [\mathbf{A}, \mathbf{C}]]]] - \frac{1}{120}[[\mathbf{A}, \mathbf{C}], [\mathbf{A}, [\mathbf{A}, \mathbf{C}]]] - \frac{1}{180}[\mathbf{C}, [\mathbf{A}, [\mathbf{A}, [\mathbf{A}, \mathbf{C}]]]] + \\ &\quad - \frac{1}{360}[[\mathbf{A}, \mathbf{C}], [\mathbf{C}, [\mathbf{A}, \mathbf{C}]]] + \frac{1}{180}[\mathbf{C}, [\mathbf{C}, [\mathbf{A}, [\mathbf{A}, \mathbf{C}]]]] + \frac{1}{720}[\mathbf{C}, [\mathbf{C}, [\mathbf{C}, [\mathbf{A}, \mathbf{C}]]]] + \dots \end{aligned} \quad (4.7)$$

Dla układów nilpotentnych szereg (4.7) jest skończony.

Ilustrując rozwiązanie równania (3.7) dla bardziej skomplikowanej reprezentacji ruchu rozważmy następujący przykład.

Przykład 17. Niech układ o stopniu nilpotentności równym 3 będzie rozpięty przez dwa generatory $\mathbf{B}_1, \mathbf{B}_2$, a reprezentacja ruchu jest typu „w przód”

$$\mathbf{S}(t) = e^{h_1(t)\mathbf{B}_1} e^{h_2(t)\mathbf{B}_2} e^{h_3(t)\mathbf{B}_3} e^{h_4(t)\mathbf{B}_4} e^{h_5(t)\mathbf{B}_5}, \quad (4.8)$$

gdzie $\mathbf{B}_3 = [\mathbf{B}_1, \mathbf{B}_2]$, $\mathbf{B}_4 = [\mathbf{B}_1, \mathbf{B}_3]$, $\mathbf{B}_5 = [\mathbf{B}_2, \mathbf{B}_3]$. Dla reprezentacji (4.8) równanie (4.4) jest następujące

$$\begin{aligned} \mathbf{S}^{-1}\dot{\mathbf{S}} = & e^{-h_5\mathbf{B}_5} e^{-h_4\mathbf{B}_4} e^{-h_3\mathbf{B}_3} e^{-h_2\mathbf{B}_2} e^{-h_1\mathbf{B}_1} \dot{h}_1\mathbf{B}_1 e^{h_1(t)\mathbf{B}_1} e^{h_2(t)\mathbf{B}_2} e^{h_3\mathbf{B}_3} e^{h_4(t)\mathbf{B}_4} e^{h_5(t)\mathbf{B}_5} + \\ & + e^{-h_5\mathbf{B}_5} e^{-h_4\mathbf{B}_4} e^{-h_3\mathbf{B}_3} e^{-h_2\mathbf{B}_2} \dot{h}_2\mathbf{B}_2 e^{h_2(t)\mathbf{B}_2} e^{h_3\mathbf{B}_3} e^{h_4(t)\mathbf{B}_4} e^{h_5(t)\mathbf{B}_5} + \\ & + e^{-h_5\mathbf{B}_5} e^{-h_4\mathbf{B}_4} e^{-h_3\mathbf{B}_3} \dot{h}_3\mathbf{B}_3 e^{h_3\mathbf{B}_3} e^{h_4(t)\mathbf{B}_4} e^{h_5(t)\mathbf{B}_5} + \\ & + e^{-h_5\mathbf{B}_5} e^{-h_4\mathbf{B}_4} \dot{h}_4\mathbf{B}_4 e^{h_4(t)\mathbf{B}_4} e^{h_5(t)\mathbf{B}_5} + e^{-h_5\mathbf{B}_5} \dot{h}_5\mathbf{B}_5 e^{h_5(t)\mathbf{B}_5}. \end{aligned} \quad (4.9)$$

Zauważmy, że elementy równania (4.9) mogą być grupowane na dwa sposoby. Pierwsze wyrażenie w (4.9) może zostać zapisane jako

$$\left(e^{-h_5\mathbf{B}_5} e^{-h_4\mathbf{B}_4} e^{-h_3\mathbf{B}_3} e^{-h_2\mathbf{B}_2} e^{-h_1\mathbf{B}_1} \right) \dot{h}_1\mathbf{B}_1 \left(e^{h_1(t)\mathbf{B}_1} e^{h_2(t)\mathbf{B}_2} e^{h_3\mathbf{B}_3} e^{h_4(t)\mathbf{B}_4} e^{h_5(t)\mathbf{B}_5} \right), \quad (4.10)$$

lub

$$e^{-h_5\mathbf{B}_5} \left\{ e^{-h_4\mathbf{B}_4} \left\{ e^{-h_3\mathbf{B}_3} \left\{ e^{-h_2\mathbf{B}_2} \left\{ e^{-h_1\mathbf{B}_1} \dot{h}_1\mathbf{B}_1 e^{h_1(t)\mathbf{B}_1} \right\} e^{h_2(t)\mathbf{B}_2} \right\} e^{h_3\mathbf{B}_3} \right\} e^{h_4(t)\mathbf{B}_4} \right\} e^{h_5(t)\mathbf{B}_5}.$$

Weźmy pod uwagę drugi sposób grupowania. Wykorzystując wielokrotnie równanie (4.5) oraz biorąc pod uwagę stopień nilpotentności równy 3, zależność (4.9) przyjmuje postać

$$\mathbf{S}^{-1}\dot{\mathbf{S}} = \dot{h}_1(\mathbf{B}_1 + h_2\mathbf{B}_3 + h_3\mathbf{B}_4 - \frac{1}{2}h_2^2\mathbf{B}_5) + \dot{h}_2(\mathbf{B}_2 + h_3\mathbf{B}_5) + \dot{h}_3\mathbf{B}_3 + \dot{h}_4\mathbf{B}_4 + \dot{h}_5\mathbf{B}_5.$$

Porównując współczynniki identycznych elementów bazowych \mathbf{B}_i , $i = 1, \dots, 5$ w równaniu (3.7) powstaje równanie CFS

$$\begin{cases} \dot{h}_1(t) = v_1(t), \\ \dot{h}_2(t) = v_2(t) \\ \dot{h}_3(t) = -h_2(t)v_1(t) + v_3(t) \\ \dot{h}_4(t) = -h_3(t)v_1(t) + v_4(t) \\ \dot{h}_5(t) = \frac{1}{2}h_2^2v_1 - h_3(t)v_2(t) + v_5(t). \end{cases} \quad (4.11)$$

Otrzymane równania CFS (3.16), (4.11) dla wybranych reprezentacji mają szczególną postać. Ogólny wynik dotyczący postaci równania CFS podaje twierdzenie [21] (prawdziwe dla dowolnej reprezentacji ruchu, jednak jego dowód obejmuje reprezentacje z $r = \bar{r}$, dla których możliwe jest kompaktowe przedstawienie pochodnych $e^{(\rho_i \star h(t), \mathbf{B})}$, porównaj przykład 18)

Twierdzenie 1. *Równanie CFS odpowiadające reprezentacji ruchu (4.1), (4.2) jest postaci*

$$\dot{\mathbf{h}}_i(t) = \sum_{k=1}^i W_{i,k}(\mathbf{h}(t)) \mathbf{v}_k(t), \quad i = 1, \dots, r, \quad (4.12)$$

gdzie elementy bazy \mathbf{B}_i , $i = 1, \dots, r$ odpowiadają współrzędnym h_i , $W_{i,k}$ są jednorodnymi wielomianami zmiennej \mathbf{h} (niektóre z nich mogą zniknąć), takimi że $\deg(W_{i,k}(\mathbf{h})\mathbf{v}_k) = \deg(h_i)$ (z definicji $\deg(h_i) = \deg(\dot{h}_i) = \deg(\mathbf{v}_i) = \deg(\mathbf{B}_i)$, oraz wszystkie stałe jednomiany mają stopień równy 0).

Dowód: Uwzględniając (4.4), (4.5) wyrażenie $\mathbf{S}^{-1}(t)\dot{\mathbf{S}}(t)$ jest postaci $\sum_{i=1}^r f_i(\mathbf{h}, \dot{\mathbf{h}})\mathbf{B}_i$, więc wykorzystując równanie (3.7) mamy $\forall_{i=1, \dots, r} f_i(\mathbf{h}, \dot{\mathbf{h}}) = \mathbf{v}_i$. Przeanalizujemy postać wyrażen na f_i podczas wyprowadzania równania CFS wykorzystując (4.4). By móc zastosować (4.5), należy wyliczyć $\prod_{j=i+1}^{\bar{r}} e^{(\rho_j * \mathbf{h}(t), \mathbf{B}_j)}$, która jest $\bar{r} - i$ elementową konkatencją o składowych $e^{(\rho_j * \mathbf{h}(t), \mathbf{B}_j)}$. Stosując odpowiednio wiele razy zależność (4.6), z właściwie dobranymi \mathbf{A} , \mathbf{C} , stosując antysymetrię nawiasu Liego i tożsamość Jacobiego w przypadku gdy nawias Liego elementów bazowych \mathbf{B}_i , \mathbf{B}_j nie jest bazowym, otrzymujemy wyrażenie $e^{\sum_{i=1}^r \beta_i(\mathbf{h})\mathbf{B}_i}$, gdzie liczba składowych eksponenty jest ograniczona do r ze względu na nilpotentność. Twierdzimy, że każdy element $\beta_i(\mathbf{h})$ ma stopień równy stopniowi \mathbf{B}_i . Z definicji mamy $\deg(h_i) = \deg(\mathbf{B}_i)$. Jedynymi operacjami przekształcającymi wyrażenia są: nawias Liego, który zachowuje stopień w sensie: jeśli $[h_i \mathbf{B}_i, h_j \mathbf{B}_j] = h_i h_j [\mathbf{B}_i, \mathbf{B}_j]$, to $\deg(h_i h_j) = \deg([\mathbf{B}_i, \mathbf{B}_j])$, tożsamość Jacobiego, która także zachowuje stopień swojego argumentu (gdy przy pomocy tożsamości Jacobiego wyrażamy jeden niebazowy jednomian Liego jako kombinację bazowych), oraz antysymetria nawiasu Liego, która zmienia jedynie znaki współczynników, a nie ich stopnie. Zatem każda składowa wielomianu $\beta_i(\mathbf{h})$ dziedziczy stopień po elemencie bazowym \mathbf{B}_i , a więc wielomian jest jednorodny o stopniu równym $\deg(\mathbf{B}_i)$. Powracamy więc do (4.4), które obecnie przyjmuje postać

$$\mathbf{S}^{-1}(t)\dot{\mathbf{S}}(t) = \sum_{i=1}^r e^{-\sum_{j=1}^r \beta_{i,j}(\mathbf{h})\mathbf{B}_j} \dot{h}_i \mathbf{B}_i e^{\sum_{j=1}^r \beta_{i,j}(\mathbf{h})\mathbf{B}_j} \quad (4.13)$$

gdzie $\beta_{i,j}(\mathbf{h})$, $i, j = 1, \dots, r$ są wielomianami jednorodnymi stopnia j . Wykorzystując (4.5) oraz dwuliniowość nawiasu Liego, równanie (4.13) przyjmuje następującą postać

$$\begin{aligned} \sum_{i=1}^r \mathbf{B}_i \mathbf{v}_i &= \mathbf{S}^{-1}(t)\dot{\mathbf{S}}(t) \\ &= \sum_{i=1}^r \dot{h}_i (\mathbf{B}_i - [\sum_{j_1=1}^r \beta_{i,j_1} \mathbf{B}_{j_1}, \mathbf{B}_i] + \frac{1}{2} [\sum_{j_2=1}^r \beta_{i,j_2} \mathbf{B}_{j_2}, [\sum_{j_1=1}^r \beta_{i,j_1} \mathbf{B}_{j_1}, \mathbf{B}_i]] + \dots) \\ &= \sum_{i=1}^r (\dot{h}_i \mathbf{B}_i - \dot{h}_i \sum_{j_1=1}^r \beta_{i,j_1} [\mathbf{B}_{j_1}, \mathbf{B}_i] + \frac{1}{2} \dot{h}_i \sum_{j_2, j_1=1}^r \beta_{i,j_2} \beta_{i,j_1} [\mathbf{B}_{j_2}, [\mathbf{B}_{j_1}, \mathbf{B}_i]] + \dots) \end{aligned} \quad (4.14)$$

Argumentując podobnie jak uprzednio, dla ustalonego i , wyrażenia pojawiające się przy $[\mathbf{B}_{j_1}, \mathbf{B}_i]$, $[\mathbf{B}_{j_2}, [\mathbf{B}_{j_1}, \mathbf{B}_i]]$, mają stopień równy $i+j_1$, $i+j_1+j_2$, ponieważ $\beta_{i,j}$ dziedziczy stopień po \mathbf{B}_j , natomiast \dot{h}_i po \mathbf{B}_i (dla jednomianów Liego wyższych stopni rozumowanie analogiczne).

Wynikowe równanie CFS będzie analizowane indukcyjnie, warstwa po warstwie. W równaniu (4.14), każdy \mathbf{B}_s należący do pierwszej warstwy ($\deg(\mathbf{B}_s) = 1$) jest unikalny i mnożony przez \dot{h}_s . Ponieważ każdy inny (prócz \mathbf{B}_s) bazowy element, mnożący \dot{h}_s w szeregu (4.14) ma stopień wyższy niż 1, więc \mathbf{B}_s nie może być generowany w inny sposób. Zatem równania CFS dla pierwszej warstwy są następujące

$$\dot{h}_i(t) = v_i(t). \quad (4.15)$$

Teraz zajmijmy się wyższymi warstwami. Każdy element \mathbf{B}_s który należy do ustalonej warstwy jest generowany jednoznacznie przez pierwszy składnik w równaniu (4.14) i jest mnożony przez \dot{h}_s . Może również pojawić się jako iloczyn, kiedy generuje się \dot{h}_i dla wcześniejszej warstwy, pod warunkiem, że $i + \sum_{k=1}^{s-1} j_k = \deg(\mathbf{B}_s) \stackrel{\text{def}}{=} d_s$. W tym przypadku mnożąc przez $\sum_{k=1}^{s-1} \dot{h}_k \tilde{W}_{s,k}(\mathbf{h})$, gdzie $\tilde{W}_{s,k}$, są pewnymi (być może zerującymi się) wielomianami o stopniu $d_s - \deg(h_k)$. Po podstawieniu poprzednio otrzymanego \dot{h}_k jako wielomian \mathbf{v} i \mathbf{h} stopnia $\deg(h_k)$, otrzymujemy postulowaną postać

$$\dot{h}_s(t) + \sum_{k=1}^{s-1} \tilde{W}_{s,k}(\mathbf{h}(t)) \dot{h}_k(t) = v_s(t) \quad \Rightarrow \quad \dot{h}_s(t) = \sum_{k=1}^s W_{s,k}(\mathbf{h}(t)) v_k(t), \quad (4.16)$$

co kończy dowód. □

Konsekwencją twierdzenia jest kilka własności równania CFS:

Własność 1. Dla każdej warstwy, prawa strona równań CFS na \dot{h}_s zawiera v_s ze współczynnikiem równym 1. W szczególności, równania CFS dla pierwszej warstwy nie zależą od żadnych elementów \mathbf{h} .

Własność 2. Prawa strona równań CFS na \dot{h}_s jest liniowa ze względu \mathbf{v} i nie zależy od zmiennych v_k , $k = s+1, \dots, r$ odpowiadających elementom $B_{s+1}, B_{s+2}, \dots, B_r$.

Własność 3. Wielomiany $\tilde{W}_{s,k}$, $s, k = 1, \dots$, pojawiające się w równaniach CFS na \dot{h}_s , są stopnia $\deg(\mathbf{B}_s) - \deg(h_k) > 0$ i nie zawierają żadnych elementów \mathbf{h} z indeksami odpowiadającymi stopniowi \dot{h}_s , ani żadnym innym z wyższych warstw.

4.1. Algorytmizacja równań CFS

Drugi etap algorytmu LS planowania ruchu układów nilpotentnych polega na uzyskaniu równań różniczkowych CFS dla wybranej reprezentacji ruchu (4.1) i wybranej bazy algebry Liego. Rozwiązanie równań CFS pozwala na uzyskanie współczynników bazowych opisujących interwały ruchu wzdłuż pól wektorowych, których konkatenacja

stanowi reprezentację ruchu. Wyliczenie równań CFS jest tym bardziej skomplikowane im wyższy stopień nilpotentności układu i większa liczba generatorów. W literaturze metod planowania ruchu wyróżniamy dwa sposoby wyliczenia CFS. Pierwszy zaproponował Sussmann w pracy [98], a jego cechą charakterystyczną jest przeprowadzenie obliczeń w dziedzinie pól wektorowych. Drugi, powstały na potrzeby algorytmu badanego w niniejszej dysertacji, zaproponował Sussmann z Lafferrierem [53, 54, 55], a jego domeną jest działanie na formalnym odpowiedniku równania rozszerzonego, gdzie wszelkich obliczeń dokonuje się w wolnej grupie Liego i jej algebrze. Celem niniejszego podrozdziału jest, w miarę szczegółowe, przedstawienie obu sposobów generacji CFS, oraz podanie ich zalet i wad. Na tle znanych sposobów uzyskania równań CFS, przedstawiony zostanie autorski algorytm wyliczenia równań, w pełni automatyczny, efektywny, a także nie wymagający ustalania żadnych parametrów.

4.1.1. CFS w dziedzinie pól wektorowych

Pierwszym sposobem wyznaczenia równań CFS jest procedura wykorzystująca operacje przeprowadzane w dziedzinie pól wektorowych, oparta na formalizmie pochodzącym z prac [1, 98]. Niech \mathbf{f} oznacza gładkie pole wektorowe, a dla opisu strumienia zostanie wykorzystana wykładnicza (eksponencjalna) notacja. Strumień zapisuje się po prawej stronie konfiguracji $t \rightarrow \mathbf{q}e^{t\mathbf{f}}$ co oznacza krzywą całkową, która przechodzi przez \mathbf{q} w chwili $t = 0$. Zgodnie z tą konwencją także pole wektorowe zapisujemy z prawej strony $\mathbf{q}\mathbf{f}$, co oznacza $\mathbf{f}(\mathbf{q})$. Konsekwentnie, różniczka strumienia opisana jest zależnością $\frac{d}{dt}e^{h(t)\mathbf{f}} = e^{h(t)\mathbf{f}}\dot{h}(t)\mathbf{f}$, gdzie $h(t)$ to dowolna funkcja, przynajmniej raz różniczkowalna. Dla prezentowanego formalizmu spełnione są następujące własności wprowadzone w stylu i na bazie pracy [98]:

Lemat 1. *o przemienności pola i strumienia identycznego pola*

$$h_\alpha(t)\mathbf{f}e^{h_\beta(t)\mathbf{f}} = e^{h_\beta(t)\mathbf{f}}h_\alpha(t)\mathbf{f}.$$

Dowód: z zależności (2.27) i (nie)komutującego mnożenia mamy tezę

$$h_\alpha(t)\mathbf{f}e^{h_\beta(t)\mathbf{f}} = h_\alpha(t)\mathbf{f}\left(\mathbf{I} + \sum_{k=1}^{\infty} \left(\frac{h_\beta^k(t)}{k!}\mathbf{f}^k\right)\right) = \left(\mathbf{I} + \sum_{k=1}^{\infty} \left(\frac{h_\beta^k(t)}{k!}\mathbf{f}^k\right)\right)h_\alpha(t)\mathbf{f} = e^{h_\beta(t)\mathbf{f}}h_\alpha(t)\mathbf{f}.$$

□

Lemat 2. *o różniczce pola otoczonego sprzężonymi strumieniami innego pola*

$$\frac{d}{dt}\left(\bar{\mathbf{q}}e^{-h(t)\mathbf{g}}\mathbf{f}e^{h(t)\mathbf{g}}\right) = -\bar{\mathbf{q}}e^{-h(t)\mathbf{g}}\dot{h}(t)[\mathbf{g}, \mathbf{f}]e^{h(t)\mathbf{g}}.$$

Dowód: Pamiętając o (2.28), definicji nawiasu Liego $[\mathbf{g}, \mathbf{f}] = \mathbf{g}\mathbf{f} - \mathbf{f}\mathbf{g}$, i korzystając z Lematu 1, zapisujemy

$$\begin{aligned} \frac{d}{dt} \left(\bar{\mathbf{q}} e^{-\mathbf{h}(t)\mathbf{g}} \mathbf{f} e^{\mathbf{h}(t)\mathbf{g}} \right) &= -\bar{\mathbf{q}} e^{-\mathbf{h}(t)\mathbf{g}} \dot{\mathbf{h}}(t) \mathbf{g} \mathbf{f} e^{\mathbf{h}(t)\mathbf{g}} + \bar{\mathbf{q}} e^{-\mathbf{h}(t)\mathbf{g}} \mathbf{f} e^{\mathbf{h}(t)\mathbf{g}} \dot{\mathbf{h}}(t) \mathbf{g} = \\ &= -\bar{\mathbf{q}} e^{-\mathbf{h}(t)\mathbf{g}} \dot{\mathbf{h}}(t) \mathbf{g} \mathbf{f} e^{\mathbf{h}(t)\mathbf{g}} + \bar{\mathbf{q}} e^{-\mathbf{h}(t)\mathbf{g}} \dot{\mathbf{h}}(t) \mathbf{f} \mathbf{g} e^{\mathbf{h}(t)\mathbf{g}} = -\bar{\mathbf{q}} e^{-\mathbf{h}(t)\mathbf{g}} \dot{\mathbf{h}}(t) [\mathbf{g}, \mathbf{f}] e^{\mathbf{h}(t)\mathbf{g}}, \end{aligned}$$

co kończy dowód. □

Lemat 3. *o zamianie pola i strumienia komutujących pól*

Jeżeli $[\mathbf{f}, \mathbf{g}] = \mathbf{0}$, to

$$\bar{\mathbf{q}} e^{\mathbf{h}(t)\mathbf{g}} \mathbf{f} = \bar{\mathbf{q}} \mathbf{f} e^{\mathbf{h}(t)\mathbf{g}}.$$

Dowód: z założenia i Lematu 2 otrzymujemy równanie $\frac{d}{dt} (\bar{\mathbf{q}} e^{\mathbf{h}(t)\mathbf{g}} \mathbf{f} e^{-\mathbf{h}(t)\mathbf{g}}) = \mathbf{0}$, które po scałkowaniu, z warunkiem początkowym $\bar{\mathbf{q}} e^{\mathbf{0}} \mathbf{f} e^{\mathbf{0}} = \bar{\mathbf{q}} \mathbf{f}$, wiedzie do

$$\bar{\mathbf{q}} e^{\mathbf{h}(t)\mathbf{g}} \mathbf{f} e^{-\mathbf{h}(t)\mathbf{g}} = \text{const.} = \bar{\mathbf{q}} \mathbf{f}.$$

Prawostronne pomnożenie równania przez $e^{\mathbf{h}(t)\mathbf{g}}$ prowadzi do tezy lematu. □

Lemat 4. *Jeżeli $[\mathbf{g}, [\mathbf{g}, \mathbf{f}]] = \mathbf{0}$, to*

$$\bar{\mathbf{q}} e^{-\mathbf{h}(t)\mathbf{g}} \mathbf{f} e^{\mathbf{h}(t)\mathbf{g}} = \bar{\mathbf{q}} (\mathbf{f} - \mathbf{h}(t) [\mathbf{g}, \mathbf{f}]).$$

Dowód: Wykorzystując Lemat 2 oraz założenie $[\mathbf{g}, [\mathbf{g}, \mathbf{f}]] = \mathbf{0}$ umożliwiające korzystanie z Lematu 3, można zamienić strumień $e^{\mathbf{h}(t)\mathbf{g}}$ z polem $[\mathbf{g}, \mathbf{f}]$ w zależności poniższej

$$\frac{d}{dt} \bar{\mathbf{q}} e^{-\mathbf{h}(t)\mathbf{g}} \mathbf{f} e^{\mathbf{h}(t)\mathbf{g}} = -\bar{\mathbf{q}} e^{-\mathbf{h}(t)\mathbf{g}} \dot{\mathbf{h}}(t) [\mathbf{g}, \mathbf{f}] e^{\mathbf{h}(t)\mathbf{g}} = -\bar{\mathbf{q}} \dot{\mathbf{h}}(t) [\mathbf{g}, \mathbf{f}].$$

Całkując powyższe równanie otrzymujemy $\bar{\mathbf{q}} e^{-\mathbf{h}(t)\mathbf{g}} \mathbf{f} e^{\mathbf{h}(t)\mathbf{g}} = -\bar{\mathbf{q}} \mathbf{h}(t) [\mathbf{g}, \mathbf{f}] + \bar{\mathbf{q}} \mathbf{f}$, gdzie $\bar{\mathbf{q}} \mathbf{f}$ pochodzi od warunku początkowego. □

Lemat 5. *Niech $[\mathbf{f}, [\mathbf{f}, [\mathbf{f}, \mathbf{g}]]] = [\mathbf{g}, [\mathbf{f}, [\mathbf{f}, \mathbf{g}]]] = [\mathbf{f}, [\mathbf{g}, [\mathbf{f}, \mathbf{g}]]] = \mathbf{0}$ (czwarta warstwa pól się zeruje) to*

$$\bar{\mathbf{q}} e^{-\mathbf{h}(t)\mathbf{g}} \mathbf{f} e^{\mathbf{h}(t)\mathbf{g}} = \bar{\mathbf{q}} (\mathbf{f} - \mathbf{h}(t) [\mathbf{g}, \mathbf{f}] + \frac{\mathbf{h}^2(t)}{2} [\mathbf{g}, [\mathbf{g}, \mathbf{f}]]).$$

Dowód: Z Lematu 2 oraz 4 (z założeniem $[\mathbf{g}, [\mathbf{g}, [\mathbf{g}, \mathbf{f}]]] = \mathbf{0}$) wynika, że można zamienić strumień $e^{h(t)\mathbf{g}}$ z polem $[\mathbf{g}, \mathbf{f}]$

$$\frac{d}{dt} \bar{\mathbf{q}} e^{-h(t)\mathbf{g}} \mathbf{f} e^{h(t)\mathbf{g}} = -\bar{\mathbf{q}} e^{-h(t)\mathbf{g}} \dot{h}(t) [\mathbf{g}, \mathbf{f}] e^{h(t)\mathbf{g}} = -\bar{\mathbf{q}} \dot{h}(t) ([\mathbf{g}, \mathbf{f}] - h(t) [\mathbf{g}, [\mathbf{g}, \mathbf{f}]]) .$$

Pamiętając, że $\int h'(t)h(t)dt = h^2(t)/2$ można scałkować powyższe równanie

$$\bar{\mathbf{q}} e^{-h(t)\mathbf{g}} \mathbf{f} e^{h(t)\mathbf{g}} = \bar{\mathbf{q}} (-h(t) [\mathbf{g}, \mathbf{f}] + \frac{h^2(t)}{2} [\mathbf{g}, [\mathbf{g}, \mathbf{f}]] + \bar{\mathbf{q}} \mathbf{f},$$

gdzie $\bar{\mathbf{q}} \mathbf{f}$ pochodzi od warunku początkowego. \square

Wyposażeni w zestaw własności pól i strumieni przystępujemy do wyliczenia równań CFS. Rozważmy układ dwuwęściowy

$$\dot{\mathbf{q}}(t) = \mathbf{u}_1(t)\mathbf{f}_1(t) + \mathbf{u}_2(t)\mathbf{f}_2(t) \quad (4.17)$$

o stopniu nilpotentności równym 3. Postulujemy rozwiązanie w bazie Halla w postaci

$$\mathbf{q}(t) = \bar{\mathbf{q}} e^{h_5(t)\mathbf{f}_5} e^{h_4(t)\mathbf{f}_4} e^{h_3(t)\mathbf{f}_3} e^{h_2(t)\mathbf{f}_2} e^{h_1(t)\mathbf{f}_1}, \quad (4.18)$$

gdzie $\mathbf{f}_1 = \mathbf{f}$, $\mathbf{f}_2 = \mathbf{g}$, $\mathbf{f}_3 = [\mathbf{f}, \mathbf{g}]$, $\mathbf{f}_4 = [\mathbf{f}, [\mathbf{f}, \mathbf{g}]]$, $\mathbf{f}_5 = [\mathbf{g}, [\mathbf{f}, \mathbf{g}]]$, a wszystkie pola wyższych stopni się zerują. W wyniku zróżniczkowania (4.18) powstaje

$$\begin{aligned} \dot{\mathbf{q}}(t) = & \bar{\mathbf{q}} e^{h_5(t)\mathbf{f}_5} \dot{h}_5(t)\mathbf{f}_5 \prod_{k=4}^1 e^{h_k(t)\mathbf{f}_k} + \bar{\mathbf{q}} \prod_{k=5}^4 e^{h_k(t)\mathbf{f}_k} \dot{h}_4(t)\mathbf{f}_4 \prod_{k=3}^1 e^{h_k(t)\mathbf{f}_k} + \\ & + \bar{\mathbf{q}} \prod_{k=5}^3 e^{h_k(t)\mathbf{f}_k} \dot{h}_3(t)\mathbf{f}_3 \prod_{k=2}^1 e^{h_k(t)\mathbf{f}_k} + \bar{\mathbf{q}} \prod_{k=5}^2 e^{h_k(t)\mathbf{f}_k} \dot{h}_2(t)\mathbf{f}_2 e^{h_1(t)\mathbf{f}_1} + \bar{\mathbf{q}} \prod_{k=5}^1 e^{h_k(t)\mathbf{f}_k} \dot{h}_1(t)\mathbf{f}_1. \end{aligned} \quad (4.19)$$

Dzięki własności (2.29) otrzymujemy zależność $\bar{\mathbf{q}} \mathbf{f} e^{h(t)\mathbf{g}} = \bar{\mathbf{q}} e^{h(t)\mathbf{g}} e^{-h(t)\mathbf{g}} \mathbf{f} e^{h(t)\mathbf{g}}$, która pozwala na zastosowanie przedstawionych lematów i takie przekształcenie (4.19), by pola \mathbf{f}_i znajdowały się na prawo od wszystkich strumieni. Korzystając z Lematu 3 i nilpotentności ($\forall i \in \{1, \dots, 5\}$ jest $[\mathbf{f}_4, \mathbf{f}_i] = [\mathbf{f}_5, \mathbf{f}_i] = \mathbf{0}$), dwa pierwsze składniki (4.19) są następujące

$$\begin{aligned} \bar{\mathbf{q}} e^{h_5(t)\mathbf{f}_5} \dot{h}_5(t)\mathbf{f}_5 \prod_{k=4}^1 e^{h_k(t)\mathbf{f}_k} &= \bar{\mathbf{q}} \prod_{k=5}^1 e^{h_k(t)\mathbf{f}_k} \dot{h}_5(t)\mathbf{f}_5 \\ \bar{\mathbf{q}} \prod_{k=5}^4 e^{h_k(t)\mathbf{f}_k} \dot{h}_4(t)\mathbf{f}_4 \prod_{k=3}^1 e^{h_k(t)\mathbf{f}_k} &= \bar{\mathbf{q}} \prod_{k=5}^1 e^{h_k(t)\mathbf{f}_k} \dot{h}_4(t)\mathbf{f}_4. \end{aligned}$$

Trzeci składnik (4.19) ma postać

$$\bar{\mathbf{z}} e^{h_3(t)\mathbf{f}_3} \dot{h}_3(t)\mathbf{f}_3 e^{h_2(t)\mathbf{f}_2} e^{h_1(t)\mathbf{f}_1}, \quad \text{gdzie } \bar{\mathbf{z}} = \bar{\mathbf{q}} \prod_{k=5}^4 e^{h_k(t)\mathbf{f}_k}.$$

Wykorzystując Lemat 4 można zamienić pole \mathbf{f}_3 ze strumieniem $e^{h_2(t)\mathbf{f}_2}$,

$$\bar{\mathbf{z}}e^{h_3(t)\mathbf{f}_3}e^{h_2(t)\mathbf{f}_2}\dot{h}_3(t)(\mathbf{f}_3 - h_2(t)[\mathbf{f}_2, \mathbf{f}_3])e^{h_1(t)\mathbf{f}_1}.$$

Ponieważ $[\mathbf{f}_1, [\mathbf{f}_2, \mathbf{f}_3]] = \mathbf{0}$, na mocy Lematu 3 następuje zamiana pola $[\mathbf{f}_2, \mathbf{f}_3]$ ze strumieniem $e^{h_1(t)\mathbf{f}_1}$, natomiast do zamiany \mathbf{f}_3 z $e^{h_1(t)\mathbf{f}_1}$, po raz drugi, wykorzystuje się Lemat 4 i rozważany składnik przyjmuje postać

$$\bar{\mathbf{z}}e^{h_3(t)\mathbf{f}_3}e^{h_2(t)\mathbf{f}_2}e^{h_1(t)\mathbf{f}_1}(\dot{h}_3(t)(\mathbf{f}_3 - h_1(t)[\mathbf{f}_1, \mathbf{f}_3] - h_2(t)[\mathbf{f}_2, \mathbf{f}_3])).$$

Zapiszmy czwarty składnik (4.19) następująco

$$\bar{\mathbf{z}}e^{h_2(t)\mathbf{f}_2}\dot{h}_2(t)\mathbf{f}_2e^{h_1(t)\mathbf{f}_1}, \quad \text{gdzie} \quad \bar{\mathbf{z}} = \bar{\mathbf{q}} \prod_{k=5}^3 e^{h_k(t)\mathbf{f}_k}.$$

aby zamienić pole \mathbf{f}_2 ze strumieniem $e^{h_1(t)\mathbf{f}_1}$ wykorzystuje się Lemat 5, stąd

$$\bar{\mathbf{z}}e^{h_2(t)\mathbf{f}_2}e^{h_1(t)\mathbf{f}_1}\dot{h}_2(t)(\mathbf{f}_2 - h_1(t)[\mathbf{f}_1, \mathbf{f}_2] + \frac{h_1^2(t)}{2}[\mathbf{f}_1, [\mathbf{f}_1, \mathbf{f}_2]]).$$

Po zebraniu wyników częściowych i wyrażeniu nawiasów Liego przez jednomiany \mathbf{f}_i , równanie (4.19) jest postaci

$$\begin{aligned} \dot{\mathbf{q}}(t) = & \mathbf{q}(t)(\dot{h}_5(t)\mathbf{f}_5 + \dot{h}_4(t)\mathbf{f}_4 + \dot{h}_3(t)(\mathbf{f}_3 - h_1(t)\mathbf{f}_4 - h_2(t)\mathbf{f}_5) + \\ & + \dot{h}_2(t)(\mathbf{f}_2 - h_1(t)\mathbf{f}_3 + \frac{h_1^2(t)}{2}\mathbf{f}_4) + \dot{h}_1(t)\mathbf{f}_1). \end{aligned} \quad (4.20)$$

a po zgrupowaniu elementów odpowiadających tym samym polom wektorowym

$$\begin{aligned} \dot{\mathbf{q}}(t) = & \mathbf{q}(t)(\dot{h}_1(t)\mathbf{f}_1 + \dot{h}_2(t)\mathbf{f}_2 + (\dot{h}_3(t) - \dot{h}_2(t)h_1(t))\mathbf{f}_3 + \\ & + (\dot{h}_4(t) - \dot{h}_3(t)h_1(t) + \dot{h}_2(t)\frac{h_1^2(t)}{2})\mathbf{f}_4 + (\dot{h}_5(t) - \dot{h}_3(t)h_2(t))\mathbf{f}_5). \end{aligned} \quad (4.21)$$

Układ rozszerzony, odpowiadający układowi (4.17), o pola wektorowe drugiej i trzeciej warstwy ma postać

$$\dot{\mathbf{q}}(t) = \mathbf{q}(t)(v_1(t)\mathbf{f}_1 + v_2(t)\mathbf{f}_2 + v_3(t)\mathbf{f}_3 + v_4(t)\mathbf{f}_4 + v_5(t)\mathbf{f}_5). \quad (4.22)$$

Porównując (4.21) z (4.22) otrzymujemy równanie CFS

$$\begin{cases} \dot{h}_1(t) &= v_1(t) \\ \dot{h}_2(t) &= v_2(t) \\ \dot{h}_3(t) &= h_1(t)v_2(t) + v_3(t) \\ \dot{h}_4(t) &= \frac{h_1^2(t)}{2}v_2(t) + h_1(t)v_3(t) + v_4(t) \\ \dot{h}_5(t) &= h_1(t)h_2(t)v_2(t) + h_2(t)v_3(t) + v_5(t), \end{cases} \quad (4.23)$$

które scałkowane, na horyzoncie czasu T z warunkiem brzegowym $\mathbf{h}(0) = \mathbf{0}$, opisuje współczynniki Halla $\mathbf{h}(T)$ dla reprezentacji ruchu (4.17). Przedstawiony sposób postępowania formalizujemy jako:

Algorytm 3. CFS generowane metodą pól wektorowych

Krok 1 Dla nilpotentnego, sterowalnego układu (2.36), w jednej z baz algebry Liego (Halla, Lyndona, Chibrikova), wybrać reprezentację ruchu daną równaniem (4.1). Reprezentacja jest scharakteryzowana przez wektor funkcji $\mathbf{h}(t)$ oraz wektory selekcyjne ρ_i , $i = 1, \dots, \bar{r}$.

Krok 2 Dla zadanej konfiguracji $\bar{\mathbf{q}}$, wyliczyć różniczkę reprezentacji ruchu

$$\dot{\mathbf{q}}(t) = \frac{d}{dt} \bar{\mathbf{q}} e^{\langle \rho_1 * \mathbf{h}(t), \mathbf{f} \rangle} \dots e^{\langle \rho_{\bar{r}} * \mathbf{h}(t), \mathbf{f} \rangle} \quad (4.24)$$

gdzie $\mathbf{f} = (\mathbf{f}_1, \dots, \mathbf{f}_{\bar{r}})^T$ jest wektorem wszystkich elementów wybranej bazy do stopnia nilpotentności włącznie.

Krok 3a Wykorzystując uogólnienie Lematów 4 i 5 pochodzące z pracy [65] (argument poniżej podano w nawiasie, gdyż często tę tożsamość zapisuje się bez argumentu) **Lemat 6.**

$$(\bar{\mathbf{q}}) e^{-\mathbf{h}(t)\mathbf{f}} \mathbf{g} e^{\mathbf{h}(t)\mathbf{f}} = (\bar{\mathbf{q}}) \sum_{n=0}^{\infty} \frac{(-\mathbf{h}(t))^n}{n!} \text{ad}_{\mathbf{f}}^n \mathbf{g} = (\bar{\mathbf{q}}) (\mathbf{g} - \mathbf{h}(t)[\mathbf{f}, \mathbf{g}] + \frac{\mathbf{h}^2(t)}{2} [\mathbf{f}, [\mathbf{f}, \mathbf{g}]] + \dots),$$

przekształcić równanie (4.24) tak, by pola wektorowe znalazły się z prawej strony wszystkich strumieni.

Krok 3b Powstałe pola wektorowe wyrazić jako kombinacje liniowe elementów wybranej bazy algebry Liego wykorzystując tożsamość Jacobiego i antysymetrię nawiasu Liego.

Krok 4 Z tak przekształconego równania (4.24) będącego kombinacją liniową elementów bazy oraz równania na sterowania rozszerzone stworzyć układ równań różniczkowych $\dot{\mathbf{h}} = W(\mathbf{h}, \mathbf{v})$ porównując elementy równań znajdujące się przy tych samych elementach bazy.

Krok 5 Dla zadanych sterowań rozszerzonych $\mathbf{v}(\cdot)$, $t \in [0, T]$, wyznaczyć z równania różniczkowego współczynniki bazowe $\mathbf{h}(T)$.

Warto zauważyć, że każda zamiana pola ze strumieniem (Krok 3a) powoduje powstanie nowego, bardziej skomplikowanego pola, bowiem do pola przemieszczanego dołączane jest pole ze strumienia. Po tym przekształceniu nie ma gwarancji, iż nowo powstały nawias Liego należy do bazy. Na przykład w bazie Halla zamieniając miejscami $\dot{\mathbf{h}}_1(t)\mathbf{X}$ z $e^{\mathbf{h}_2(t)[\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]}$ zgodnie z Lematem 6

$$e^{-\mathbf{h}_2(t)[\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]} \dot{\mathbf{h}}_1(t)\mathbf{X} e^{\mathbf{h}_2(t)[\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]}} = \dot{\mathbf{h}}_1(t)(\mathbf{X} + \mathbf{h}_2(t)[\mathbf{X}, [\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]] + \dots),$$

ale $[\mathbf{X}, [\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]]$ nie należy do bazy Halla, gdyż w warstwie czwartej są elementy (pola) $[\mathbf{X}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]$, $[\mathbf{Y}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]$, $[\mathbf{Y}, [\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]]$ (zob. tab. 2.1). Z tożsamości Jacobiego wynika, że $[\mathbf{X}, [\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]] = [\mathbf{Y}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]$. Zatem w Kroku 3 należy pola wyrazić w odpowiedniej bazie algebry Liego (stąd Krok 3b). Krok 3b dla pól z wyższych warstw może

okazać się bardzo skomplikowany i kosztowny obliczeniowo. Algorytm pozwalający wyrazić dowolny jednomian Liego jako kombinację elementów bazy Halla znajduje się w pracy [18].

Podsumowując zaprezentowany sposób uzyskania równań CFS wykorzystujący operacje na polach wektorowych należy stwierdzić, że jest on wyjątkowo żmudny i dla skomplikowanych układów (wiele generatorów i wysoki stopień nilpotentności) znacznie przekracza możliwości ręcznych obliczeń.

4.1.2. Równanie CFS otrzymane z formalnej grupy Liego

Drugim sposobem otrzymywania równań CFS jest metoda bazująca na obliczeniach w formalnej grupie Liego (przykład obliczeniowy można znaleźć w książce LaValla [60]). Wyliczymy jeszcze raz równanie CFS dla układu dwuwęściowego (4.17), nilpotentnego – o stopniu nilpotentności równym 3 i formalnym odpowiedniku równania rozszerzonego zadany przez (3.4), który dla nilpotentnej grupy Liego i bazy Halla przyjmuje postać

$$\dot{\mathbf{S}}(t) = \mathbf{S}(t)(v_1\mathbf{B}_1 + v_2\mathbf{B}_2 + v_3\mathbf{B}_3 + v_4\mathbf{B}_4 + v_5\mathbf{B}_5), \quad (4.25)$$

gdzie \mathbf{B}_i są elementami bazy Halla do stopnia trzeciego włącznie, stąd $r = 5$. Równoważnie (4.25) zapisujemy jako

$$\mathbf{S}^{-1}(t)\dot{\mathbf{S}}(t) = (v_1\mathbf{B}_1 + v_2\mathbf{B}_2 + v_3\mathbf{B}_3 + v_4\mathbf{B}_4 + v_5\mathbf{B}_5). \quad (4.26)$$

Formalnym odpowiednikiem reprezentacji „wstecz” (4.18) jest

$$\mathbf{S}(t) = e^{h_5(t)\mathbf{B}_5} e^{h_4(t)\mathbf{B}_4} e^{h_3(t)\mathbf{B}_3} e^{h_2(t)\mathbf{B}_2} e^{h_1(t)\mathbf{B}_1}, \quad (4.27)$$

gdzie $\mathbf{B}_1, \mathbf{B}_2$ są formalnymi generatorami, a $\mathbf{B}_3 = [\mathbf{B}_1, \mathbf{B}_2]$, $\mathbf{B}_4 = [\mathbf{B}_1, [\mathbf{B}_1, \mathbf{B}_2]]$, $\mathbf{B}_5 = [\mathbf{B}_2, [\mathbf{B}_1, \mathbf{B}_2]]$ jednomianami Liego z warstwy drugiej i trzeciej. Zróżniczkowanie zależności (4.27) prowadzi do

$$\begin{aligned} \dot{\mathbf{S}}(t) = & \dot{h}_5(t)\mathbf{B}_5 e^{h_5(t)\mathbf{B}_5} e^{h_4(t)\mathbf{B}_4} e^{h_3(t)\mathbf{B}_3} e^{h_2(t)\mathbf{B}_2} e^{h_1(t)\mathbf{B}_1} + \dots \\ & \dots + e^{h_5(t)\mathbf{B}_5} e^{h_4(t)\mathbf{B}_4} e^{h_3(t)\mathbf{B}_3} e^{h_2(t)\mathbf{B}_2} \dot{h}_1(t)\mathbf{B}_1 e^{h_1(t)\mathbf{B}_1}. \end{aligned} \quad (4.28)$$

Z tożsamości $\mathbf{S}(t)\mathbf{S}^{-1}(t) = \mathbf{I}$ wynika

$$\mathbf{S}^{-1}(t) = e^{-h_1(t)\mathbf{B}_1} e^{-h_2(t)\mathbf{B}_2} e^{-h_3(t)\mathbf{B}_3} e^{-h_4(t)\mathbf{B}_4} e^{-h_5(t)\mathbf{B}_5}. \quad (4.29)$$

Dla uproszczenia zapisu pomijając argument (t) i podstawiając (4.28), (4.29) do (4.26) otrzymujemy

$$\begin{aligned} \mathbf{S}^{-1}\dot{\mathbf{S}} = & e^{-h_1\mathbf{B}_1} e^{-h_2\mathbf{B}_2} e^{-h_3\mathbf{B}_3} e^{-h_4\mathbf{B}_4} e^{-h_5\mathbf{B}_5} \dot{h}_5\mathbf{B}_5 e^{h_5\mathbf{B}_5} e^{h_4\mathbf{B}_4} e^{h_3\mathbf{B}_3} e^{h_2\mathbf{B}_2} e^{h_1\mathbf{B}_1} \\ & + e^{-h_1\mathbf{B}_1} e^{-h_2\mathbf{B}_2} e^{-h_3\mathbf{B}_3} e^{-h_4\mathbf{B}_4} \dot{h}_4\mathbf{B}_4 e^{h_4\mathbf{B}_4} e^{h_3\mathbf{B}_3} e^{h_2\mathbf{B}_2} e^{h_1\mathbf{B}_1} \\ & + e^{-h_1\mathbf{B}_1} e^{-h_2\mathbf{B}_2} e^{-h_3\mathbf{B}_3} \dot{h}_3\mathbf{B}_3 e^{h_3\mathbf{B}_3} e^{h_2\mathbf{B}_2} e^{h_1\mathbf{B}_1} + \\ & + e^{-h_1\mathbf{B}_1} e^{-h_2\mathbf{B}_2} \dot{h}_2\mathbf{B}_2 e^{h_2\mathbf{B}_2} e^{h_1\mathbf{B}_1} + \\ & + e^{-h_1\mathbf{B}_1} \dot{h}_1\mathbf{B}_1 e^{h_1\mathbf{B}_1}. \end{aligned} \quad (4.30)$$

Rozważany układ jest nilpotentny rzędu 3, więc wszystkie jednomiany Liego stopnia większego niż 3 zerują się tożsamościowo. Stąd skończone rozwinięcia eksponent w formalne odpowiedniki szeregów Taylora

$$e^{\pm h_i \mathbf{B}_i} = \mathbf{I} \pm h_i \mathbf{B}_i + \frac{1}{2} h_i^2 \mathbf{B}_i^2 \pm \frac{1}{6} h_i^3 \mathbf{B}_i^3, \quad i \in \{1, 2\}, \quad e^{\pm h_i \mathbf{B}_i} = \mathbf{I} \pm h_i \mathbf{B}_i, \quad i \in \{3, 4, 5\}, \quad (4.31)$$

gdyż $\deg(\mathbf{B}_3^2) = 4$, a $\deg(\mathbf{B}_4^2) = \deg(\mathbf{B}_5^2) = 6$, zatem $\mathbf{B}_3^2 = \mathbf{B}_4^2 = \mathbf{B}_5^2 = 0$. Następnie podstawiając (4.31) do (4.30), pamiętając o nieprzemiennym mnożeniu, oraz wykorzystując definicję nawiasu Liego (2.23) otrzymujemy

$$\begin{aligned} \mathbf{S}^{-1} \dot{\mathbf{S}} &= \dot{h}_5 \mathbf{B}_5 + \dot{h}_4 \mathbf{B}_4 + \\ &+ (\mathbf{I} - h_1 \mathbf{B}_1)(\mathbf{I} - h_2 \mathbf{B}_2) \dot{h}_3 \mathbf{B}_3 (\mathbf{I} + h_1 \mathbf{B}_1)(\mathbf{I} + h_2 \mathbf{B}_2) + \\ &+ (\mathbf{I} - h_1 \mathbf{B}_1 + \frac{1}{2} h_1^2 \mathbf{B}_1^2)(\mathbf{I} - h_2 \mathbf{B}_2 + \frac{1}{2} h_2^2 \mathbf{B}_2^2) \dot{h}_2 \mathbf{B}_2 (\mathbf{I} + h_2 \mathbf{B}_2 + \frac{1}{2} h_2^2 \mathbf{B}_2^2)(\mathbf{I} + h_1 \mathbf{B}_1 + \frac{1}{2} h_1^2 \mathbf{B}_1^2) \\ &+ (\mathbf{I} - h_1 \mathbf{B}_1 + \frac{1}{2} h_1^2 \mathbf{B}_1^2) \dot{h}_1 \mathbf{B}_1 (\mathbf{I} + h_1 \mathbf{B}_1 + \frac{1}{2} h_1^2 \mathbf{B}_1^2). \end{aligned}$$

z pominiętymi elementami, które po przemnożeniu przez $\dot{h}_i \mathbf{B}_i$ przekroczyłyby zadany rząd nilpotentności. Po wymnożeniu i zgrupowaniu odpowiednich elementów mamy

$$\begin{aligned} \mathbf{S}^{-1} \dot{\mathbf{S}} &= \dot{h}_5 \mathbf{B}_5 + \dot{h}_4 \mathbf{B}_4 + \dot{h}_3 \mathbf{B}_3 + \dot{h}_3 h_1 \overbrace{(\mathbf{B}_3 \mathbf{B}_1 - \mathbf{B}_1 \mathbf{B}_3)}^{-\mathbf{B}_4} + \dot{h}_3 h_2 \overbrace{(\mathbf{B}_3 \mathbf{B}_2 - \mathbf{B}_2 \mathbf{B}_3)}^{-\mathbf{B}_5} + \\ &\dot{h}_2 \mathbf{B}_2 + \dot{h}_2 h_1 \underbrace{(\mathbf{B}_2 \mathbf{B}_1 - \mathbf{B}_1 \mathbf{B}_2)}_{-\mathbf{B}_3} + \frac{1}{2} \dot{h}_2 h_1^2 \underbrace{(\mathbf{B}_1^2 \mathbf{B}_2 - 2 \mathbf{B}_1 \mathbf{B}_2 \mathbf{B}_1 + \mathbf{B}_2 \mathbf{B}_1^2)}_{\mathbf{B}_4} + \dot{h}_1 \mathbf{B}_1. \end{aligned} \quad (4.32)$$

Porównując odpowiednie współczynniki przy \mathbf{B}_i , prawych stron równań (4.26), (4.32) uzyskujemy równanie CFS na współczynniki Halla w postaci identycznej z otrzymaną uprzednio (4.23). Przedstawioną metodę postępowania podsumowuje następująca procedura (Krok 1 oraz Krok 5 identyczne jak w Algorytmie 3).

Algorytm 4. CFS generowane metodą formalnej grupy Liego, wersja 1

Krok 2 Wykorzystując (2.29) oraz różniczkując $\mathbf{S}(t)$ należy wyliczyć lewą stronę równania

$$\mathbf{S}^{-1}(t) \dot{\mathbf{S}}(t) = \sum_{i=1}^r \mathbf{B}_i v_i(t). \quad (4.33)$$

Krok 3 Zastosować wzór na formalne rozwinięcie w szereg Taylora formalnych strumieni (korzystając istotnie z nilpotentności) i obliczyć $\mathbf{S}^{-1}(t) \dot{\mathbf{S}}(t)$. Na podstawie definicji nawiasu Liego (2.23) poskładać odpowiednie jednomiany Liego na nawiasy Liego z odpowiedniej bazy.

Krok 4 Porównać współczynniki identycznych elementów bazy algebry Liego otrzymanego równania z tymi z (4.33) uzyskując układ równań różniczkowych CFS.

Podobnie jak w Algorytmie 3 zamiana kombinacji jednomianów Liego w nawiasy Liego może okazać się trudnym zadaniem dla wyższych warstw, szczególnie gdy jednomiany są elementami składowymi kilku nawiasów Liego. Okazuje się jednak, że nie jest konieczne

przechodzenie z przestrzeni jednomianów do przestrzeni rozpiętej przez elementy wybranej bazy algebry Liego. Modyfikując Krok 3. i 4. Algorytmu 4 otrzymujemy autorski

Algorytm 5. *CFS generowane metodą formalnej grupy Liego, wersja 2*

Krok 3 Zastosować wzór (2.27) na formalne rozwinięcie eksponent (strumieni) w szereg Taylora i wyznaczyć $\mathbf{S}^{-1}(t)\dot{\mathbf{S}}(t)$. Na podstawie definicji nawiasu Liego (2.23) rozwinąć lewą i prawą stronę równania (4.33) do postaci kombinacji jednomianów.

Krok 4 Porównać odpowiednie jednomiany lewej i prawej strony (4.33), uzyskując redundantny układ równań różniczkowych CFS.

Istotną cechą Algorytmu 5 są operacje na jednomianach, a nie jednomianach Liego.

Przykład 18. *Aby zilustrować działanie Algorytmu 5, rozważmy nieco prostszy układ o rzędzie nilpotentności 2 z dwoma generatorami formalnymi \mathbf{X}, \mathbf{Y} .*

Wybrana reprezentacja w bazie Halla zadaje ruch jako

$$\mathbf{S}(t) = e^{h_2(t)\mathbf{Y}} e^{h_1(t)\mathbf{X} + h_3(t)[\mathbf{X}, \mathbf{Y}]} \quad (4.34)$$

Najpierw określimy pochodną $\mathbf{S}(t)$, gdyż jest wyliczana (częściowo) nieco inaczej niż ze wzoru (2.23) ważnego dla jednomianów Liego

$$\begin{aligned} \frac{d}{dt}\mathbf{S}(t) &= \left(\frac{d}{dt}e^{h_2(t)\mathbf{Y}}\right) e^{h_1(t)\mathbf{X} + h_3(t)[\mathbf{X}, \mathbf{Y}]} + e^{h_2(t)\mathbf{Y}} \left(\frac{d}{dt}e^{h_1(t)\mathbf{X} + h_3(t)[\mathbf{X}, \mathbf{Y}]}\right) = \\ & \dot{h}_2(t)\mathbf{Y} e^{h_2(t)\mathbf{Y}} e^{h_1(t)\mathbf{X} + h_3(t)[\mathbf{X}, \mathbf{Y}]} + e^{h_2(t)\mathbf{Y}} \left(\frac{d}{dt}e^{h_1(t)\mathbf{X} + h_3(t)[\mathbf{X}, \mathbf{Y}]}\right) = \\ & e^{h_2(t)\mathbf{Y}} \left(\dot{h}_2(t)\mathbf{Y} e^{h_1(t)\mathbf{X} + h_3(t)[\mathbf{X}, \mathbf{Y}]} + \frac{d}{dt}e^{h_1(t)\mathbf{X} + h_3(t)[\mathbf{X}, \mathbf{Y}]}\right). \end{aligned} \quad (4.35)$$

W ostatnim przekształceniu (4.35) wykorzystaliśmy przemienność $\mathbf{A}e^{\mathbf{A}} = e^{\mathbf{A}}\mathbf{A}$. Wyliczmy zatem ostatnią pochodną przez różniczkowanie odpowiedniego szeregu oraz upraszczając wyrażenia z uwzględnieniem zerowania jednomianów stopnia większego niż 2.

$$\begin{aligned} \frac{d}{dt}e^{h_1(t)\mathbf{X} + h_3(t)[\mathbf{X}, \mathbf{Y}]} &= \frac{d}{dt} \left(\mathbf{I} + h_1(t)\mathbf{X} + h_3(t)[\mathbf{X}, \mathbf{Y}] + \frac{1}{2}(h_1(t)\mathbf{X})^2 \right) = \\ &= \dot{h}_1(t)\mathbf{X} + \dot{h}_3(t)[\mathbf{X}, \mathbf{Y}] + h_1(t)\dot{h}_1(t)\mathbf{X}^2. \end{aligned} \quad (4.36)$$

Teraz dla reprezentacji (4.34), wyliczamy równanie (4.33) (z pominięciem argumentów t)

$$\begin{aligned} \mathbf{S}^{-1}\dot{\mathbf{S}} &= e^{-h_1\mathbf{X} - h_3[\mathbf{X}, \mathbf{Y}]} e^{-h_2\mathbf{Y}} e^{h_2\mathbf{Y}} \left(\dot{h}_2\mathbf{Y} e^{h_1\mathbf{X} + h_3[\mathbf{X}, \mathbf{Y}]} + \frac{d}{dt}e^{h_1\mathbf{X} + h_3[\mathbf{X}, \mathbf{Y}]} \right) = \\ & e^{-h_1\mathbf{X} - h_3[\mathbf{X}, \mathbf{Y}]} \left(\dot{h}_2\mathbf{Y} e^{h_1\mathbf{X} + h_3[\mathbf{X}, \mathbf{Y}]} + \dot{h}_1\mathbf{X} + \dot{h}_3[\mathbf{X}, \mathbf{Y}] + h_1\dot{h}_1\mathbf{X}^2 \right) \end{aligned} \quad (4.37)$$

Biorąc pod uwagę nilpotentność układu otrzymujemy zależności

$$e^{\pm(h_1\mathbf{X} + h_3[\mathbf{X}, \mathbf{Y}])} = \mathbf{I} \pm (h_1\mathbf{X} + h_3[\mathbf{X}, \mathbf{Y}]) + \frac{1}{2}(h_1\mathbf{X})^2 = \mathbf{I} \pm h_1\mathbf{X} \pm h_3\mathbf{X}\mathbf{Y} \mp h_3\mathbf{Y}\mathbf{X} + \frac{1}{2}h_1^2\mathbf{X}^2. \quad (4.38)$$

Następnie równania (4.38) podstawia się do zależności (4.37) (niekomutująco) wymnaża wszystkie elementy, pomija składowe stopnia większego od 2, otrzymując ostatecznie

$$\dot{h}_1 \mathbf{X} + \dot{h}_2 \mathbf{Y} + (\dot{h}_3 - h_1 \dot{h}_2)(\mathbf{XY} - \mathbf{YX}) = \mathbf{X}v_1 + \mathbf{Y}v_2 + (\mathbf{XY} - \mathbf{YX})v_3. \quad (4.39)$$

Porównując współczynniki przy identycznych jednomianach $\mathbf{X}, \mathbf{Y}, \mathbf{XY}, \mathbf{YX}$, otrzymuje się cztery równania, przy czym równania dla \mathbf{XY}, \mathbf{YX} są liniowo zależne. Niezależne są jedynie trzy równania CFS

$$\dot{h}_1 = v_1, \quad \dot{h}_2 = v_2, \quad \dot{h}_3 = h_1 v_2 + v_3. \quad (4.40)$$

Zauważmy, że nawet dla tak prostego przykładu (dwóch generatorów i małego stopnia nilpotentności) obliczenia są dość skomplikowane, choć mechaniczne. Tylko w nielicznych przypadkach równania CFS mogą być wyliczone w rozsądnym czasie bez pomocy wsparcia dedykowanego oprogramowania algebraicznego. Nawet dla pakietów algebraicznych zadanie wyznaczenia niezależnego zbioru równań CFS jest trudne ze względu na liczbę wygenerowanych równań (równą liczbie wszystkich jednomianów o stopniu nie przewyższającym stopnia nilpotencji). Regułą jest, że pojawiają się równania nadmiarowe, gdyż liczba niezależnych równań jest równa liczbie elementów bazy algebry Liego do stopnia nilpotencji włącznie. Liczba równań nadmiarowych szybko rośnie wraz ze wzrostem stopnia nilpotencji. Aby ograniczyć lub uniknąć generowania nadmiarowych równań zaproponowano w pracy [22] algorytm, który polega na zastąpieniu Kroków 3–4 Algorytmu 5 następującymi:

Algorytm 6. *CFS generowane metodą formalnej grupy Liego, wersja 3*

Krok 3a Przekształcić obie strony równania (4.33) do niekomutującej postaci, czyli zastosować (2.27), (2.28), (2.29) oraz zastąpić elementy bazy, czyli \mathbf{B}_i , kombinacjami niekomutujących jednomianów wykorzystując wielokrotnie (2.23) (przykłady takich kombinacji dla początkowych elementów bazy Halla zebrano w tab. 4.2).

Krok 3b Przenieść wszystkie elementy na jedną stronę równania i pogrupować niekomutujące jednomiany do następującej postaci

$$\sum_i f_i(\mathbf{h}, \dot{\mathbf{h}}, \mathbf{v}) \cdot (\text{niekomutujący jednomian})_i = 0,$$

gdzie funkcje $f_i(\mathbf{h}, \dot{\mathbf{h}}, \mathbf{v})$ są liniowe ze względu na $\dot{\mathbf{h}}$. W ten sposób otrzymujemy (nadmiarowy) układ równań: $f_i(\mathbf{h}, \dot{\mathbf{h}}, \mathbf{v}) = 0$, $i = 1, \dots, s$, gdzie s jest liczbą wszystkich niekomutujących jednomianów o stopniu nie większym od stopnia nilpotencji układu.

Krok 4a Każdemu jednomianowi przyporządkować jego indeks, zob. strona 20. Pogrupować otrzymane równania ze względu na stopień jednomianów.

Krok 4b Z każdej grupy usunąć równania nadmiarowe, wykorzystując indeks jednomianu, a więc jeśli indeks jednomianu w danej grupie jest unikalny, to generuje jedno równanie.

Tablica 4.2. Baza Halla rozpięta przez generatory \mathbf{X}, \mathbf{Y} do warstwy piątej włącznie, wraz z parametrami (warstwą i indeksem) oraz niekomutującą postacią elementów

element Bazy Halla	warstwa	indeks	niekomutująca postać
\mathbf{X}	1	(1, 0)	\mathbf{X}
\mathbf{Y}	1	(0, 1)	\mathbf{Y}
$[\mathbf{X}, \mathbf{Y}]$	2	(1, 1)	$\mathbf{XY} - \mathbf{YX}$
$[\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]$	3	(2, 1)	$\mathbf{X}^2\mathbf{Y} - 2\mathbf{XYX} + \mathbf{YX}^2$
$[\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]$	3	(1, 2)	$-\mathbf{XY}^2 + 2\mathbf{YXY} - \mathbf{Y}^2\mathbf{X}$
$[\mathbf{X}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]$	4	(3, 1)	$\mathbf{X}^3\mathbf{Y} - 3\mathbf{X}^2\mathbf{YX} + 3\mathbf{XYX}^2 - \mathbf{YX}^3$
$[\mathbf{Y}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]$	4	(2, 2)	$-\mathbf{X}^2\mathbf{Y}^2 + 2\mathbf{XYXY} - 2\mathbf{YXYX} + \mathbf{Y}^2\mathbf{X}$
$[\mathbf{Y}, [\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]]$	4	(1, 3)	$\mathbf{XY}^3 - 3\mathbf{YXY}^2 + 3\mathbf{Y}^2\mathbf{XY} - \mathbf{Y}^3\mathbf{X}$
$[\mathbf{X}, [\mathbf{X}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]]$	5	(4, 1)	$\mathbf{X}^4\mathbf{Y} - 4\mathbf{X}^3\mathbf{YX} + 6\mathbf{X}^2\mathbf{YX}^2 - 4\mathbf{XYX}^3 + \mathbf{YX}^4$
$[\mathbf{Y}, [\mathbf{X}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]]$	5	(3, 2)	$-\mathbf{X}^3\mathbf{Y}^2 + 3\mathbf{X}^2\mathbf{YXY} - 3\mathbf{XYX}^2\mathbf{Y} + 2\mathbf{YX}^3\mathbf{Y} - 3\mathbf{YX}^2\mathbf{YX} + 3\mathbf{YXYX}^2 - \mathbf{Y}^2\mathbf{X}^3$
$[\mathbf{Y}, [\mathbf{Y}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]]$	5	(2, 3)	$\mathbf{X}^2\mathbf{Y}^3 - 2\mathbf{XYXY}^2 - \mathbf{YX}^2\mathbf{Y}^2 + 4\mathbf{YXYXY} - \mathbf{Y}^2\mathbf{X}^2\mathbf{Y} - 2\mathbf{Y}^2\mathbf{XYX} + \mathbf{Y}^3\mathbf{X}^2$
$[\mathbf{Y}, [\mathbf{Y}, [\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]]]$	5	(1, 4)	$-\mathbf{XY}^4 + 4\mathbf{YXY}^3 - 6\mathbf{Y}^2\mathbf{XY}^2 + 4\mathbf{Y}^3\mathbf{XY} - \mathbf{Y}^4\mathbf{X}$
$[[\mathbf{X}, \mathbf{Y}], [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]$	5	(3, 2)	$-\mathbf{X}^2\mathbf{YXY} + \mathbf{X}^2\mathbf{Y}^2\mathbf{X} + 3\mathbf{XYX}^2\mathbf{Y} - 4\mathbf{XYXYX} + \mathbf{XY}^2\mathbf{X}^2 - 2\mathbf{YX}^3\mathbf{Y} + 3\mathbf{YX}^2\mathbf{YX} - \mathbf{YXYX}^2$
$[[\mathbf{X}, \mathbf{Y}], [\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]]$	5	(2, 3)	$-\mathbf{XYXY}^2 + 3\mathbf{XY}^2\mathbf{XY} - 2\mathbf{XY}^3\mathbf{X} + \mathbf{YX}^2\mathbf{Y}^2 - 4\mathbf{YXYXY} + 3\mathbf{YXY}^2\mathbf{X} + \mathbf{Y}^2\mathbf{X}^2\mathbf{Y} - \mathbf{Y}^2\mathbf{XYX}$

Krok 5 Rozwiązać równania z każdej grupy (w naturalnej kolejności), aby otrzymać składowe $\dot{\mathbf{h}}$ i podstawić uzyskane rezultaty w kolejnych równaniach.

Niektóre podzadania pojawiające się w Algorytmie 6 są wspomagane przez zaawansowane pakiety algebraiczne (np. Mathematica), przykładowo: niekomutujące mnożenie, grupowanie elementów spełniające zadane kryteria, rozwiązywanie liniowego układu równań. Do zadań programisty należy dodanie specjalizowanych procedur wyboru i grupowania równań. Dla ilustracji działania Algorytmu 6, jeszcze raz rozważmy przykład otrzymywania równań CFS dla reprezentacji (4.34). Po Kroku 3a,b otrzymujemy równanie

$$(\dot{h}_1 - v_1)\mathbf{X} + (\dot{h}_2 - v_2)\mathbf{Y} + (-h_1\dot{h}_2 + \dot{h}_3 - v_3)\mathbf{XY} - (-h_1\dot{h}_2 + \dot{h}_3 - v_3)\mathbf{YX} = 0, \quad (4.41)$$

tworzące dwie grupy równań (Krok 4a). Pierwszą grupę równań generują jednomiany stopnia 1 (warstwa 1)

$$\mathbf{X} \rightarrow \dot{h}_1 - v_1 = 0, \quad \mathbf{Y} \rightarrow \dot{h}_2 - v_2 = 0, \quad (4.42)$$

natomiast drugą – jednomiany stopnia 2 (warstwa 2)

$$\mathbf{XY} \rightarrow -h_1\dot{h}_2 + \dot{h}_3 - v_3 = 0, \quad \mathbf{YX} \rightarrow -h_1\dot{h}_2 + \dot{h}_3 - v_3 = 0. \quad (4.43)$$

Ponieważ indeksy niekomutujących jednomianów \mathbf{X}, \mathbf{Y} z pierwszej grupy różnią się (wynoszą odpowiednio $(1, 0)$ i $(0, 1)$), więc obydwie równania wchodzą do zbioru równań niezależnych. Dla drugiej grupy równań indeksy jednomiany \mathbf{XY}, \mathbf{YX} są identyczne i równe $(1, 1)$, zatem tylko jedno z równań powinno być brane pod uwagę podczas wyliczania (4.40) (Krok 4b). Zatem dla przykładowego układu z zadaną reprezentacją ruchu równania CFS wyznaczono jednoznacznie bez żadnych dodatkowych zabiegów.

Rozważmy teraz układ dwuwęściowy o rzędzie nilpotencji równym 3 i warstwy trzeciej rozpiętej przez elementy bazy Halla $[\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]$, $[\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]$. Indeksy dla dwóch podgrup jednomianów odpowiadających warstwie trzeciej są różne i wynoszą $(2, 1)$ i $(1, 2)$, nie ma więc problemu z wyborem odpowiednich równań. Dla pierwszego podzbioru dla warstwy trzeciej wybieramy jedno z równań odpowiadające dowolnemu z niekomutujących jednomianów $\mathbf{X}^2\mathbf{Y}, \mathbf{XYX}, \mathbf{YX}^2$, a do drugiej odpowiadające jednemu z jednomianów $\mathbf{XY}^2, \mathbf{YXY}, \mathbf{XY}^2$. Zatem także w tym przypadku procedura selekcjonuje równania CFS bez żadnych dodatkowych obliczeń. Podobnie się dzieje, gdy rząd nilpotentności wynosi 4. Pomimo większej liczby generowanych równań, jakościowo sytuacja jest identyczna jak dla niższych warstw (por. tab. 4.2). Bardziej interesujący jest przypadek, gdy rząd nilpotentności wynosi 5. Dla podzbioru równań dla jednomianów posiadających unikalne indeksy $(4, 1)$ i $(1, 4)$ odpowiadające elementom bazy Halla $[\mathbf{X}, [\mathbf{X}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]]$ oraz $[\mathbf{Y}, [\mathbf{Y}, [\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]]]$, można wybrać dowolny jednomian i odpowiadające mu równanie (np. $\mathbf{X}^4\mathbf{Y}$ i $\mathbf{X}^4\mathbf{Y}$). Jednakże pojawiają się również podzbiory, które posiadają równe indeksy: przykładowo indeksowi $(2, 3)$ odpowiadają dwa elementy bazy Halla $[\mathbf{Y}, [\mathbf{Y}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]]$, $[[\mathbf{X}, \mathbf{Y}], [\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]]$, a indeksowi $(3, 2)$ – dwa inne $[\mathbf{Y}, [\mathbf{X}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]]$, $[[\mathbf{X}, \mathbf{Y}], [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]$. Dla każdego podzbioru powinno się wybrać dwa równania i rozwiązać dwa liniowe układy równań. Możliwe jest jednak uniknięcie nawet tego nieskomplikowanego obliczenia przez odpowiedni dobór równań. Zauważmy, że pierwszy podzbiór (odpowiadający indeksowi $(2, 3)$) składa się z jednomianów: $\mathbf{X}^2\mathbf{Y}^3$, \mathbf{XYXY}^2 , $\mathbf{YX}^2\mathbf{Y}^2$, \mathbf{YXYXY} , $\mathbf{Y}^2\mathbf{X}^2\mathbf{Y}$, $\mathbf{Y}^2\mathbf{XYX}$, $\mathbf{Y}^3\mathbf{X}^2$, a drugi podzbiór (dla indeksu $(3, 2)$) z jednomianów \mathbf{XYXY}^2 , $\mathbf{XY}^2\mathbf{XY}$, $\mathbf{XY}^3\mathbf{X}$, $\mathbf{YX}^2\mathbf{Y}^2$, \mathbf{YXYXY} , $\mathbf{YXY}^2\mathbf{X}$, $\mathbf{Y}^2\mathbf{X}^2\mathbf{Y}$, $\mathbf{Y}^2\mathbf{XYX}$. Wiele jednomianów pojawia się w obu podzbiorych: \mathbf{XYXY}^2 , $\mathbf{YX}^2\mathbf{Y}^2$, \mathbf{YXYXY} , $\mathbf{Y}^2\mathbf{X}^2\mathbf{Y}$, $\mathbf{Y}^2\mathbf{XYX}$ część tylko w pierwszym podzbiorych: $\mathbf{X}^2\mathbf{Y}^3$, $\mathbf{Y}^3\mathbf{X}^2$, a inne – tylko w drugim: $\mathbf{XY}^2\mathbf{XY}$, $\mathbf{XY}^3\mathbf{X}$, $\mathbf{YXY}^2\mathbf{X}$. Jeśli wybierze się równanie odpowiadające jednomianowi należącemu tylko do jednej podgrupy, będzie to równanie niezależne, wchodzące automatycznie do zestawu równań CFS, dlatego nie ma potrzeby rozważania dodatkowych równań. Ponieważ jednomiany wynikają z istnienia odpowiedniej bazy (w przykładzie Halla), zatem zawsze istnieje możliwość uniknięcia dodatkowego rozwiązywania układu równań w celu wyznaczenia równań CFS.

Dla nilpotentnych układów dwuwęściowych zależność między rzędem nilpotencji, a liczbą generowanych równań Algorytmami 5 i 6 zamieszczono w tab. 4.3. Z danych wynika, że Algorytm 6 jest zdecydowanie bardziej efektywny. Tym algorytmem wygenerowano bardziej zaawansowane równania CFS, praktycznie niemożliwe do przetwarzania ręcznego. W pierwszym z przykładów generujemy równania CFS dla układu

Tablica 4.3. Liczba równań generowanych przez Algorytmy 5 i 6, w zależności od rzędu nilpotencji (dla dwóch formalnych generatorów \mathbf{X}, \mathbf{Y})

Algorytm	rzęd nilpotencji			
	2	3	4	5
5	4	10	22	52
6	3	5	8	14

o dwóch generatorach formalnych \mathbf{X}, \mathbf{Y} , rzędzie nilpotencji równym 5 i reprezentacji „wstecz”

$$\begin{aligned} \mathbf{S}(t) = & e^{h_{14}[\mathbf{X}, \mathbf{Y}], [\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]} e^{h_{13}[\mathbf{X}, \mathbf{Y}], [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]} e^{h_{12}[\mathbf{Y}, [\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]]} e^{h_{11}[\mathbf{Y}, [\mathbf{Y}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]]} \\ & e^{h_{10}[\mathbf{Y}, [\mathbf{X}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]]} e^{h_9[\mathbf{X}, [\mathbf{X}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]]} e^{h_8[\mathbf{Y}, [\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]]} e^{h_7[\mathbf{Y}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]} \\ & e^{h_6[\mathbf{X}, [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]]} e^{h_5[\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]} e^{h_4[\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]} e^{h_3[\mathbf{X}, \mathbf{Y}]} e^{h_2 \mathbf{Y}} e^{h_1 \mathbf{X}}. \end{aligned}$$

Wynikowe równania CFS są następujące

$$\begin{aligned} \dot{h}_1 &= v_1, & \dot{h}_2 &= v_2, & \dot{h}_3 &= h_1 v_2 + v_3, \\ \dot{h}_4 &= (h_1^2 v_2)/2 + h_1 v_3 + v_4, & \dot{h}_5 &= h_1 h_2 v_2 + h_2 v_3 + v_5, \\ \dot{h}_6 &= (h_1^3 v_2)/6 + (h_1^2 v_3)/2 + h_1 v_4 + v_6, \\ \dot{h}_7 &= (h_1^2 h_2 v_2)/2 + h_1 h_2 v_3 + h_2 v_4 + h_1 v_5 + v_7, \\ \dot{h}_8 &= (h_1 h_2^2 v_2)/2 + (h_2^2 v_3)/2 + h_2 v_5 + v_8, \\ \dot{h}_9 &= (h_1^4 v_2)/24 + (h_1^3 v_3)/6 + (h_1^2 v_4)/2 + h_1 v_6 + v_9, \\ \dot{h}_{10} &= (h_1^3 h_2 v_2)/6 + (h_1^2 h_2 v_3)/2 + h_1 h_2 v_4 + (h_1^2 v_5)/2 + h_2 v_6 + h_1 v_7 + v_{10}, \\ \dot{h}_{11} &= (h_1^2 h_2^2 v_2)/4 + (h_1 h_2^2 v_3)/2 + (h_2^2 v_4)/2 + h_1 h_2 v_5 + h_2 v_7 + h_1 v_8 + v_{11}, \\ \dot{h}_{12} &= (h_1 h_2^3 v_2)/6 + (h_2^3 v_3)/6 + (h_2^2 v_5)/2 + h_2 v_8 + v_{12}, \\ \dot{h}_{13} &= (h_1^2 h_3 v_2)/2 + h_1 h_3 v_3 + h_3 v_4 + (h_1^2 v_5)/2 + h_1 v_7 + v_{13}, \\ \dot{h}_{14} &= h_1 h_2 h_3 v_2 + h_2 h_3 v_3 + h_3 v_5 + h_1 v_8 + v_{14}. \end{aligned}$$

Drugi przykład dotyczy układu o trzech generatorach formalnych $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$, rzędzie nilpotencji równym 3 i reprezentacji „w przód” zadanej jako

$$\begin{aligned} \mathbf{S}(t) = & e^{h_1 \mathbf{X}} e^{h_2 \mathbf{Y}} e^{h_3 \mathbf{Z}} e^{h_4 [\mathbf{X}, \mathbf{Y}]} e^{h_5 [\mathbf{X}, \mathbf{Z}]} e^{h_6 [\mathbf{Y}, \mathbf{Z}]} e^{h_7 [\mathbf{X}, [\mathbf{X}, \mathbf{Y}]]} e^{h_8 [\mathbf{X}, [\mathbf{X}, \mathbf{Z}]]} \\ & e^{h_9 [\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]]} e^{h_{10} [\mathbf{Y}, [\mathbf{X}, \mathbf{Z}]]} e^{h_{11} [\mathbf{Y}, [\mathbf{Y}, \mathbf{Z}]]} e^{h_{12} [\mathbf{Z}, [\mathbf{X}, \mathbf{Y}]]} e^{h_{13} [\mathbf{Z}, [\mathbf{X}, \mathbf{Z}]]} e^{h_{14} [\mathbf{Z}, [\mathbf{Y}, \mathbf{Z}]]}. \end{aligned}$$

Algorytm 6 wygenerował następujące równania CFS

$$\begin{aligned} \dot{h}_1 &= v_1, & \dot{h}_2 &= v_2, & \dot{h}_3 &= v_3, & \dot{h}_4 &= -h_2 v_1 + v_4, \\ \dot{h}_5 &= -h_3 v_1 + v_5, & \dot{h}_6 &= -h_3 v_2 + v_6, & \dot{h}_7 &= -h_4 v_1 + v_7, \\ \dot{h}_8 &= -h_5 v_1 + v_8, & \dot{h}_9 &= (h_2^2 v_1)/2 - h_4 v_2 + v_9, \\ \dot{h}_{10} &= -h_6 v_1 - h_5 v_2 + v_{10}, & \dot{h}_{11} &= -h_6 v_2 + v_{11}, \\ \dot{h}_{12} &= h_2 h_3 v_1 + h_6 v_1 - h_4 v_3 + v_{12}, & \dot{h}_{13} &= (h_3^2 v_1)/2 - h_5 v_3 + v_{13}, \\ \dot{h}_{14} &= (h_3^2 v_2)/2 - h_6 v_3 + v_{14}. \end{aligned}$$

Z przedstawionych trzech sposobów wyliczaniu równań CFS bazujących na obliczeniach w formalnej grupie Liego, tylko ostatni jest w pełni zautomatyzowany i efektywny. Algorytm 6 został zaimplementowany w pakiecie Mathematica, wykorzystując wsparcie wbudowanych funkcji tego symbolicznego języka programowania.

4.2. Wpływ reprezentacji ruchu na przebieg trajektorii

Jak pokazano, istnieje wiele reprezentacji ruchu służących do osiągnięcia celu zadania planowania ruchu. Jednak identyczność punktów brzegowych trajektorii nie oznacza tożsamości między nimi. Ten aspekt doboru reprezentacji ruchu będzie przedmiotem badania w kolejnych podrozdziałach. Najpierw wprowadzono miary oceny trajektorii (podrozdział 4.2.1), a następnie je zastosowano dla dwóch układów nilpotentnych (podrozdział 4.2.2). Zawartość podrozdziału bazuje na pracy [23].

4.2.1. Miary oceny trajektorii dla zadanej reprezentacji ruchu

Do zbadania wpływu reprezentacji ruchu na wynikową trajektorię układu (2.36), wyliczoną algorytmem LS, zostaną wprowadzone miary oceny trajektorii (a więc także reprezentacji ruchu). Ponieważ naturalną trajektorią łączącą punkt początkowy z końcowym planowania jest odcinek prostej, dlatego warto zdefiniować miary opisujące wynikową trajektorię względem tej naturalnej trajektorii (toru). Warto zauważyć, że układy nieholonomiczne praktycznie nigdy nie generują trajektorii w postaci linii prostej, która jest trudna do realizacji za pomocą dopuszczalnych sterowań. Definiowanie miar jakości trajektorii poprzedzimy krokiem wstępnym. Najpierw następuje przemieszczenie współrzędnych $\hat{\mathbf{q}} = \mathbf{q} - \mathbf{q}_0$ tak, aby usytuować punkt początkowy planowania ruchu \mathbf{q}_0 w początku układu współrzędnych, a nowy punkt docelowy w punkcie $\hat{\mathbf{q}}_f = \mathbf{q}_f - \mathbf{q}_0$ (punkty trajektorii również są odpowiednio transformowane). Następnie kanoniczne wersory początkowego układu współrzędnych \mathbf{e}_k^n , $k = 1, \dots, n$ są obracane w taki sposób, aby wersor $\hat{\mathbf{e}}_1^n$ nowego układu współrzędnych był współliniowy z $\hat{\mathbf{q}}_f$ i równy $(1, 0, \dots, 0)^T$, a pozostałe wersory $\hat{\mathbf{e}}_k^n$, $k = 2, \dots, n$ są wybierane zgodnie ze zmodyfikowanym algorytmem Gramma-Schmidta [32]. Zainicjowany zbiorem wersorów $\{\hat{\mathbf{q}}_f / \|\hat{\mathbf{q}}_f\|, \mathbf{e}_2^n, \dots, \mathbf{e}_n^n\}$, algorytm Gramma-Schmidta generuje ortogonalną bazę $\hat{\mathbf{e}}_k^n$, $k = 1, \dots, n$. W bazie tej, badana trajektoria jest wyrażona jako $\hat{\mathbf{q}}(t)$, $t \in [0, T]$, a punktem docelowym jest $\hat{\mathbf{q}}_f = (\|\hat{\mathbf{q}}_f\|, 0, \dots, 0)^T$. W otrzymanym układzie współrzędnych, pierwsza zmienna naturalnie parametryzuje prostą w kierunku celu. Gdy transformacja zmiennych jest przeprowadzona w środowisku kolizyjnym, również przeszkody są odpowiednio przekształcone, by zachować ich właściwą relację z trajektorią.

Aby przy pomocy skondensowanych danych oddać kształt trajektorii $\hat{\mathbf{q}}(\cdot)$ w nowym układzie współrzędnych, zdefiniowano dwa typy miar. Miara odległościowa, opisująca

euklidesową odległość bieżącego punktu trajektorii $\hat{\mathbf{q}}(t)$ od prostej wyznaczonej punktami $\mathbf{0}$, $\hat{\mathbf{q}}_f$, może być opisana analitycznie jako

$$\text{dist}(t) = \text{dist}(\hat{\mathbf{q}}(t), \text{prosta}(\mathbf{0}, \hat{\mathbf{q}}_f)) = \sqrt{\sum_{i=2}^n \hat{q}_i^2(t)}. \quad (4.44)$$

Istnieje nieskończenie wiele punktów na $(n-1)$ wymiarowej sferze, dla których miara (4.44) przyjmuje ustaloną wartość. Dlatego powinna być także zdefiniowana pewna miara kątowa do bardziej precyzyjnego opisu $\hat{\mathbf{q}}(t)$. Ponieważ w nowym układzie współrzędnych $\hat{\mathbf{e}}_i^n$ są kanonicznymi wersorami, dlatego kąt pomiędzy $(n-1)$ wersorami a wektorem $\hat{\mathbf{q}}(t)$ (pierwszy kąt nie jest brany pod uwagę, gdyż nie niesie znaczącej informacji kątowej)

$$\text{kąt}_i(t) = \left\langle \frac{\hat{\mathbf{q}}(t)}{\|\hat{\mathbf{q}}(t)\|}, \hat{\mathbf{e}}_{i+1}^n \right\rangle \quad i = 1, \dots, n-1. \quad (4.45)$$

Definiując zredukowany wektor $\tilde{\mathbf{q}}_c$ jako wektor \mathbf{q}_c (odpowiadający $\hat{\mathbf{q}}(t)$) z opuszczoną pierwszą składową, można wykorzystać uogólnione współrzędne sferyczne $(\theta_1, \dots, \theta_{n-1})^T$ jako miarę kątową. Wektor $\tilde{\mathbf{q}}_c = (q_{c2}, \dots, q_{cn})^T$ jest zależny od współrzędnych sferycznych w następujących sposób

$$q_{c2} = \|\tilde{\mathbf{q}}_c\| \cos \theta_1, \quad q_{c(i+1)} = \|\tilde{\mathbf{q}}_c\| \cos \theta_i \prod_{j=1}^{i-1} \sin \theta_j, \quad i = 2, \dots, n-2,$$

$$q_{cn} = \|\tilde{\mathbf{q}}_c\| \prod_{j=1}^{n-1} \sin \theta_j,$$

a współrzędne sferyczne odpowiadające wektorowi $\tilde{\mathbf{q}}_c$ zadane są formułą

$$\theta_{n-1} = \text{atan2}(q_{cn}, q_{c(n-1)}), \quad \theta_i = \text{atan2}(q_{c(i+1)}, q_{ci} \cos \theta_{i+1}), \quad i = n-2, \dots, 1, \quad (4.46)$$

gdzie wykorzystano standardową funkcję numeryczną atan2 , aby uniknąć dzielenia przez zero, przy obliczaniu kolejnych kątów θ_i . W przypadku przestrzeni trójwymiarowej, $n=3$, jest tylko jeden kąt określający obie kątowe miary (4.45), (4.46), które są równoważne. Zaprezentowane miary (4.44), (4.45) opisują lokalny aspekt trajektorii, bowiem wartościują jeden (bieżący) punkt trajektorii. Bazując na przedstawionych miarach lokalnych można badać także globalne charakterystyki trajektorii. Dla miary pozycyjnej (4.44) są to

$$\text{dist}_\Sigma = \int_0^T \text{dist}(t) dt \quad (4.47)$$

całkująca miarę lokalną wzdłuż trajektorii, natomiast

$$\text{dist}_{\max} = \max_{t \in [0, T]} \text{dist}(t) \quad (4.48)$$

określa maksymalne odchylenie trajektorii $\hat{\mathbf{q}}(\cdot)$ od prostej łączącej $\mathbf{0}$ z $\hat{\mathbf{q}}_f$.

Tablica 4.4. Reprezentacje ruchu P. Halla (dla $m = 2$ i stopnia nilpotencji 2) oraz typy ich równań CFS

Lp.	reprezentacja Halla, $\mathbf{S}(t)$	typ	Lp.	reprezentacja Halla, $\mathbf{S}(t)$	typ
1	$e^{h_1 X + h_2 Y + h_3 [X, Y]}$	#1	8	$e^{h_1 X} e^{h_2 Y} e^{h_3 [X, Y]}$	#2
2	$e^{h_1 X + h_2 Y} e^{h_3 [X, Y]}$	#1	9	$e^{h_1 X} e^{h_3 [X, Y]} e^{h_2 Y}$	#2
3	$e^{h_3 [X, Y]} e^{h_1 X + h_2 Y}$	#1	10	$e^{h_2 Y} e^{h_1 X} e^{h_3 [X, Y]}$	#3
4	$e^{h_1 X + h_3 [X, Y]} e^{h_2 Y}$	#2	11	$e^{h_2 Y} e^{h_3 [X, Y]} e^{h_1 X}$	#3
5	$e^{h_2 Y} e^{h_1 X + h_3 [X, Y]}$	#3	12	$e^{h_3 [X, Y]} e^{h_1 X} e^{h_2 Y}$	#2
6	$e^{h_2 Y + h_3 [X, Y]} e^{h_1 X}$	#3	13	$e^{h_3 [X, Y]} e^{h_2 Y} e^{h_1 X}$	#3
7	$e^{h_1 X} e^{h_2 Y + h_3 [X, Y]}$	#2			

4.2.2. Wyniki symulacji

W pierwszym zadaniu symulacyjnym algorytm LS został zastosowany do obliczenia trajektorii integratora Brocketta opisanego równaniami (A.1). Zadanie planowania ruchu polega na wyznaczeniu trajektorii między punktem początkowym $\mathbf{q}_0 = (0, 0, 0)^T$ a konfiguracją docelową $\mathbf{q}_f = (1, 2, 3.2)^T$ na zadanym horyzoncie czasowym $T = 1$. Układ Brocketta jest przykładem sterowalnego układu nilpotentnego dwuwęściowego o stopniu nilpotentności równym 2, a więc $r = 3$. Istnieje 13 różnych reprezentacji ruchu, dla układu Brocketta (A.1), które zostały zebrane w tab. 4.4. Dla każdej reprezentacji zostały wyliczone równania CFS (3.8), które mogą być podzielone na trzy grupy, zgodnie z typami wymienionymi w tablicy 4.4

$$\begin{array}{ccc}
 \text{typ \#1} & \text{typ \#2} & \text{typ \#3} \\
 \left\{ \begin{array}{l} \dot{h}_1 = v_1 \\ \dot{h}_2 = v_2 \\ \dot{h}_3 = \frac{1}{2} h_1 v_2 - \frac{1}{2} h_2 v_1 + v_3 \end{array} \right. & \left\{ \begin{array}{l} \dot{h}_1 = v_1 \\ \dot{h}_2 = v_2 \\ \dot{h}_3 = -h_2 v_1 + v_3 \end{array} \right. & \left\{ \begin{array}{l} \dot{h}_1 = v_1 \\ \dot{h}_2 = v_2 \\ \dot{h}_3 = h_1 v_2 + v_3, \end{array} \right.
 \end{array}$$

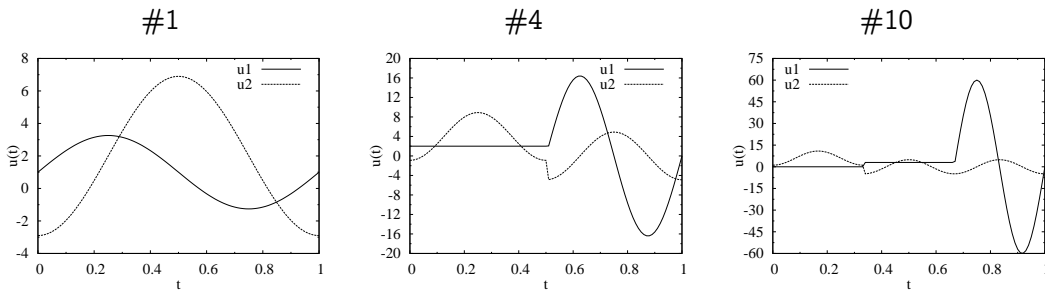
zainicjalizowane w $\mathbf{h}(0) = \mathbf{0}$.

Jako trajektoria referencyjna została obrana $\mathbf{q}(t) = \mathbf{q}_0 + (\mathbf{q}_f - \mathbf{q}_0)t/T$. W ostatnim kroku algorytmu LS, dla zadanej reprezentacji (z wyliczonymi parametrami $h_i(T)$, $i = 1, \dots, r$), wykorzystano formułę gCBHD [26] do generacji sterowań. Dla każdego z \bar{r} interwałów czasowych o długości $T' = T/\bar{r}$ zastosowano następujące sterowania $u_1(s) = p_1 + p_2 \sin(2\pi s/T')$, $u_2(s) = p_3 + p_4 \cos(2\pi s/T')$ gdzie $s \in [0, T']$ oraz $\mathbf{p} = (p_1, \dots, p_4)^T$ są parametrami do wyznaczenia. Parametry wyliczono za pomocą wersji algorytmu Newtona służącego do rozwiązania zadania odwrotnego do $\mathbf{f}(\mathbf{p}) = \mathbf{h}(T)$ ze znanym $\mathbf{h}(T)$ i funkcjami $\mathbf{f}(\mathbf{p})$ wyznaczonymi za pomocą formuły gCBHD dla zadanej postaci sterowań. Na rys. 4.2 zaprezentowano wykresy miary odległościowej (4.44) dla różnych reprezentacji. Na każdym rysunku przedstawiono rodziny charakterystyk z uzmiennionym punktem docelowym danym jako $\mathbf{q}_f(\xi) = \mathbf{q}_0 + \xi(\mathbf{q}_f - \mathbf{q}_0)$ gdzie $\xi \in (0, 1]$. W tab. 4.5 zebrano wartości globalnych miar pozycyjnych (4.47), (4.48), natomiast na

rys. 4.1 przedstawiono sygnały sterujące dla wybranych reprezentacji ruchu. Nietrudno zauważyć (por. rys. 4.2), że wybór reprezentacji ruchu istotnie wpływa na kształt trajektorii ruchu między zadanymi punktami brzegowymi. Dla wybranych warunków brzegowych reprezentacja #1 daje najmniejszą odległość od prostej $\mathbf{q}_0\mathbf{q}_f$ i jest praktycznie o rząd efektywniejsza niż reprezentacja #10. Często dla jednej reprezentacji istnieje wiele ekstremów funkcji $\text{dist}(\cdot)$, a wartość maksymalna jest osiągnięta dla różnych t (w różnych miejscach wzdłuż linii łączącej \mathbf{q}_0 z $\mathbf{q}_f(\xi)$). Z rys. 4.2 i danych z tab. 4.5 można wyciągnąć wniosek, że trajektorie (dla każdej reprezentacji) mniej oddalają się od prostej $\mathbf{q}_0\mathbf{q}_f(\xi)$, gdy punkt docelowy zbliża się do punktu początkowego. Fakt ten jest bardzo istotny dla zastosowań algorytmu LS w środowisku kolizyjnym. Jeśli bowiem pewna trajektoria (niekoniecznie realizowalna za pomocą dopuszczalnych sterowań) otrzymana za pomocą holonomicznej metody planowania ruchu (pól potencjałów, diagramu Voronoia, czy innej) jest otwarta (czyli jej najbliższe sąsiedztwo jest wolne od przeszkód), wtedy trajektoria ta może być aproksymowana (realizowana) za pomocą nieholonomicznej metody LS oraz trajektoria rzeczywista będzie bezkolizyjna.

Sterowania przedstawione na rys. 4.1 są konkatencją (w zależności od \bar{r}) jednego, dwóch lub trzech interwałów czasowych równej długości, a ich amplitudy różnią się znacząco. Reprezentacje ruchu z mniejszą liczbą segmentów wydają się być bardziej efektywne energetycznie niż te z większą liczbą. Można również wyciągnąć wniosek, że pierwsza – kanoniczna reprezentacja jest najlepsza z całej rodziny reprezentacji. Ponadto, trójsegmentowe reprezentacje znane w literaturze robotycznej jako „w przód” (#8) i „wstecz” (#13) są statystycznie mniej efektywne niż reprezentacje dwu- czy jednosegmentowe.

Rezultaty dla przykładowych zadań dostarczają wizualnej informacji o różnicach między reprezentacjami ruchu. Wnioski z pojedynczych przykładów mogą być niewiarygodne, dlatego rozpatrzono zbiór zadań, aby uzyskać statystycznie istotną informację. Punkt $(0, 0, 0)^T$ jest początkowym dla wszystkich zadań, a horyzont czasowy ustalono na $T = 1$. Punkty docelowe wybrano na sferze $(R, \alpha, \beta)^T$, o promieniu $R = 1$, z $\alpha \in \{i \cdot 10^\circ\}$, $i = 0, \dots, 35$, $\beta \in \{j \cdot 10^\circ\}$, $j = -9, \dots, 9$. Uwzględniono dwa kryteria: ener-



Rysunek 4.1. Sterowania realizujące ruch integratora Brocketta między \mathbf{q}_0 a \mathbf{q}_f dla wybranych reprezentacji

Tablica 4.5. Wartości miar pozycyjnych dla punktów docelowych (skalowanych współczynnikiem ξ) i różnych reprezentacji ruchu

nr rep.	Miara pozycyjna (4.47)					Miara pozycyjna (4.48)				
	ξ					ξ				
	0.2	0.4	0.6	0.8	1.0	0.2	0.4	0.6	0.8	1.0
1	0.40	0.40	0.41	0.43	0.45	0.65	0.65	0.68	0.73	0.82
2	0.35	0.68	1.14	1.75	2.52	0.64	1.65	3.12	5.14	7.80
3	0.30	0.45	0.56	0.64	0.69	0.56	0.90	1.15	1.30	1.33
4	0.28	0.38	0.47	0.54	0.61	0.50	0.75	0.92	1.03	1.12
5	0.26	0.37	0.51	0.69	0.91	0.40	0.63	0.87	1.25	1.76
6	0.37	0.68	1.05	1.49	1.99	0.68	1.56	2.64	3.94	5.46
7	0.31	0.53	0.76	1.01	1.26	0.59	0.97	1.42	1.88	2.35
8	0.32	0.55	0.74	0.88	0.95	0.84	1.64	2.31	2.77	2.90
9	0.30	0.48	0.64	0.76	0.86	0.82	1.45	1.91	2.19	2.90
10	0.41	0.90	1.53	2.34	3.34	1.08	2.73	5.09	8.32	12.53
11	0.31	0.63	1.08	1.68	2.45	0.86	2.24	4.26	7.06	10.78
12	0.30	0.48	0.60	0.66	0.67	0.64	1.07	1.34	1.43	1.34
13	0.33	0.67	1.10	1.61	2.20	0.81	1.78	2.93	4.26	5.76

gię sterowań energia($\mathbf{u}(\cdot)$) = $\int_{t=0}^T (u_1^2 + u_2^2) dt$ oraz maksymalną odległość d_{\max} daną równaniem (4.48). Dla wszystkich trzynastu reprezentacji wyliczono wartość średnią ($\bar{x} = \sum_{i=1}^s x_i/s$) oraz odchylenie standardowe (odch std = $\sqrt{\sum_{i=1}^s (x_i - \bar{x})^2/(s-1)}$) dla wyznaczonych wartości kryteriów $\mathbf{x} = (x_1, \dots, x_s)^T$ ze wszystkich s rozwiązanych zadań planowania, a wyniki zebrano w tab. 4.6. Okazuje się, że obserwacje poczynione dla pojedynczego zadania są również ważne dla zbioru zadań. Ponadto relatywnie większe wartości odchylenia standardowego energii wskazują, że istnieją energetycznie tanie i drogie kierunki ruchu (por. [27, 91]).

Drugi przykład dotyczy czterowymiarowego, sterowalnego, nilpotentnego układu łańcuchowego z dwoma sterowaniami o rzędzie nilpotentności równym 3, którego model (A.2) i własności zabrano w dodatku A. W tym przypadku istnieje bardzo wiele reprezentacji (por. tab. 4.1). Do obliczeń wybieramy jedynie trzy reprezentacje: kanoniczną, „w przód” i „wstecz”. Równania CFS dla tych reprezentacji są następujące: kanoniczna

$$\begin{aligned}
(\dot{h}_1, \dot{h}_2, \dot{h}_3, \dot{h}_4, \dot{h}_5)^T &= (v_1, v_2, \frac{1}{2}(h_1 v_2 - h_2 v_1) + v_3, \\
v_4 + \frac{1}{12} h_1 (h_1 v_2 - h_2 v_1) + \frac{1}{2} (h_1 v_3 - h_3 v_1), v_5 + \frac{1}{12} h_2 (h_1 v_2 - h_2 v_1) + \frac{1}{2} (h_2 v_3 - h_3 v_2))^T,
\end{aligned}
\tag{4.49}$$

Tablica 4.6. Statystyczna ocena reprezentacji ze względu na wydatki energetyczne oraz maksymalną odległość od prostej łączącej \mathbf{q}_0 z \mathbf{q}_f dla układu Brocketta

nr rep.	całkowita energia		maks. odległość	
	średnia	odch. std.	średnia	odch. std.
1	4.96	1.71	0.50	0.12
2	11.46	2.20	0.67	0.13
3	11.44	2.19	0.62	0.15
4	13.04	2.66	0.71	0.16
5	13.05	2.66	0.65	0.12
6	10.25	3.29	0.61	0.10
7	10.25	3.29	0.60	0.11
8	19.85	4.29	0.73	0.14
9	19.85	4.29	0.76	0.14
10	19.85	4.29	0.72	0.14
11	19.85	4.29	0.74	0.11
12	19.85	4.29	0.69	0.13
13	19.85	4.29	0.68	0.12

„w przód”

$$(\dot{h}_1, \dot{h}_2, \dot{h}_3, \dot{h}_4, \dot{h}_5)^T = (v_1, v_2, v_3 - h_2 v_1, v_4 - h_3 v_1, v_5 - h_3 v_2 + \frac{1}{2} h_2^2 v_1)^T, \quad (4.50)$$

„wstecz”

$$(\dot{h}_1, \dot{h}_2, \dot{h}_3, \dot{h}_4, \dot{h}_5)^T = (v_1, v_2, v_3 + h_1 v_2, v_4 + h_1 v_3 + \frac{h_1^2}{2} v_2, v_5 + h_2 v_3 + h_1 h_2 v_2)^T. \quad (4.51)$$

Warto zauważyć, że choć układ łańcuchowy (A.2) jest czterowymiarowy, $n = 4$, odpowiadające mu równania CFS są pięciowymiarowe, bowiem równania CFS muszą pokryć pełne warstwy pól wektorowych. Każde pole wektorowe pojawiające się w danej warstwie posiada porównywalny wpływ na wynikowy ruch, a zatem żadnego z nich pominąć nie można. Dla układu (A.2) współczynnik h_5 nie wpływa na ruch układu łańcuchowego, ponieważ jest pomnożony przez zerowe pole wektorowe $[\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]]$.

Do generacji $\mathbf{h}(T)$, dla każdego segmentu $i = 1, \dots, \bar{r}$ sterowania wybrano z rodziny $u_1(s) = p_1 + p_2 \sin(\omega s) + p_3 \sin(2\omega s)$, $u_2(s) = p_4 + p_5 \cos(\omega s) + p_6 \cos(2\omega s)$ gdzie $s \in [0, T/\bar{r} = T']$, $\omega = 2\pi/T'$. Podobnie jak dla poprzedniego przykładu wartości parametrów $\mathbf{p} = (p_1, \dots, p_6)^T$ są wyliczone za pomocą algorytmu Newtona, chociaż funkcje $\mathbf{f}(\mathbf{p})$ są zupełnie inne w tym przypadku. Symulacje zostały przeprowadzone dla punktu docelowego położonego na czterowymiarowej sferze o promieniu $R = 1$ (zostało wygenerowanych ponad 300 zadań). Punktem początkowym dla wszystkich zadań jest $\mathbf{q}_0 = (0, 0, 0, 0)^T$, a horyzont czasowy $T = 1$. Wyniki symulacji zebrano w tab. 4.7. Okazuje się, że obserwacje poczynione dla trójwymiarowego układu Brocketta pozostają ważne także dla czterowymiarowego przypadku.

Tablica 4.7. Statystyczna ocena trzech reprezentacji ze względu na wydatki energetyczne oraz maksymalną odległość od prostej $\mathbf{q}_0\mathbf{q}_f$ dla czterowymiarowego układu łańcuchowego

reprezentacja	energia całkowita		maks. odległość	
	średnia	odch. std.	średnia	odch. std.
„wstecz”	924.54	359.57	2.99	0.35
„w przód”	965.49	364.77	3.26	0.42
kanoniczna	68.49	63.29	2.17	0.96

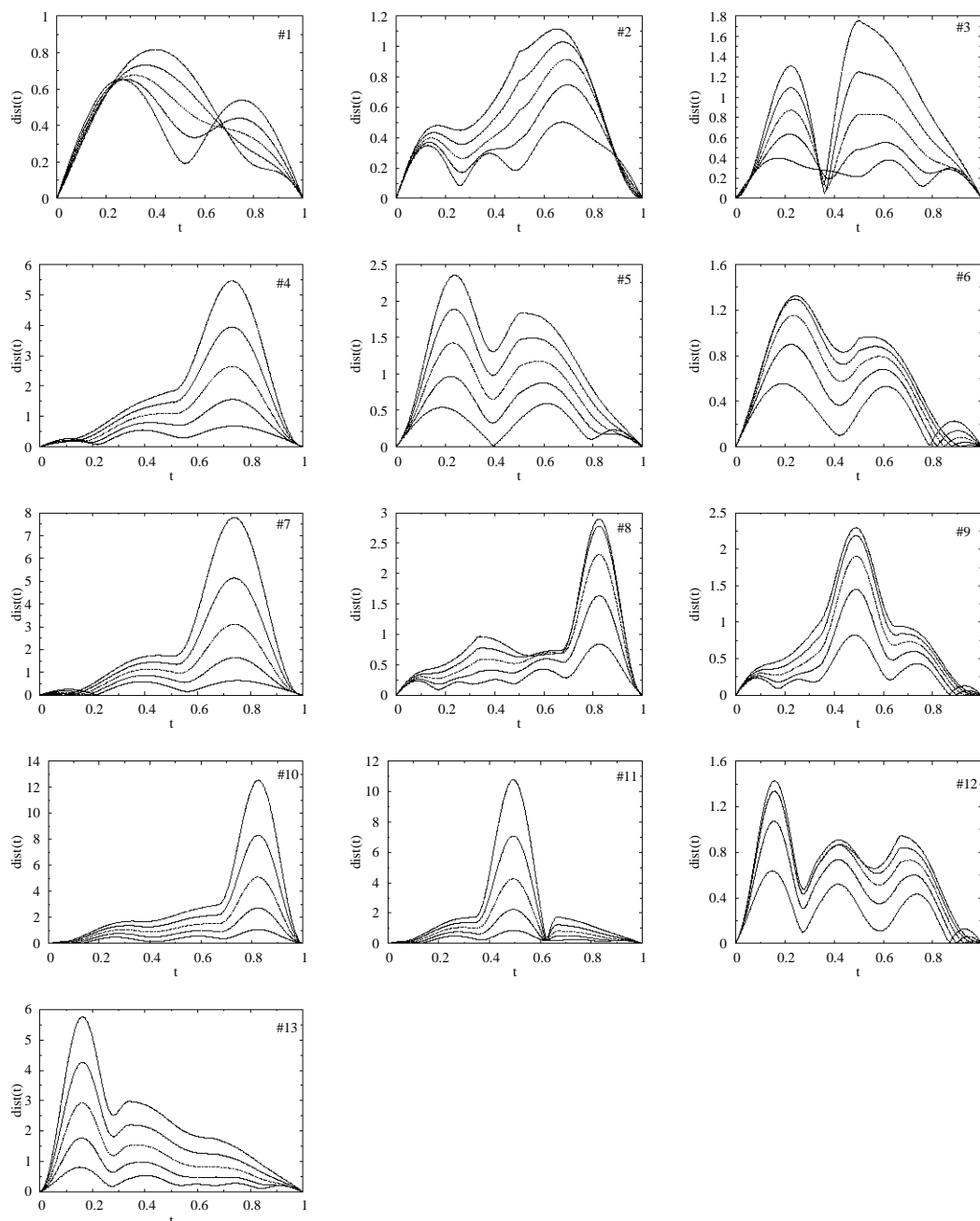
W prezentacji i porównaniu reprezentacji ruchu warto uwzględnić także ich złożoność obliczeniową. W wielu przypadkach złożoność obliczeniowa nie jest krytycznym problemem w planowaniu ruchu, ponieważ obliczenia mogą być przeprowadzone w trybie off-line (pre-planowanie), a wynikowa trajektoria lub ścieżka powinna być dostarczona do modułu sterującego (pracującego w trybie czasu rzeczywistego), który odwzorowuje ją najlepiej jak to możliwe. Złożoność obliczeniowa przy wyborze reprezentacji nie jest istotna, gdyż dla zadanej reprezentacji równania różniczkowe CFS mogą być wyliczone w trybie off-line, gdyż zależą od bardzo ogólnych własności nilpotentnego układu bezdryfowego (m, n, r) , natomiast nie zależą od konfiguracji brzegowych planowania. Wyliczenie $\mathbf{h}(T)$ jest mniej więcej tak samo czasochłonne dla każdej reprezentacji, a jest to czas potrzebny do całkowania równań CFS, odpowiadającym danej reprezentacji i wyliczonym $\mathbf{v}(\cdot)$. Jedyną własność różnicującą reprezentacje ruchu pod względem złożoności obliczeniowej polega na zmiennej liczbie segmentów \bar{r} . Zwykle im liczba segmentów jest większa, tym bardziej złożone jest zadanie planowania, gdyż dla wyliczenia sterowań algorytm Newtona musi być stosowany \bar{r} krotnie. W tym aspekcie kanoniczna reprezentacja wydaje się najmniej złożoną obliczeniowo, ($\bar{r} = 1$). Jednak w reprezentacji ruchu „w przód”, i „wstecz”, o największej liczbie segmentów $\bar{r} = r \geq n$ pojawiają się elementy takie jak $e^{h_i X}$, $e^{h_j Y}$, które mogą być wyliczone bez stosowania algorytmu Newtona, gdyż ich generacja wymaga włączenia lub wyłączenia odpowiednich sterowań. Dla kanonicznej reprezentacji ruchu takie łatwe do generacji segmenty nie pojawiają się.

Innym ciekawym pytaniem jest, która z reprezentacji jest najlepsza wśród możliwych do wygenerowania? Na podstawie przedstawionych symulacji, wydaje się, że kanoniczna reprezentacja jest jedną z najlepszych dla planowania ruchu w środowisku bezkolizyjnym. Jednakże, z ważnej dla planowania ruchu perspektywy unikania przeszkód, reprezentacje ruchu są trudniejsze do porównania. Trajektorie łączące dwa ustalone punkty brzegowe i odpowiadające różnym reprezentacjom ruchu przechodzą przez różne obszary przestrzeni stanu (por. rys. 4.2). W tym sensie każda reprezentacja może być najlepsza wśród rozważanych w zależności od rozmieszczenia przeszkód w jej otoczeniu. W praktyce jedynym sposobem oceny, czy odpowiadająca danej reprezentacji ruchu trajektoria jest kolizyjna z przeszkodami czy nie, jest wygenerowanie trajek-

torii i sprawdzenie jej odległości od najbliższej przeszkody. Nie jest to procedura zbyt kosztowna obliczeniowo, gdyż wymaga jedynie całkowania numerycznego równań CFS.

4.2.3. Podsumowanie

W podrozdziale przedstawiono działanie algorytmu LS dla różnych reprezentacji. Reprezentacje zostały przedstawione i przetestowane na dwóch bezdryfowych układach nilpotentnych: układzie Brocketta i czterowymiarowym układzie łańcuchowym. Zaproponowano miary odległościowe i kątowe do oceny trajektorii powstałej w wyniku działania algorytmu LS, w których istotną rolę odgrywa położenie trajektorii względem prostej łączącej punkty brzegowe planowania. Pokazano symulacyjnie, że reprezentacja ma istotny wpływ na ewolucję trajektorii łączącej punkty brzegowe. Przez zmianę reprezentacji ruchu możliwa jest optymalizacja kształtu trajektorii, której celem jest omijanie przeszkód w środowisku kolizyjnym.



Rysunek 4.2. Odległość $q(t)$ od prostej q_0q_f dla rodziny punktów docelowych skalowanych współczynnikiem $\xi = 0.2 \cdot i$, $i = 1, \dots, 5$ dla różnych reprezentacji. W prawie wszystkich przypadkach krzywe odpowiadające mniejszemu i leżą poniżej krzywych z większym i

5. Metoda sterowań sinusoidalnych jako szczególna postać algorytmu LS

Do planowania ruchu szczególnej postaci układów nilpotentnych, zwanych w literaturze robotycznej łańcuchowymi służy, zaproponowana przez Murraya i Sastry'ego, metoda sterowań sinusoidalnych [71]. W pracy [70] opisano warunki istnienia transformacji bezdryfowych dwuwęściowych układów nieholonomicznych do postaci łańcuchowej, stosując sprzężenie zwrotne zależne od stanu. Ponieważ każdy układ łańcuchowy jest nilpotentny, zatem można wykorzystać algorytm LS do planowania jego ruchu. Rozważania przeprowadzono dla dwuwęściowych, $m = 2$, układów jednołańcuchowych. Dla uproszczenia notacji przyjmujemy, że horyzont czasowy sterowań wynosi $T = 1$, a wykorzystywaną bazą jest baza Halla. Zbadano jakie własności układów łańcuchowych można wykorzystać do uproszczenia algorytmu LS, a zawartość rozdziału bazuje na pracy autora [45].

Układy jednołańcuchowe są zadane równaniem

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \vdots \\ \dot{q}_n \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ q_2 \\ q_3 \\ \vdots \\ q_{n-1} \end{bmatrix} \mathbf{u}_1 + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \mathbf{u}_2 = \mathbf{g}_1(\mathbf{q})\mathbf{u}_1 + \mathbf{g}_2(\mathbf{q})\mathbf{u}_2. \quad (5.1)$$

Metoda sterowań sinusoidalnych Murraya-Sastriego (MS) zakłada podział zmiennych \mathbf{q} na grupy. Do pierwszej grupy należą zmienne q_1, q_2 , a następne grupy tworzą, po jednej, kolejne zmienne. Ideą metody jest doprowadzanie kolejnych grup współrzędnych (poprzez kreślenie pętli w przestrzeni sterowań) do ich docelowych wartości, nie zmieniając już osiągniętych wartości docelowych współrzędnych odpowiadających grupom poprzednim. Najpierw, dowolnym sposobem (najczęściej sterowaniami stałymi), zmienne q_1, q_2 są doprowadzane do ich docelowych wartości. Dalej założymy, że ten etap już wykonano i zadaniem planowania ruchu jest przemieszczenie stanu z konfiguracji $(0, 0, q_3(0), q_4(0), \dots, q_n(0))^T$ do konfiguracji $(0, 0, q_3(1), q_4(1), \dots, q_n(1))^T$. Do sprowadzenia kolejnych zmiennych do ich końcowych wartości są stosowane sterowania

$$\mathbf{u}_1(t) = p_1 \sin(2\pi t), \quad \mathbf{u}_2(t) = p_2 \cos(2k\pi t), \quad t \in [0, 1], \quad (5.2)$$

gdzie parametry sterowań p_1, p_2 spełniają warunek

$$q_{k+2}(1) - q_{k+2}(0) = \left(\frac{p_1}{4\pi}\right)^k \frac{p_2}{k!}, \quad k \in \{1, \dots, n-2\}. \quad (5.3)$$

Zastosujemy metodę LS do planowania ruchu układu (5.1). Łatwo sprawdzić obliczeniami, z wykorzystaniem definicji nawiasu Liego (2.24), [70], że jedynymi niezerowymi nawiasami Liego dla $0 \leq k \leq n-2$ generowane przez $\mathbf{g}_1, \mathbf{g}_2$ są

$$\text{ad}_{\mathbf{g}_1}^k \mathbf{g}_2 = (0, 0, \dots, 0, (-1)^{k+1}, 0, \dots, 0)^T, \quad (5.4)$$

gdzie ± 1 zajmuje $(k+2)$ -gą pozycję. Korzystając z (5.4) wnioskujemy, że układ łańcuchowy (5.1) jest nilpotentny stopnia $n-1$, gdyż $\text{ad}_{\mathbf{g}_1}^{n-2} \mathbf{g}_2 = (0, 0, \dots, (-1)^n)^T$, a $\text{ad}_{\mathbf{g}_1}^{n-1} \mathbf{g}_2 = (0, \dots, 0)^T$, oraz pole $\text{ad}_{\mathbf{g}_1}^{n-2} \mathbf{g}_2$ należy do warstwy $(n-1)$ -szej. Po wyliczeniu nawiasów Liego, kolejny krok algorytmu LS wymaga utworzenia układu rozszerzonego

$$\dot{\mathbf{q}} = \mathbf{g}_1(\mathbf{q})v_1 + \mathbf{g}_2(\mathbf{q})v_2 + \dots + \mathbf{g}_n(\mathbf{q})v_n = \mathbf{G}_{\text{ext}}(\mathbf{q})\mathbf{v} \quad (5.5)$$

z kwadratową macierzą $\mathbf{G}_{\text{ext}}(\mathbf{q})$ o kolumnach zawierających kolejne niezerowe nawiasy Liego

$$\mathbf{G}_{\text{ext}}(\mathbf{q}) = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ q_2 & 0 & -1 & 0 & \dots & 0 \\ q_3 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ q_{n-1} & 0 & 0 & 0 & \dots & (-1)^n \end{bmatrix}.$$

Macierz $\mathbf{G}_{\text{ext}}(\mathbf{q})$ jest nieosobliwa w każdym punkcie przestrzeni konfiguracyjnej, a więc dokładnie n pól wystarczy do rozpięcia całej przestrzeni stanu. Macierz odwrotna do $\mathbf{G}_{\text{ext}}(\mathbf{q})$ jest postaci

$$\mathbf{G}_{\text{ext}}^{-1}(\mathbf{q}) = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ q_2 & 0 & -1 & 0 & \dots & 0 \\ -q_3 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ (-1)^{n-1}q_{n-1} & 0 & 0 & 0 & \dots & (-1)^n \end{bmatrix}. \quad (5.6)$$

Mnożąc lewostronnie równanie (5.5) przez (5.6) otrzymujemy sterowania rozszerzone w postaci

$$\mathbf{v} = \mathbf{G}_{\text{ext}}^{-1}(\boldsymbol{\lambda})\dot{\boldsymbol{\lambda}} = \begin{bmatrix} \dot{\lambda}_1 \\ \dot{\lambda}_2 \\ \lambda_2 \dot{\lambda}_1 - \dot{\lambda}_3 \\ -q_3 \dot{\lambda}_1 - \dot{\lambda}_4 \\ \vdots \\ (-1)^{n-1} \lambda_{n-1} \dot{\lambda}_1 + (-1)^n \dot{\lambda}_n \end{bmatrix}. \quad (5.7)$$

gdzie λ i $\dot{\lambda}$ oznacza, odpowiednio, trajektorie referencyjną i jej pochodną (por. rozdział 3.1).

Następnym etapem algorytmu LS jest wyliczenie współczynników Halla. W tym celu rozważmy równanie formalne związane z układem (5.5)

$$\dot{\mathbf{S}}(t) = \mathbf{S}(t)(v_1 \mathbf{B}_1 + v_2 \mathbf{B}_2 + \dots + v_n \mathbf{B}_n), \quad (5.8)$$

gdzie symbole formalne \mathbf{B}_i są odpowiednikami pól wektorowych \mathbf{g}_i występujących we wzorze (5.5), $\mathbf{g}_1 \leftrightarrow \mathbf{B}_1$ oraz $\mathbf{g}_i \leftrightarrow \mathbf{B}_i = \text{ad}_{\mathbf{B}_1}^{i-2} \mathbf{B}_2$ dla $n \geq i \geq 2$. Jako $\mathbf{S}(t)$ wybieramy reprezentację „w przód”

$$\mathbf{S}(t) = e^{h_1(t)\mathbf{B}_1} e^{h_2(t)\mathbf{B}_2} \dots e^{h_n(t)\mathbf{B}_n}, \quad \text{z warunkiem początkowym } \mathbf{h}(0) = \mathbf{0}. \quad (5.9)$$

Po wyliczeniu $\dot{\mathbf{S}}(t)$ i \mathbf{S}^{-1} oraz, dla uproszczenia notacji, chwilowym pominięciu zależności \mathbf{h} od czasu otrzymujemy

$$\begin{aligned} \mathbf{S}^{-1}(t)\dot{\mathbf{S}}(t) &= e^{-h_n \mathbf{B}_n} \dots e^{-h_2 \mathbf{B}_2} e^{-h_1 \mathbf{B}_1} \dot{h}_1 \mathbf{B}_1 e^{h_1 \mathbf{B}_1} e^{h_2 \mathbf{B}_2} \dots e^{h_n \mathbf{B}_n} \\ &+ e^{-h_n \mathbf{B}_n} \dots e^{-h_2 \mathbf{B}_2} \dot{h}_2 \mathbf{B}_2 e^{h_2 \mathbf{B}_2} \dots e^{h_n \mathbf{B}_n} + \dots + e^{-h_n \mathbf{B}_n} \dot{h}_n \mathbf{B}_n e^{h_n \mathbf{B}_n}. \end{aligned} \quad (5.10)$$

Podstawiając rozwinięcie $e^{\pm h_k \mathbf{B}_k}$ w szereg Taylora (2.27) do (5.10) powstaje wyrażenie pozwalające na obliczenie współczynników Halla \mathbf{h} . Jednak obliczenia mogą być uciążliwe (szczególnie ręczne), więc postaramy się je znacznie uprościć. Spójrzmy na ostatnie wyrażenie równania (5.10). Po zastosowaniu (4.5) iloczyn upraszcza się do $\dot{h}_n \mathbf{B}_n$ ze względu na nilpotentność układu oznaczającą tożsamościowe zerowanie pól stopnia większego od $n - 1$. Podobnie upraszczamy wszystkie wiersze od $\lfloor n/2 \rfloor$ do n . Jedną z własności układu łańcuchowego stanowi, że dowolny nawias Liego zawierający przynajmniej dwa pola (odpowiedniki) \mathbf{B}_2 (np. $[\mathbf{B}_1, [\mathbf{B}_1, [\mathbf{B}_2, [\mathbf{B}_1, \mathbf{B}_2]]]]$) jest równy zero. Zauważmy, iż jednomiany wchodzące w skład takiego pola również w sumie muszą dać $\mathbf{0}$. Zatem jedyne niezerowe jednomiany składają się z jednego \mathbf{B}_2 i dowolnej liczby \mathbf{B}_1 . Tylko pierwszy wiersz równania (5.10) posiada taki zestaw $\mathbf{B}_1, \mathbf{B}_2$, natomiast pozostałe upraszczają się do następującego wzoru

$$\mathbf{S}^{-1}(t)\dot{\mathbf{S}}(t) = e^{-h_n \mathbf{B}_n} \dots e^{-h_1 \mathbf{B}_1} \dot{h}_1 \mathbf{B}_1 e^{h_1 \mathbf{B}_1} \dots e^{h_n \mathbf{B}_n} + \dot{h}_2 \mathbf{B}_2 + \dots + \dot{h}_{n-1} \mathbf{B}_{n-1} + \dot{h}_n \mathbf{B}_n. \quad (5.11)$$

Po uwzględnieniu (2.27), wyrażenie przekształca się do zwartej postaci

$$\mathbf{S}^{-1}(t)\dot{\mathbf{S}}(t) = \prod_{j=n}^2 \left(\sum_{i=0}^n (-1)^i \frac{h_j^i \mathbf{B}_j^i}{i!} \right) \dot{h}_1 \mathbf{B}_1 \prod_{j=2}^n \left(\sum_{i=0}^n \frac{h_j^i \mathbf{B}_j^i}{i!} \right) + \sum_{i=2}^n \dot{h}_i \mathbf{B}_i. \quad (5.12)$$

Ponieważ żaden jednomian nie może zawierać więcej niż jednego \mathbf{B}_2 , a więc $\mathbf{B}_2^k = \mathbf{0}$ dla $k \geq 2$, ale też $\mathbf{B}_s^k = \mathbf{0}$, $s, k \geq 2$, bowiem \mathbf{B}_s , $s > 2$ zawiera przynajmniej jedno wystąpienie \mathbf{B}_2 . Stąd równanie (5.12) upraszczamy

$$\mathbf{S}^{-1}(t)\dot{\mathbf{S}}(t) = \prod_{i=n}^2 (\mathbf{I} - h_i \mathbf{B}_i) \left(\sum_{i=0}^n (-1)^i \frac{h_1^i \mathbf{B}_1^i}{i!} \right) \dot{h}_1 \mathbf{B}_1 \left(\sum_{i=0}^n \frac{h_1^i \mathbf{B}_1^i}{i!} \right) \prod_{i=2}^n (\mathbf{I} + h_i \mathbf{B}_i) + \sum_{i=2}^n \dot{h}_i \mathbf{B}_i. \quad (5.13)$$

Zauważmy, że wewnętrzny iloczyn $\left(\sum_{i=0}^n (-1)^i \frac{h_i^i \mathbf{B}_i^i}{i!}\right) \dot{h}_1 \mathbf{B}_1 \left(\sum_{i=0}^n \frac{h_i^i \mathbf{B}_i^i}{i!}\right)$ jest zależny jedynie od \mathbf{B}_1 , zatem jest przemienny, więc na mocy (4.5) wynosi $\dot{h}_1 \mathbf{B}_1$. Stąd

$$\mathbf{S}^{-1}(t) \dot{\mathbf{S}}(t) = \prod_{i=n}^2 (\mathbf{I} - h_i \mathbf{B}_i) \dot{h}_1 \mathbf{B}_1 \prod_{i=2}^n (\mathbf{I} + h_i \mathbf{B}_i) + \sum_{i=2}^n \dot{h}_i \mathbf{B}_i. \quad (5.14)$$

W zależności (5.14) iloczyny typu $h_i \mathbf{B}_i \dot{h}_1 \mathbf{B}_1 h_j \mathbf{B}_j$, $i, j \neq 1$ zerują się, gdyż wystąpią w nich przynajmniej dwa elementy \mathbf{B}_2 , a pozostaną $\dot{h}_1 \mathbf{B}_1 \mathbf{I}$, $-h_i \mathbf{B}_i \dot{h}_1 \mathbf{B}_1 \mathbf{I}$, $\dot{h}_1 \mathbf{B}_1 h_i \mathbf{B}_i$, $i \geq 2$, a więc

$$\mathbf{S}^{-1}(t) \dot{\mathbf{S}}(t) = \sum_{i=2}^n \left(\dot{h}_1 \mathbf{B}_1 h_i \mathbf{B}_i - h_i \mathbf{B}_i \dot{h}_1 \mathbf{B}_1 \right) + \sum_{i=1}^n \dot{h}_i \mathbf{B}_i. \quad (5.15)$$

Korzystając z definicji nawiasu Liego, dla układu łańcuchowego zachodzi $\mathbf{B}_1 \mathbf{B}_i - \mathbf{B}_i \mathbf{B}_1 = [\mathbf{B}_1, \mathbf{B}_i] = \mathbf{B}_{i+1}$, a więc ostatecznie

$$\mathbf{S}^{-1}(t) \dot{\mathbf{S}}(t) = \sum_{i=3}^n \left(\dot{h}_1 h_{i-1} \mathbf{B}_i \right) + \sum_{i=1}^n \dot{h}_i \mathbf{B}_i = \dot{h}_1 \mathbf{B}_1 + \dot{h}_2 \mathbf{B}_2 + \sum_{i=3}^n (\dot{h}_1 h_{i-1} + \dot{h}_i) \mathbf{B}_i. \quad (5.16)$$

Przyrównując (5.16) do równania (5.8) otrzymujemy następujące równanie różniczkowe na współczynniki Halla dla układu łańcuchowego (5.1) z reprezentacją „w przód” zadaną przez (5.9)

$$\dot{h}_1 = v_1, \quad \dot{h}_2 = v_2, \quad \dot{h}_3 = v_3 - h_2 v_1, \quad \dot{h}_4 = v_4 - h_3 v_1, \quad \dots \quad \dot{h}_n = v_n - h_{n-1} v_1, \quad (5.17)$$

z warunkami początkowymi $\mathbf{h}(0) = \mathbf{0}$. Podstawiając (5.7) do równań (5.17) i uwzględniając warunki początkowe wyznaczamy wartości współczynników Halla w chwili końcowej $T = 1$

$$h_1(1) = 0, \quad h_2(1) = 0, \quad h_k(1) = (-1)^k (q_3(1) - q_3(0)), \quad \text{dla } k \geq 3. \quad (5.18)$$

Ostatnim krokiem algorytmu LS dla układu łańcuchowego, jest wygenerowanie $\mathbf{h}(1)$ za pomocą rzeczywistych sterowań $\mathbf{u}(\cdot)$ (a nie wirtualnych $\mathbf{v}(\cdot)$). Dla uproszczenia obliczeń w dalszych rozważaniach ograniczymy rozmiar wektora \mathbf{q} do 6. Podstawiamy więc sterowania metody MS zadane przez (5.2), dla kolejnych wartości k , do formuły gCBHD, tab. 3.1, i porównujemy wyliczone współczynniki Halla $\beta(1)$ z $\mathbf{h}(1)$ dla identycznych pól uzyskując

$$\begin{cases} h_3(1) &= \beta_{[\mathbf{g}_1, \mathbf{g}_2]}(1) = -\left(\frac{p_1}{4\pi}\right)^1 \frac{p_2}{1!} \\ h_4(1) &= \beta_{[\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]]}(1) = \left(\frac{p_1}{4\pi}\right)^2 \frac{p_2}{2!} \\ h_5(1) &= \beta_{[\mathbf{g}_1, [\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]]]}(1) = -\left(\frac{p_1}{4\pi}\right)^3 \frac{p_2}{3!} \\ h_6(1) &= \beta_{[\mathbf{g}_1, [\mathbf{g}_1, [\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]]]]}(1) = \left(\frac{p_1}{4\pi}\right)^4 \frac{p_2}{4!}. \end{cases} \quad (5.19)$$

Zauważmy, że zgodnie z podanymi własnościami układów łańcuchowych, pola $[\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]]$, $[\mathbf{g}_2, [\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]]]$, $[\mathbf{g}_2, [\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]]]$ zerują się tożsamościowo, więc ich współczynników nie wyliczamy. W wyniku przeprowadzenia algorytmu LS dla układu łańcuchowego (5.1), przy zadanej klasie sterowań (5.2) i reprezentacji ruchu $\mathbf{S}(t)$ „w przód” zadanej przez (5.9), uzyskano identyczną postać parametrów sterowań (5.19), (5.18) jak w przypadku literaturowego algorytmu MS (5.3).

Podsumowanie

Choć metoda LS jest ogólniejsza, gdyż działa dla dowolnych układów nilpotentnych, a metoda MS jedynie dla układów nilpotentnych o odpowiedniej strukturze, to dla tej właśnie podklasy możliwe jest porównanie obu metod. Układ (jedno) łańcuchowy jest szczególnym przypadkiem układu nilpotentnego, który charakteryzuje się istnieniem (oprócz warstwy pierwszej) dokładnie jednego niezerowego pola wektorowego postaci $\alpha d_{\mathbf{g}_1}^k \mathbf{g}_2$ w każdej warstwie. Dzięki tej strukturze, możliwe jest znaczne uproszczenie obliczeń w metodzie LS, a uzyskane wyniki (współczynniki Halla, równanie CFS) są w pełni analityczne. Pokazano równoważność algorytmów LS i MS na przykładzie układu łańcuchowego o 6-iu zmiennych stanu. Problem z rozszerzeniem tego wyniku do wektora stanu o dowolnej długości zaszyty jest w procedurze wyliczania równań (tab. 3.1) z formuły gCBHD. Równania te uzyskuje się poprzez kombinatoryczne przekształcenia, natomiast nie jest znany wzór na jawną postać formuły gCBHD. Problem ten został opisany przez Kawskiego w [47]. Gdyby pojawiło się jego rozwiązanie, zaprezentowany wynik można byłoby rozszerzyć na przestrzenie konfiguracyjne wyższych wymiarowości.

Zauważmy jeszcze jedną ważną prawidłowość: współczynniki Halla $\mathbf{h}(\cdot)$ zwartościowane dla czasu końcowego $t = T$ są liczbami (zależnymi jedynie od wartości brzegowych konfiguracji), które opisują, zgodnie z przyjętą reprezentacją ruchu (5.9) czasy ruchu wzdłuż poszczególnych pól wektorowych do osiągnięcia zadanego stanu docelowego. Zatem równania (5.18) w chwili czasu T należy interpretować następująco $h_k(T) = (-1)^k (\lambda_k(T) - \lambda_k(0))$ dla $k \geq 3$, skąd wnioskujemy, że trajektoria referencyjna $\lambda(\cdot)$ nie wpływa na wartość współczynników Halla dla $t = T$, a istotny wpływ na te wartości mają jedynie konfiguracja początkowa i końcowa.

Dla typowych układów bezdryfowych barierą w wykorzystaniu algorytmu LS jest brak ich nilpotentności. Dlatego warto rozważyć możliwość transformacji układu do postaci łańcuchowej, który oprócz nilpotentności posiada własności znacznie ułatwiające wykorzystanie algorytmu LS.

6. Trajektoria referencyjna

Algorytm Lafferriera–Sussmanna jest metodą planowania ruchu do punktu docelowego, zatem nie odtwarza trajektorii zadanej. Jednak w początkowej fazie algorytmu jest wybierana różniczkowalna i łatwo parametryzowalna trajektoria referencyjna $\boldsymbol{\lambda}(\cdot)$ łącząca punkt startowy $\boldsymbol{\lambda}(0) = \mathbf{q}_0$ z końcowym $\boldsymbol{\lambda}(T) = \mathbf{q}_f$ w celu uzyskania sterowań rozszerzonych. W publikacjach dotyczących metody LS [55, 60] badacze przyjmowali odcinek jako trajektorię referencyjną $\boldsymbol{\lambda}(t)$. Wybór ten był podyktowany łatwą parametryzacją odcinka oraz faktem, że (w przestrzeniach jednorodnych) odcinek jest najkrótszą krzywą łączącą punkty. W niniejszym rozdziale zostanie rozważony wpływ tej krzywej na działanie algorytmu LS.

Łącząc równanie CFS (3.8) z równaniem na sterowania rozszerzone (3.3)

$$\dot{\mathbf{h}}(t) = \mathbf{M}(\mathbf{h}(t)) \left(\mathbf{G}_{\text{ext}}^{\#}(\boldsymbol{\lambda}(t)) \dot{\boldsymbol{\lambda}}(t) \right), \quad \mathbf{h}(0) = \mathbf{0}, \quad (6.1)$$

łatwo zauważyć, że trajektoria referencyjna $\boldsymbol{\lambda}(\cdot)$ ma wpływ na współczynniki (np. Hal-la) \mathbf{h} , a zatem także na generowane sterowania. Naturalnym pomysłem jest wybór trajektorii referencyjnej w klasie wielomianów (prosta, czy parabola) i dzięki temu równanie różniczkowe CFS, jakkolwiek skomplikowane, będzie łatwiejsze do rozwiązania. Przetestujemy trzy typy trajektorii referencyjnych: odcinek, wielomian dowolnego stopnia oraz odcinek z nałożonym zaburzeniem sinusoidalnym. Trajektoria referencyjna będąca odcinkiem zadana jest wzorem

$$\boldsymbol{\lambda}(t) = \frac{\mathbf{q}_f - \mathbf{q}_0}{T} t + \mathbf{q}_0, \quad t \in [0, T]. \quad (6.2)$$

Drugą trajektorią referencyjną jest rodzina wielomianów łączących punkty brzegowe

$$\boldsymbol{\lambda}(t) = \mathbf{a}_k t^k + \mathbf{a}_{k-1} t^{k-1} + \dots + \mathbf{a}_2 t^2 + \bar{\mathbf{a}}_1 t + \mathbf{q}_0, \quad t \in [0, T], \quad k \in \mathbb{N}, \quad (6.3)$$

gdzie $\bar{\mathbf{a}}_1 = (\mathbf{q}_f - \mathbf{q}_0)/T - \mathbf{a}_k T^{k-1} - \mathbf{a}_{k-1} T^{k-2} - \dots - \mathbf{a}_2 T$ i wolnymi parametrami $\mathbf{a}_2, \dots, \mathbf{a}_k$. Trzecią klasę trajektorii referencyjnych stanowi odcinek z zaburzeniem sinusoidalnym o całkowitej wielokrotności pełnego okresu T

$$\boldsymbol{\lambda}(t) = \frac{\mathbf{q}_f - \mathbf{q}_0}{T} t + \mathbf{q}_0 + \mathbf{A} \sin\left(\frac{2\pi \mathbf{k}}{T} t\right), \quad t \in [0, T], \quad (6.4)$$

gdzie $\mathbf{A} \in \mathbb{R}^n$ oznacza wektor amplitud, natomiast $\mathbf{k} \in \mathbb{N}$ – harmoniczne, wpływające na wynikową częstotliwość. Odcinek prostej zapewnia osiągnięcie punktów brzegowych, natomiast sinusoida pozwala na regulowanie długości trajektorii przez zmiany amplitudy i częstotliwości.

6.1. Badania symulacyjne

Badania symulacyjne wpływu trajektorii referencyjnych na ewolucję układu nilpotentnego zostaną przeprowadzone na integratorze Brocketta o modelu (A.1) z zadaniem sterowania polegającym na przemieszczeniu układu z punktu początkowego $\mathbf{q}_0 = (q_{01}, q_{02}, q_{03})^T$ do docelowego $\mathbf{q}_f = (q_{f1}, q_{f2}, q_{f3})^T$ w zadanym czasie T . Integrator Brocketta jest sterowalnym układem nilpotentnym rzędu 2 (własności układu zawiera dodatek A), zatem układ rozszerzony jest postaci

$$\dot{\mathbf{q}} = \begin{bmatrix} 1 \\ 0 \\ -q_2 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 1 \\ q_1 \end{bmatrix} v_2 + \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} v_3 = \mathbf{g}_1(\mathbf{q})v_1 + \mathbf{g}_2(\mathbf{q})v_2 + [\mathbf{g}_1, \mathbf{g}_2](\mathbf{q})v_3. \quad (6.5)$$

Testowanymi trajektoriami referencyjnymi $\lambda(t)$ niech będą wielomiany stopnia k dane wzorem (6.3), które zapewniają spełnienie warunków brzegowych. Pochodne $\lambda(t)$ dane są zależnościami

$$\dot{\lambda}_i(t) = k a_{ik} t^{k-1} + (k-1) a_{i(k-1)} t^{k-2} + \dots + 2 a_{i2} t + \bar{a}_{i1}, \quad i = 1, 2, 3. \quad (6.6)$$

Zgodnie ze wzorem (3.3) sterowania rozszerzone dla integratora przyjmują postać

$$\mathbf{v}(t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{\lambda_2(t)}{2} & -\frac{\lambda_1(t)}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \dot{\lambda}_1(t) \\ \dot{\lambda}_2(t) \\ \dot{\lambda}_3(t) \end{bmatrix} = \begin{bmatrix} \dot{\lambda}_1(t) \\ \dot{\lambda}_2(t) \\ \frac{1}{2} (\dot{\lambda}_1(t)\lambda_2(t) - \lambda_1(t)\dot{\lambda}_2(t) + \dot{\lambda}_3(t)) \end{bmatrix}. \quad (6.7)$$

Następnym krokiem jest wyznaczenie równania CFS dla integratora Brocketta, które dla reprezentacji „wstecz” zadanej wzorem (3.10), zgodnie z przykładem 13, przyjmuje postać (3.16). Podstawiając zależności na sterowania rozszerzone (6.7) do wzoru (3.16) i uwzględniając warunek początkowy $\mathbf{h}(0) = \mathbf{0}$ otrzymujemy

$$\begin{aligned} h_1(t) &= \int_0^t \dot{\lambda}_1(s) ds, & h_2(t) &= \int_0^t \dot{\lambda}_2(s) ds, \\ h_3(t) &= \int_0^t \left(\frac{1}{2} (\dot{\lambda}_1(s)\lambda_2(s) - \lambda_1(s)\dot{\lambda}_2(s) + \dot{\lambda}_3(s)) + h_1(s)\dot{\lambda}_2(s) \right) ds. \end{aligned} \quad (6.8)$$

Zatem dwa pierwsze współczynniki Halla są następujące

$$h_i(t) = a_{ik} t^k + a_{i(k-1)} t^{k-1} + \dots + a_{i2} t^2 + \bar{a}_{i1} t, \quad i \in \{1, 2\}.$$

Natomiast trzeci współczynnik Halla wyliczony ze wzoru (6.8), po całkowaniu przez części, jest postaci

$$h_3(t) = \frac{1}{2} \lambda_1(t)\lambda_2(t) - \frac{q_{01}q_{02}}{2} - \int_0^t \lambda_1(s)\dot{\lambda}_2(s) ds + \int_0^t \frac{1}{2} \dot{\lambda}_3(s) ds + \int_0^t h_1(s)\dot{\lambda}_2(s) ds.$$

a po podstawieniu $\lambda_1(t) = h_1(t) + q_{01}$ dany wyrażeniem

$$h_3(t) = \frac{\lambda_1(t)\lambda_2(t)}{2} - \int_0^t q_{01}\dot{\lambda}_2(s)ds + \int_0^t \frac{1}{2}\dot{\lambda}_3(s)ds = \frac{\lambda_1(t)\lambda_2(t)}{2} - q_{01}\tilde{\lambda}_2(t) + \frac{1}{2}\tilde{\lambda}_3(t) - \frac{q_{01}q_{02}}{2},$$

gdzie $\tilde{\lambda}_i(t)$ oznacza $\lambda_i(t)$ bez składowej stałej. Zatem, aby zrealizować cel sterowania, należy wygenerować

$$\mathbf{S}(T) = e^{h_3(T)\mathbf{g}_1}\mathbf{g}_2 e^{h_2(T)\mathbf{g}_2} e^{h_1(T)\mathbf{g}_1}, \quad (6.9)$$

ze współczynnikami Halla w chwili końcowej $t = T$ równymi

$$\begin{aligned} h_1(T) &= q_{f1} - q_{01}, & h_2(T) &= q_{f2} - q_{02}, \\ h_3(T) &= \frac{1}{2}(q_{f1}q_{f2} - q_{01}q_{02} + q_{f3} - q_{03}) - q_{01}(q_{f2} - q_{02}). \end{aligned} \quad (6.10)$$

Najprostszymi sterowaniami wymuszającymi ruch w kierunkach odpowiednich pól wektorowych są funkcje kawałkami stałe. Korzystając z formuły CBHD (której szczegóły można znaleźć np. w [17]), jeden z możliwych scenariuszy generacji $\mathbf{S}(T)$ zadaje sekwencja strumieni determinujących sterowania

$$e^{h_3(T)\mathbf{g}_1}\mathbf{g}_2 e^{h_2(T)\mathbf{g}_2} e^{h_1(T)\mathbf{g}_1} = e^{\sqrt{h_3(T)}\mathbf{g}_1} e^{\sqrt{h_3(T)}\mathbf{g}_2} e^{-\sqrt{h_3(T)}\mathbf{g}_1} e^{-\sqrt{h_3(T)}\mathbf{g}_2} e^{h_2(T)\mathbf{g}_2} e^{h_1(T)\mathbf{g}_1}. \quad (6.11)$$

Cel sterowania jest osiągnięty konkatencją pięciu segmentów sterowań (uwzględniając $e^{-\sqrt{h_3(T)}\mathbf{g}_2} e^{h_2(T)\mathbf{g}_2} = e^{(-\sqrt{h_3(T)}+h_2(T))\mathbf{g}_2}$)

$$\mathbf{u}^t(s) = \begin{cases} u_1^t = 1 & u_2^t = 0 & s \in [0, \sqrt{h_3}] \\ u_1^t = 0 & u_2^t = 1 & s \in [\sqrt{h_3}, 2\sqrt{h_3}] \\ u_1^t = -1 & u_2^t = 0 & s \in [2\sqrt{h_3}, 3\sqrt{h_3}] \\ u_1^t = 0 & u_2^t = -1 & s \in [3\sqrt{h_3}, 4\sqrt{h_3} - h_2] \\ u_1^t = 1 & u_2^t = 0 & s \in [4\sqrt{h_3} - h_2, 4\sqrt{h_3} - h_2 + h_1], \end{cases} \quad (6.12)$$

gdzie $h_i = h_i(t)$. W sterowaniach $\mathbf{u}^t(s)$ zmienna t jest parametrem oznaczającym (bieżący) horyzont czasu ruchu (w nominalnym zadaniu $t = T$), a $s \in [0, t]$ odmierza czas rzeczywistego sterowania.

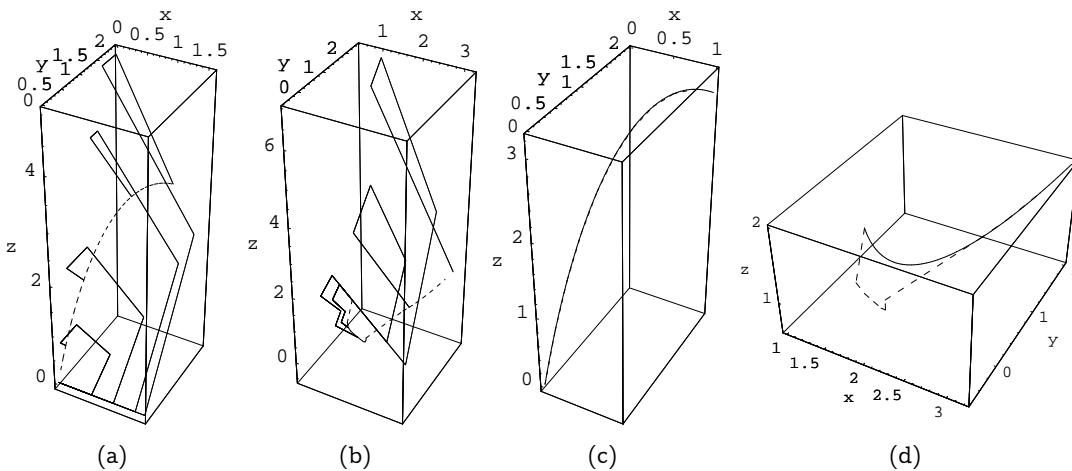
Wielomianowa trajektoria referencyjna wpływa na ruch poprzez współczynniki Halla $\mathbf{h}(T)$ zgodnie z równaniami (6.10). Okazuje się, że współczynniki α_i , $i = 2, \dots, k$ wielomianu oraz jego stopień nie występują w tych zależnościach, zatem nie mają wpływu na rzeczywisty ruch układu (innymi słowy, współczynniki α_i mają wpływ na zmiany współczynników Halla w czasie ($\mathbf{h}(t)$, $t \in [0, T]$), ale nie na ich wartość końcową).

Przeprowadzono podobne obliczenia dla przypadku trajektorii referencyjnej opisanej odcinkiem z zaburzeniem sinusoidalnym (6.4). Korzystając z zależności (6.8), po żmudnych obliczeniach, można wyliczyć współczynniki Halla dla czasu $t = T$, które są identycznie jak w poprzednim przypadku i zadane przez (6.10). Podobnie jak w poprzednim przypadku, zauważamy, że zmiana współczynników trajektorii referencyjnej (amplitudy - \mathbf{A} , częstotliwości - \mathbf{k}) nie ma wpływu na generowane sterowania i ewolucję układu. Stąd jeszcze raz potwierdza się wniosek, że trajektoria referencyjna nie ma wpływu na ewolucję integratora Brocketta.

6.1.1. Trajektoria symulacyjna

Dla integratora Brocketta trajektoria referencyjna nie ma wpływu na ewolucję układu, jednak jej dobór wpływa na ruch układu w inny sposób. Sterowania wyliczone na podstawie algorytmu LS (6.12) są zależne od dwóch zmiennych (t, s). Zmienna s oznacza rzeczywisty czas ruchu. Gdy chcemy przeprowadzić symulację na pełnym horyzoncie $s \in [0, T]$ wtedy $t = T$. Prześledzimy ewolucję integratora Brocketta dla $s \in [0, t]$ i $t \in [0, T]$. Oznacza to próbę zbadania ewolucji układu przy zmniejszonym czasie symulacji, ale zmniejszonym w takim sensie, że do wyliczonych zależności na współczynniki Halla zostanie podstawiony nie docelowy czas T lecz bieżący parametr $t \leq T$. Skoro tak ($t < T$), to układ nie zdoła osiągnąć punktu docelowego \mathbf{q}_f . Zatem seria punktów kończących trajektorie dla czasów $t \in [0, T]$ wyznaczy pewną trajektorię, którą nazywamy rzeczywistą trajektorią symulacyjną. Sposób powstawania trajektorii symulacyjnej ilustruje rys. 6.1a,b – wraz z rzeczywistymi trajektoriami, dla kilku czasów $t < T$ (dokładnie $0.1T, 0.3T, 0.7T$ i T). Na rys. 6.1c,d zestawiono trajektorie symulacyjną i zadaną. Na rys. 6.1c – trajektorie pokrywają się, natomiast na 6.1 d różnią się. Na podstawie przeprowadzonych symulacji ustalono następujące własności trajektorii symulacyjnej:

- rozpoczyna się w punkcie \mathbf{q}_0 a kończy \mathbf{q}_f ,
- w ogólności nie podąża za trajektorią zadaną,
- pokrywa się z zadaną, gdy współczynniki Halla $\mathbf{h}(t)$ dla każdego $t \in [0, T]$ nie zmieniają znaku. Innymi słowy, jeżeli dowolny punkt leżący na trajektorii zadanej



Rysunek 6.1. Wyniki symulacji dla integratora Brocketta dla sterowań (6.12), trajektorii referencyjnej w kształcie paraboli i warunkach brzegowych ($T = 1$): a),c) $\mathbf{q}_0 = (0, 0, 0)^T$, $\mathbf{q}_f = (1, 2, 3)^T$, $\mathbf{a}_2 = (0, 1, -5)^T$ b),d) $\mathbf{q}_0 = (1, 1, 1)^T$, $\mathbf{q}_f = (3.2, 2, 2)^T$, $T = 1$, $\mathbf{a}_2 = (0.2, 2.8, 2)^T$. Trajektoria symulacyjna (linia przerywana), trajektorie rzeczywiste (linie ciągłe)

da się wygenerować przy pomocy identycznych pól (z dokładnością do znaku) jak punkt końcowy \mathbf{q}_f .

6.1.2. Podsumowanie

Podsumujmy eksperymenty przeprowadzone na integratorze Brocketta. Pokazano (przeciwnie niż sugerowali autorzy oryginalnego algorytmu LS), że trajektoria referencyjna nie ma wpływu na ewolucję integratora Brocketta. Zatem za jej pomocą nie można kształtować trajektorii osiągnięcia zadanego punktu w przestrzeni stanu. Co więcej, trajektoria referencyjna nie ma praktycznie żadnego wpływu na rzeczywistą trajektorię realizowaną, co ma szczególne znaczenie w środowiskach kolizyjnych. Wydaje się, że przyczyna braku wpływu trajektorii referencyjnej na ewolucję układu nilpotentnego (nie tylko integratora Brocketta) tkwi w charakterze algorytmu LS dla układów nilpotentnych. Można przypuszczać, że na przestrzeni stanu rozpięta jest pewna funkcja przypominająca energię potencjalną układów mechanicznych. Zatem dla zadanych punktów początkowego i końcowego, ewolucja między nimi będzie zależna jedynie od różnicy energii na warstwicach odpowiadających punktom brzegowym, a nie od trajektorii je łączących.

6.2. Niezależności współczynników Halla od trajektorii referencyjnej

Wynikiem końcowym działania dwóch pierwszych etapów algorytmu LS są współczynniki bazowe $\mathbf{h}(T)$. Współczynniki Halla, Lyndona bądź Chibrikova (ze względu na wybraną bazę) oraz zależnie od wybranej reprezentacji (4.1) opisują czasy ruchu wzdłuż kolejnych pól wektorowych reprezentacji. Na rozwiązanie równań CFS, a zatem na $\mathbf{h}(T)$, wpływa, poprzez sterowania rozszerzone, trajektoria referencyjna. W podrozdziale 6.1, por. (6.10), pokazano symulacyjnie brak wpływu trajektorii referencyjnej na ruch integratora Brocketta. Podobny wniosek sformułowano także dla układów łańcuchowych. Spróbujmy, więc uogólnić te spostrzeżenia.

Twierdzenie 2. *Trajektoria referencyjna nie wpływa na końcową wartość $\mathbf{S}(T)$, niezależnie od wyboru reprezentacji.*

Dowód: Dowód przeprowadzimy przez sprowadzenie do sprzeczności negacji tezy. Ustalmy zatem reprezentację ruchu i załóżmy, że trajektoria referencyjna wpływa na $\mathbf{S}(T)$ (a więc na $\mathbf{h}(T)$) nie tylko przez swoje wartości brzegowe $\boldsymbol{\lambda}(0) = \mathbf{q}_0$, $\boldsymbol{\lambda}(T) = \mathbf{q}_f$, lecz także przebieg na przedziale $t \in [0, T]$. Wybierzmy więc dwie takie różne trajektorie referencyjne $\hat{\boldsymbol{\lambda}}(\cdot)$ i $\tilde{\boldsymbol{\lambda}}(\cdot)$, które generują różne $\hat{\mathbf{S}}(T) \neq \tilde{\mathbf{S}}(T)$. Oczywiście $\hat{\boldsymbol{\lambda}}(0) = \tilde{\boldsymbol{\lambda}}(0) = \mathbf{q}_0$ i $\hat{\boldsymbol{\lambda}}(T) = \tilde{\boldsymbol{\lambda}}(T) = \mathbf{q}_f$ z definicji trajektorii referencyjnej. Rozpatrujemy teraz nowe za-

danie planowania, z identyczną reprezentacją ruchu, przeprowadzenia układu z punktu \mathbf{q}_0 do \mathbf{q}_f przez punkt \mathbf{q}_f dla którego trajektorię referencyjną definiujemy jako

$$\boldsymbol{\lambda}(t) = \begin{cases} \hat{\boldsymbol{\lambda}}(t) & \text{dla } t \in [0, T] \\ \tilde{\boldsymbol{\lambda}}(2T - t) & \text{dla } t \in [T, 2T]. \end{cases} \quad (6.13)$$

(zakładamy, że składowe trajektorie referencyjne kleją się gładko w punkcie \mathbf{q}_f). Wynikowe $\mathbf{S}(2T)$ dla nowego zadania wynosi $\mathbf{S}(2T) = \hat{\mathbf{S}}(T)\tilde{\mathbf{S}}^{-1}(T) \neq \mathbf{I}$, a to oznacza, że układ nie powróci do zadanego punktu początkowego \mathbf{q}_0 . Nieprawdziwość negacji tezy dowodzi jej prawdziwości. \square

Brak wpływu trajektorii referencyjnej na ruch powoduje, iż trajektoria $\boldsymbol{\lambda}(t)$ powinna być tak wybrana, aby wyliczenie sterowań rozszerzonych było możliwie mało skomplikowane. Odcinek łączący punkt startowy i końcowy (6.2) spełnia to wymaganie. Zatem wzór na sterowania rozszerzone (3.3) przyjmie postać

$$\mathbf{v}(t) = \mathbf{G}^\# \left(\frac{\mathbf{q}_f - \mathbf{q}_0}{T} t + \mathbf{q}_0 \right) \left(\frac{\mathbf{q}_f - \mathbf{q}_0}{T} \right), \quad t \in [0, T]. \quad (6.14)$$

7. Realizacja sterowań za pomocą formuły gCBHD

Całkując równania CFS dla zadanego $\mathbf{v}(t)$, $t \in [0, T]$ i przyjętej reprezentacji (4.1) otrzymujemy $S(T)$. Ostatnim etapem algorytmu LS jest generacja $S(T)$, złożonego z ruchów elementarnych $e^{(\rho_i \star \mathbf{h}(T), \mathbf{B}_i)}$, $i = 1, \dots, \bar{r}$, za pomocą sterowań. W dalszych rozważaniach pokazano jak wygenerować jeden (dowolny) ruch elementarny. Do generacji sterowań wykorzystano formułę gCBHD, a jako klasę sterowań dopuszczalnych wybrano sterowania harmoniczne parametryzowane skończonym wektorem parametrów sterowań \mathbf{p} . Zatem zadanie polega na wyznaczeniu takiego \mathbf{p}^* , który dla danego wektora β_d i zadanych funkcji $f_{g_i}(\mathbf{p}, T)$ występujących w zależności (3.26) a wyliczonych na podstawie równania (3.25) rozwiązuje równanie $\beta_d = \mathbf{f}_g(\mathbf{p})$. Oczywiście, wyznaczone \mathbf{p}^* determinuje $\mathbf{u}(\cdot)$.

Szczególnymi ruchami elementarnymi są te w kierunkach bazowych, czyli $e^{h_i(T), \mathbf{B}_i}$, dla których sterowania można wyliczyć analitycznie minimalizując globalnie energię ruchu (podrozdział 7.1), gdy stopień \mathbf{B}_i jest niewielki. Dla wszystkich pozostałych przypadków stosujemy metodę Newtona, znaną z rozwiązania zadania kinematyki odwrotnej manipulatorów (podrozdział 7.2).

7.1. Generacja optymalnych energetycznie sterowań w kierunkach ruchu wyznaczonych przez bazę Halla

Przez generację kierunków bazy Halla rozumiemy przypadek, gdy sterowania generują pożądaną kierunek nie powodując ruchu w żadnym z kierunków bazy Halla warstw niższych i równych z generowanym polem. Przedstawimy optymalne energetycznie sterowania harmoniczne dla układów dwuwęściowych, $m = 2$, o generatorach $\mathbf{g}_1, \mathbf{g}_2$ z odpowiadającymi generatorami formalnymi $\mathbf{B}_1, \mathbf{B}_2$, oraz pól bazy Halla co najwyżej trzeciej warstwy (uogólnienie na pola warstw wyższych jest możliwe po przeprowadzeniu coraz bardziej skomplikowanych obliczeń), gdyż takie sterowania będą szczególnie chętnie wykorzystywane w symulacjach działania algorytmów planowania ruchu. Czas sterowania będzie ustalony na T . Dla tego przypadku przyjmujemy oznaczenie $\mathbf{f}_g(\mathbf{p}) = (\beta_{g_1}, \beta_{g_2}, \beta_{[g_1, g_2]}, \beta_{[g_1, [g_1, g_2]]}, \beta_{[g_2, [g_1, g_2]]}, \dots)^T$.

Korzystamy z osiągnięć pracy [17], w której wykazano, że dla minimalizacji energii sterowań podczas generacji ruchu w kierunku $[g_1, g_2]$ sterowania harmoniczne muszą być o identycznych częstotliwościach (pierwsze harmoniczne) oraz przesunięte wzglę-

dem siebie o $\pm\pi/2$. Dla ruchu w kierunkach $[\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]]$, $[\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]]$, sterowanie odpowiadające polu z większą liczbą generatorów musi mieć częstotliwość podwojoną w stosunku do drugiego sterowania o podstawowej harmonicznej. Ponadto dla każdego z pól $[\mathbf{g}_1, \mathbf{g}_2]$, $[\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]]$, $[\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]]$, do każdego ze sterowań można dodać dodatkowe przesunięcie fazowe, które nie zmienia jakości generowanego kierunku dla układów nilpotentnych. Należy jednak zwrócić uwagę, że dla przypadku ogólnych bezdryfowych układów nieholonomicznych zarówno wybór znaku przesunięcia fazowego jak i jego dodatkowa wartość może służyć jako wspomagająca zmienna optymalizowana podczas generacji ruchu (zmiana dodatkowego przesunięcia fazowego wpływa na współczynniki pól z wyższych warstw, a te nie są automatycznie zerowane, tak jak w przypadku układów nilpotentnych). Dla uproszczenia rozważań przyjmujemy, że wspólne przesunięcie fazowe dla obu sterowań wynosi 0, natomiast znak przesunięcia fazowego w jednym z nich jest dodatni.

Wyznaczenie harmonicznym sterowań optymalnych przeprowadzimy z wykorzystaniem formuły gCBHD metodą mnożników Lagrange'a, która służy do optymalizacji funkcji jakości z ograniczeniami (nie-)równościami [31]. Rozpocznijmy od najtrudniejszego z rozważanych zadań, by pokazać tok obliczeń na nietrywialnym przykładzie.

Celem jest wygenerowanie optymalnych sterowań harmonicznym realizujących ruch w kierunku pola $[\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]]$ o amplitudzie określonej przez zadane $\beta_{[\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]]}$, zob. równania (3.24) dla $t = T$ oraz (3.25). Sterowania optymalne dla tego przypadku są postaci

$$u_1(t) = p_1 \sin\left(\frac{2\pi t}{T} + \frac{\pi}{4}\right), \quad u_2(t) = p_2 \sin\left(2\frac{2\pi t}{T}\right), \quad t \in [0, T], \quad (7.1)$$

z parametrami p_1, p_2 , które należy określić. Podstawiając sterowania (7.1) do formuły gCBHD (3.24) otrzymujemy

$$\beta_{[\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]]} = -\frac{p_1^2 p_2 T^3}{32\pi^2}$$

zerując, zgodnie z założeniem, $\beta_{\mathbf{g}_1}, \beta_{\mathbf{g}_2}, \beta_{[\mathbf{g}_1, \mathbf{g}_2]}, \beta_{[\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]]}$. Ponieważ w praktycznych przypadkach wartość $\beta_{[\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]]}$ jest znana (dalej przyjmujemy jej niezerową wartość) otrzymujemy ograniczenie równościowe postaci

$$p_1^2 p_2 = -\frac{32\pi^2 \beta_{[\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]]}}{T^3} = C \neq 0. \quad (7.2)$$

Optymalizując energię sterowań (7.1) otrzymujemy funkcję jakości zależną od parametrów sterowań

$$\tilde{J}(\mathbf{u}(\cdot)) = \int_0^T (u_1^2(t) + u_2^2(t)) dt = \frac{T}{2} (p_1^2 + p_2^2) \quad \rightarrow \quad J(\mathbf{p}) = (p_1^2 + p_2^2).$$

Opuszczenie współczynnika w funkcji optymalizowanej nie wpływa na rozmieszczenie punktów, w których ekstremum jest przyjmowane, a jedynie na jej wartość funkcji. Wykorzystując metodę mnożników Lagrange'a definiujemy funkcję

$$L(\mathbf{p}, \lambda) = p_1^2 + p_2^2 + \lambda(p_1^2 p_2 - C)$$

z nieoznaczonym mnożnikiem λ . Warunki konieczne ekstremum są następujące:

$$\frac{\partial L(\mathbf{p}, \lambda)}{\partial p_1} = 2p_1 + 2\lambda p_1 p_2 = 0 = 2p_1(1 + \lambda p_2) \Rightarrow p_2 = -\frac{1}{\lambda} \quad (7.3)$$

$$\frac{\partial L(\mathbf{p}, \lambda)}{\partial p_2} = 2p_2 + \lambda p_1^2 = 0 \Rightarrow p_1^2 = -\frac{2p_2}{\lambda} = \frac{2}{\lambda^2}. \quad (7.4)$$

Podstawiając (7.3), (7.4) do (7.2) wyznaczamy mnożnik Lagrange'a

$$C = p_1^2 p_2 = -\frac{2}{\lambda^3} \Rightarrow \lambda = -\sqrt[3]{\frac{2}{C}} = \frac{T}{2} \sqrt[3]{\frac{1}{2\pi^2 \beta_{[\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]]}}},$$

a następnie współczynniki sterowań p_1 i p_2 :

$$p_1 = \frac{2\sqrt{2}}{T} \sqrt[3]{2\pi^2 |\beta_{[\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]]}|}, \quad p_2 = -\frac{2}{T} \sqrt[3]{2\pi^2 \beta_{[\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]]}}. \quad (7.5)$$

Zauważmy, że znak sterowania u_1 nie wpływa na jakość generowania kierunku $[\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]]$.

Postępując analogicznie wyznaczmy optymalne sterowania generujące jedynie kierunek pola wektorowego $[\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]]$. Optymalne sterowania [17] są w postaci

$$u_1(t) = p_1 \sin\left(2\frac{2\pi t}{T}\right), \quad u_2(t) = p_2 \sin\left(\frac{2\pi t}{T} + \frac{\pi}{4}\right). \quad (7.6)$$

wiodące do zależności współczynnika $\beta_{[\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]]}$ od parametrów sterowań opisaną wyrażeniem

$$\beta_{[\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]]} = \frac{p_1 p_2^2 T^3}{32\pi^2}.$$

Powtarzając kroki metody mnożników Lagrange'a ze zmianą $p_1 \leftrightarrow p_2$ i uwzględniając zmianę znaku stałej C otrzymujemy optymalne parametry sterowań (7.6)

$$p_1 = \frac{2}{T} \sqrt[3]{2\pi^2 \beta_{[\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]]}}, \quad p_2 = \frac{2\sqrt{2}}{T} \sqrt[3]{2\pi^2 |\beta_{[\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]]}|}.$$

W podobny sposób generujemy optymalny ruch w kierunku pola wektorowego $[\mathbf{g}_1, \mathbf{g}_2]$ sterowaniami

$$u_1(t) = p_1 \sin\left(\frac{2\pi t}{T} + \frac{\pi}{2}\right), \quad u_2(t) = p_2 \sin\left(\frac{2\pi t}{T}\right), \quad (7.7)$$

odpowiadającym sterowaniom (7.7) współczynnikiem

$$\beta_{[\mathbf{g}_1, \mathbf{g}_2]} = \frac{p_1 p_2 T^2}{4\pi} \quad (7.8)$$

i optymalnymi energetycznie wartościami parametrów

$$p_1 = \operatorname{sgn}(\beta_{[\mathbf{g}_1, \mathbf{g}_2]}) \frac{2}{T} \sqrt{\pi |\beta_{[\mathbf{g}_1, \mathbf{g}_2]}|}, \quad p_2 = \frac{2}{T} \sqrt{\pi |\beta_{[\mathbf{g}_1, \mathbf{g}_2]}|}. \quad (7.9)$$

W zależności (7.9) możliwe jest synchroniczne odwrócenie znaków p_1, p_2 .

Dla dopełnienia opisu optymalnego ruchu dla wszystkich pól do warstwy trzeciej włącznie konieczne jest wskazanie sterowań generujących ruch w kierunkach \mathbf{g}_1 i \mathbf{g}_2 . Pierwsze z nich są postaci

$$\mathbf{u}_1(t) = \frac{\beta \mathbf{g}_1}{T}, \quad \mathbf{u}_2(t) = 0,$$

a te odpowiadające przemieszczeniu w kierunku \mathbf{g}_2 są następujące

$$\mathbf{u}_1(t) = 0, \quad \mathbf{u}_2(t) = \frac{\beta \mathbf{g}_2}{T}.$$

7.2. Generacja sterowań algorytmem Newtona

Jeśli nie jest możliwe efektywne zastosowanie metody mnożników Lagrange'a (tzn. równania mogą być sformułowane zawsze, lecz ich analityczne rozwiązanie nie jest możliwe, ze względu na skomplikowanie i wymiarowość zadania), stosujemy podejście znane z pracy [17]. Najpierw dla ustalonej reprezentacji sterowań (przykładowo harmonicznym) zadanej wektorem parametrów \mathbf{p} wyliczamy pierwsze r współczynników β ze wzorów (3.24), (3.25). W ten sposób powstaje odwzorowanie przypominające kinematykę manipulatora $\mathbf{f}(\mathbf{p}) = \beta$, zwykle $n < r < \dim(\mathbf{p})$. Każdy pożądanym ruch elementarny $e^{(\mathbf{p}_i \star \mathbf{h}^{(T), \mathbf{B}})}$, $i = 1, \dots, \bar{r}$ jednoznacznie zadaje wektor β_d . Parametry sterowań realizujące ten wektor są wyznaczone algorytmem Newtona [74]

$$\mathbf{p}_{i+1} = \mathbf{p}_i + \xi \cdot \mathbf{J}^\#(\mathbf{p}_i)(\beta_d - \mathbf{f}(\mathbf{p}_i)), \quad (7.10)$$

gdzie i to licznik iteracji, ξ jest dodatnim parametrem rzeczywistym od którego zależy szybkość zbieżności procesu iteracyjnego, \mathbf{p}_0 jest wybranym początkowym zestawem parametrów sterowań, $\mathbf{J}(\mathbf{p}) = \partial \mathbf{f}(\mathbf{p}) / \partial \mathbf{p}$ – jacobianem, a $\mathbf{J}^\# = \mathbf{J}^T (\mathbf{J} \cdot \mathbf{J}^T)^{-1}$ pseudoodwrotnością Moore'a-Penrose'a. Można także zastosować algorytmu Newtona z dodatkową optymalizacją funkcji kryterialnej w przestrzeni zerowej

$$\mathbf{p}_{i+1} = \mathbf{p}_{i+1} + \xi \cdot \mathbf{J}^\#(\mathbf{p}_i)(\beta_d - \mathbf{f}(\mathbf{p}_i)) + (\mathbf{I} - \mathbf{J}^\#(\mathbf{p}_i)\mathbf{J}(\mathbf{p}_i))\mathbf{v}(\mathbf{p}_i), \quad (7.11)$$

gdzie $\mathbf{v}(\mathbf{p}_i) \in \mathbb{R}^{\dim(\mathbf{p})}$ jest dowolnym wektorem. Najczęściej optymalizację przeprowadza się w celu minimalizacji dodatkowej funkcji kryterialnej $d(\mathbf{p})$ (przykładowo energii sterowań). Wtedy \mathbf{v} w (7.11) przyjmuje postać antygradientu $\mathbf{v}(\mathbf{p}) = -\partial d(\mathbf{p}) / \partial \mathbf{p}$.

8. Aproksymacja nilpotentna układów bezdryfowych

Idea metody sterowania wykorzystującej nilpotentną aproksymację ogólnego układu bezdryfowego jest następująca: najpierw wygenerować nilpotentny odpowiednik układu sterowanego, który aproksymuje układ oryginalny wokół bieżącej konfiguracji; rozwiązać zadanie sterowania dla układu nilpotentnego (przykładowo algorytmem LS); sterowania znalezione dla układu przybliżonego wykorzystać do sterowania układem oryginalnym otrzymując pewną konfigurację końcową. Jeśli ta konfiguracja nie jest punktem docelowym planowania – procedurę powtarzać, jako początkową przyjmując osiągniętą konfigurację.

Jedną z pierwszych procedur nilpotentnej aproksymacji zaproponował Hermes [38], a efektywne rozwinięcie jego metody dla układów bezdryfowych opisano w artykule [4], w którym autorzy zaproponowali transformację układu do tak zwanej postaci trójkątnej. Dzięki spełnieniu odpowiednich warunków rzędu, funkcje pojawiające się w postaci trójkątnej tworzą układ nilpotentny. Rozszerzenie algorytmu dla układów z osobliwościami (w których rząd przestrzeni rozpiętej przez pola wektorowe kolejnych warstw układu nie rośnie tak szybko jak mógłby) opisuje artykuł [105]. Nilpotentna aproksymacja była także wykorzystana dla szczególnych instancji układów [58, 83, 106]. Sachkova [82] pokazała, że każdy dwuwęściowy układ nilpotentny może być transformowany do innego układu nilpotentnego, o $\mathbf{q} \in \mathbb{R}^3$, z zerującymi się polami poza pierwszymi dwoma warstwami.

Zatem algorytm nilpotentnej aproksymacji posłuży w dalszej części dysertacji jako wspomagający metodę planowania ruchu układów bezdryfowych. Z tego powodu opisano jego kroki bazując na pracy Lamounda i współpracowników [105] oraz przetestowano na dwóch modelach wykorzystując oprogramowanie powstałe na użytek pracy magisterskiej Koralewskiego [50]. Zaproponowano oryginalne miary efektywności nilpotentnej aproksymacji, które były przedmiotem pracy [24] oraz je przetestowano.

8.1. Algorytm nilpotentnej aproksymacji

Pola wektorowe zapisywane zwykle jako wektory funkcji, w tym rozdziale będą zapisywane jako operatory różniczkowe, gdyż w takiej notacji występują w algorytmie nilpotentnej aproksymacji [58]. Notacja ta jest logiczna i bogata w interpretacje. Dla

modelu jednokołowca (A.3) zapisowi wektorowemu pól $\mathbf{g}_1(\mathbf{q}) = (\cos(q_3), \sin(q_3), 0)^T$, $\mathbf{g}_2(\mathbf{q}) = (0, 0, 1)^T$, odpowiada następująca ich notacja różniczkowa

$$\mathbf{g}_1(\mathbf{q}) = \cos(q_3)\partial_{q_1} + \sin(q_3)\partial_{q_2}, \quad \mathbf{g}_2(\mathbf{q}) = 1 \cdot \partial_{q_3}.$$

gdzie $\partial_{q_i} = \frac{\partial}{\partial q_i}$.

Niech $\mathcal{L}_m^k(\mathbf{q})$ oznacza przestrzeń wektorową rozpiętą w konfiguracji \mathbf{q} przez pola wektorowe, utworzone z generatorów układu (2.36) w liczbie m , o stopniu nie większym niż k (czyli pola do warstwy k -tej włącznie). Jeśli pola wektorowe powstają z bazy Halla, o dwóch generatorach ($\mathbf{g}_1, \mathbf{g}_2$) to kolejne podprzestrzenie są następujące

$$\begin{aligned} \mathcal{L}_2^1(\mathbf{q}) &= \text{span}(\mathbf{g}_1(\mathbf{q}), \mathbf{g}_2(\mathbf{q})), & \mathcal{L}_2^2(\mathbf{q}) &= \text{span}(\mathbf{g}_1(\mathbf{q}), \mathbf{g}_2(\mathbf{q}), [\mathbf{g}_1, \mathbf{g}_2](\mathbf{q})), \\ \mathcal{L}_2^3(\mathbf{q}) &= \text{span}(\mathbf{g}_1(\mathbf{q}), \mathbf{g}_2(\mathbf{q}), [\mathbf{g}_1, \mathbf{g}_2](\mathbf{q}), [\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]](\mathbf{q}), [\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]](\mathbf{q})), & \dots \end{aligned}$$

Definicja 1. *Wektor wzrostu (w konfiguracji \mathbf{q})*

Wektor $\mathbf{k}(\mathbf{q}) = (\mathbf{k}_1(\mathbf{q}), \dots, \mathbf{k}_r(\mathbf{q})) \in \mathbb{N}^r$ o współrzędnych

$$\mathbf{k}_i(\mathbf{q}) = \dim(\mathcal{L}_2^i(\mathbf{q})), \quad i \in \{1, \dots, r\}, \quad (8.1)$$

nazywany jest wektorem wzrostu, gdzie r to stopień nieholonomiczności układu w konfiguracji \mathbf{q} .

Definicja 2. *Stopień nieholonomiczności (w konfiguracji \mathbf{q})*

$$r = \{\min_i : \mathbf{k}_i(\mathbf{q}) > n\}, \quad (8.2)$$

to minimalna liczba warstw algebry sterowalności układu ($n = \dim(\mathbf{q})$).

Definicja 3. *Wektor wag (w konfiguracji \mathbf{q})*

Wektor $\boldsymbol{\omega}(\mathbf{q}) \in \mathbb{N}^n$ złożony z numerów warstw, do których należą kolejne elementy algebry Liego (algebry sterowalności) układu nazywa się wektorem wag

$$\boldsymbol{\omega}_i(\mathbf{q}) = \deg(\mathbf{g}_i), \quad i \in \{1, \dots, n\}, \quad (8.3)$$

gdzie n jest wymiarem przestrzeni konfiguracyjnej.

Wektory wzrostu i wag zależą od konfiguracji, w której są obliczane i zmieniają się, jeśli pola, gwarantujące spełnienie warunku rzędu (2.37), pochodzą z różnych warstw. Jeśli dla każdej z konfiguracji dane powyższe są stałe można opuścić w ich nazwie zależność od \mathbf{q} .

Procedura nilpotentnej aproksymacji sterowalnego układu (2.36) wokół konfiguracji \mathbf{q}_a składa się z następującej sekwencji kroków [50, 105]:

Krok 1 Z bazy algebry Liego rozpiętej przez generatory układu (2.36) wybrać pola wektorowe rozpinające przestrzeń stanu w punkcie \mathbf{q}_a (czyli spełnić warunek rzędu (2.37)) i oznaczyć je przez $\bar{\mathbf{g}}_i$, $i \in \{1, \dots, n\}$.

Krok 2 W konfiguracji \mathbf{q}_a wyliczyć wektory wzrostu (8.1) i wag (8.3) odpowiadające $\bar{\mathbf{g}}_i$ dla $i \in \{1, \dots, n\}$.

Krok 3 Na podstawie oryginalnych współrzędnych \mathbf{q} wyznaczyć lokalne współrzędne \mathbf{y} z zależności

$$\mathbf{y} = \bar{\mathbf{G}}^{-1}(\mathbf{q}_a)(\mathbf{q} - \mathbf{q}_a). \quad (8.4)$$

Kolumnami macierzy $\bar{\mathbf{G}}(\mathbf{q}_a) = [\bar{\mathbf{g}}_1(\mathbf{q}_a), \dots, \bar{\mathbf{g}}_n(\mathbf{q}_a)]$ są wektory wchodzące w skład zestawu pól wektorowych spełniających twierdzenie Chow w punkcie \mathbf{q}_a . W notacji operatorów różniczkowych pola zapisują się jako

$$\mathbf{g}_j(\mathbf{q}_a) = \sum_{i=1}^n \bar{\mathbf{G}}_{ij}(\mathbf{q}_a) \partial_{q_i} |_{\mathbf{q}_a}, \quad j = 1, \dots, n. \quad (8.5)$$

Zamiana współrzędnych oznacza przeskalowanie i centrowanie pól wokół punktu \mathbf{q}_a .

Krok 4 Wyliczyć współrzędne uprzywilejowane $\mathbf{z} = (z_1, \dots, z_n)^T$ wokół \mathbf{q}_a z rekurencyjnej zależności

$$z_j = y_j + \sum_{k=2}^{\omega_j-1} h_k(y_1, \dots, y_{j-1}), \quad j = 1, \dots, n, \quad (8.6)$$

z funkcjami h_k określonymi przez

$$h_k(y_1, \dots, y_{j-1}) = - \sum_{\substack{|\alpha|=k \\ \omega(\alpha) < \omega_j}} m_j \cdot \bar{\mathbf{g}}_1^{\alpha_1} \cdots \bar{\mathbf{g}}_{j-1}^{\alpha_{j-1}} \left(y_j + \sum_{l=2}^{k-1} h_l \right) (\mathbf{q}_a), \quad (8.7)$$

gdzie $|\alpha| = \sum_{i=1}^n \alpha_i$, natomiast $m_j = \prod_{i=1}^{j-1} y_i^{\alpha_i} / \alpha_i!$. Pierwszy warunek $|\alpha| = k$ we wzorze (8.7) stanowi wstępną selekcję wektorów dopuszczalnych α , a drugi warunek $\omega(\alpha) < \omega_j$ wprowadza ostateczną selekcję. Konkatenację występującą we wzorze (8.7) interpretuje się w naturalny sposób, czyli $\bar{\mathbf{g}}^\beta \bar{\mathbf{g}}^\delta(\mathbf{q}) = \bar{\mathbf{g}}^{\beta+\delta}(\mathbf{q})$, $\bar{\mathbf{g}}^\beta(\mathbf{q}) = \bar{\mathbf{g}}^{\beta-1}(\bar{\mathbf{g}}(\mathbf{q}))$, $\bar{\mathbf{g}}^1(\mathbf{q}) = \bar{\mathbf{g}}(\mathbf{q})$.

Krok 5 Wyrazić układ oryginalny (2.36) we współrzędnych uprzywilejowanych

$$\dot{\mathbf{z}} = \frac{\partial \mathbf{F}_2}{\partial \mathbf{y}} |_{\mathbf{F}_2^{-1}(\mathbf{z})} \frac{\partial \mathbf{F}_1}{\partial \mathbf{q}} |_{\mathbf{F}_1^{-1}(\mathbf{F}_2^{-1}(\mathbf{z}))} \mathbf{G}(\mathbf{F}_1^{-1}(\mathbf{F}_2^{-1}(\mathbf{z}))) \mathbf{u} = \tilde{\mathbf{G}}(\mathbf{z}) \mathbf{u} = \sum_{i=1}^m \tilde{\mathbf{g}}_i(\mathbf{z}) u_i, \quad (8.8)$$

gdzie transformację (8.4) ze współrzędnych oryginalnych \mathbf{q} do \mathbf{y} zadaje dyfeomorfizm $\mathbf{y} = \mathbf{F}_1(\mathbf{q})$, natomiast transformację (8.6) z \mathbf{y} do \mathbf{z} opisuje $\mathbf{z} = \mathbf{F}_2(\mathbf{y})$.

Krok 6 Rozwinąć w szereg Taylora pola wektorowe $\tilde{\mathbf{g}}_i(\mathbf{z})$ w punkcie $\mathbf{0}$ i wyrazić je w postaci jednorodnych, ze względu na stopień pól wektorowych

$$\tilde{\mathbf{g}}_i(\mathbf{z}) = \tilde{\mathbf{g}}_i^{(-1)}(\mathbf{z}) + \tilde{\mathbf{g}}_i^{(0)}(\mathbf{z}) + \tilde{\mathbf{g}}_i^{(1)}(\mathbf{z}) + \dots \quad j = 1, \dots, m. \quad (8.9)$$

Krok 7 Niech $\hat{\mathbf{g}}_i(\mathbf{z}) = \tilde{\mathbf{g}}_i^{(-1)}(\mathbf{z})$, wtedy aproksymant układu (2.36) w konfiguracji \mathbf{q}_a jest zdefiniowany jako

$$\dot{z}_j = \sum_{i=1}^m \hat{\mathbf{g}}_{ij}(z_1, \dots, z_{j-1}) \mathbf{u}_i, \quad j = 1, \dots, n, \quad (8.10)$$

gdzie $\hat{\mathbf{g}}_{ij}$ są jednorodnymi wielomianami o ważonej wadze równej $\omega_j - 1$.

Układ (8.10) jest nilpotentną aproksymacją układu (2.36) we współrzędnych \mathbf{z} wyliczony w punkcie \mathbf{q}_a . Warto zauważyć, że powyższy algorytm zawiera selekcję pól w Kroku 1, zatem powinien być uzupełniony o sposób ich wyboru. Naturalnie wybierane są pola z bazy (np. Halla) algebry Liego, które wpływają na spełnienie warunku rzędu algebry Liego. Działanie algorytmu nilpotentnej aproksymacji zilustrujemy na przykładzie dwóch układów, na których będą testowane także algorytmy planowania ruchu bazujące na nilpotentnej aproksymacji. Wyniki szczegółowe wygenerowano oprogramowaniem powstałym na użytek pracy dyplomowej [50].

Przykład 19. *Nilpotentna aproksymacja jednokołowego robota mobilnego.*

Model robota o wektorze stanu $\mathbf{q} = (q_1, q_2, q_3)^\top$ opisuje równanie (A.3), a punktem wokół którego model jest aproksymowany będzie $\mathbf{q}_a = (q_{a1}, q_{a2}, q_{a3})^\top$. Kilka początkowych elementów bazy Halla algebry Liego dla jednokołowca to $[\mathbf{g}_1(\mathbf{q}), \mathbf{g}_2(\mathbf{q})] = (\sin q_3, -\cos q_3, 0)^\top$, $[\mathbf{g}_1(\mathbf{q}), [\mathbf{g}_1(\mathbf{q}), \mathbf{g}_2(\mathbf{q})]] = (0, 0, 0)^\top$, $[\mathbf{g}_2(\mathbf{q}), [\mathbf{g}_1(\mathbf{q}), \mathbf{g}_2(\mathbf{q})]] = \mathbf{g}_1(\mathbf{q})$. Warunek rzędu algebry Liego jest spełniony przez trzy pierwsze elementy bazy Halla, zatem $\bar{\mathbf{g}}_1 = \mathbf{g}_1$, $\bar{\mathbf{g}}_2 = \mathbf{g}_2$ oraz $\bar{\mathbf{g}}_3 = [\mathbf{g}_1, \mathbf{g}_2]$. Wektory wag i wzrostu wynoszą więc

$$\boldsymbol{\omega} = (1, 1, 2)^\top, \quad \mathbf{k} = (2, 3)^\top. \quad (8.11)$$

Współrzędne lokalne \mathbf{y} , zgodnie ze wzorem (8.4), określa zależność

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} \cos q_{a3} & 0 & \sin q_{a3} \\ \sin q_{a3} & 0 & -\cos q_{a3} \\ 0 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} q_1 - q_{a1} \\ q_2 - q_{a2} \\ q_3 - q_{a3} \end{bmatrix} = \begin{bmatrix} \cos q_{a3} & \sin q_{a3} & 0 \\ 0 & 0 & 1 \\ \sin q_{a3} & -\cos q_{a3} & 0 \end{bmatrix} \begin{bmatrix} q_1 - q_{a1} \\ q_2 - q_{a2} \\ q_3 - q_{a3} \end{bmatrix},$$

gdzie macierz $\bar{\mathbf{G}} \in \mathbb{SO}(3)$ jest macierzą obrotu w \mathbb{R}^3 ($\bar{\mathbf{G}}^{-1} = \bar{\mathbf{G}}^\top$). Po wymnożeniu, współrzędne lokalne przyjmują postać

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} (q_1 - q_{a1}) \cos q_{a3} + (q_2 - q_{a2}) \sin q_{a3} \\ q_3 - q_{a3} \\ (q_1 - q_{a1}) \sin q_{a3} - (q_2 - q_{a2}) \cos q_{a3} \end{bmatrix}. \quad (8.12)$$

Następnie wykorzystując wzór (8.6) otrzymujemy

$$z_i = y_i, \quad i \in \{1, 2, 3\}. \quad (8.13)$$

Współrzędne uprzywilejowane przyjmują szczególnie łatwą postać (8.13), gdyż we wzorze (8.6) sumowane są wartości od $k = 2$ do $\omega_j - 1$, zatem dla zadanego wektora

wag (8.11) współczynniki h_k wynoszą 0 w każdym przypadku. Po przekształceniu zależności (8.12) otrzymujemy $\mathbf{q}(\mathbf{y})$ oraz z (8.13) także $\mathbf{y}(\mathbf{z})$. W zmiennych uprzywilejowanych model jednokołowca (A.3) przyjmuje postać

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} = \begin{bmatrix} \cos z_2 \\ 0 \\ -\sin z_2 \end{bmatrix} \mathbf{u}_1 + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \mathbf{u}_2 = \tilde{\mathbf{g}}_1(\mathbf{z})\mathbf{u}_1 + \tilde{\mathbf{g}}_2(\mathbf{z})\mathbf{u}_2$$

i z dokładnością do dyfeomorfizmu jest równoważny modelowi (A.3). Rozwinięcie w szereg Taylora poszczególnych pól wektorowych wiedzie do zależności

$$\tilde{\mathbf{g}}_1(\mathbf{z}) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -z_2 \end{bmatrix} + \frac{1}{2!} \begin{bmatrix} -z_2^2 \\ 0 \\ 0 \end{bmatrix} + \frac{1}{3!} \begin{bmatrix} 0 \\ 0 \\ z_2^3 \end{bmatrix} + \dots, \quad \tilde{\mathbf{g}}_2(\mathbf{z}) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

Zgodnie z Krokiem 7 i zależnością (8.11), dwie pierwsze składowe pól muszą mieć stopień 0, a trzecia stopień równy 1. Stopień zmiennej z_1, z_2 wynosi 1, a zmiennej z_3 jest równy 2. Zatem po eliminacji jednomianów nie spełniających podanych warunków (między innymi z_2^2 z pierwszej składowej i z_3^2 ze składowej trzeciej) układ aproksymujący przyjmuje postać

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -z_2 \end{bmatrix} \mathbf{u}_1 + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \mathbf{u}_2 = \hat{\mathbf{g}}_1(\mathbf{z})\mathbf{u}_1 + \hat{\mathbf{g}}_2(\mathbf{z})\mathbf{u}_2. \quad (8.14)$$

Przykład 20. Nilpotentna aproksymacja trójkołowego robota mobilnego.

Czterowymiarowy model tytułowego układu jest zadany równaniem (A.4), a punktem w którym układ jest aproksymowany będzie $\mathbf{q}_a = (q_{a1}, q_{a2}, q_{a3}, q_{a4})^\top$. Oprócz generatorów układu (A.4) pozostałe elementy bazy Halla algebry Liego gwarantujące spełnienie warunku rzędu w każdej konfiguracji to $[\mathbf{g}_1, \mathbf{g}_2] = (-s_{q3}s_{q4}, c_{q3}s_{q4}, -c_{q4}, 0)^\top$ oraz $[\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]] = (-c_{q3}, -s_{q3}, 0, 0)^\top$. Wektor wag i wzrostu wynoszą więc

$$\boldsymbol{\omega} = (1, 1, 2, 3)^\top, \quad \mathbf{k} = (2, 3, 4)^\top. \quad (8.15)$$

Współrzędne lokalne \mathbf{y} , zgodnie ze wzorem (8.4) określa zależność

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} -s_{q3}c_{q4} & c_{q3}c_{q4} & s_{q4} & 0 \\ 0 & 0 & 0 & 1 \\ -s_{q3}s_{q4} & c_{q3}s_{q4} & -c_{q4} & 0 \\ -c_{q3} & -s_{q3} & 0 & 0 \end{bmatrix} \begin{bmatrix} q_1 - q_{a1} \\ q_2 - q_{a2} \\ q_3 - q_{a3} \\ q_4 - q_{a4} \end{bmatrix}. \quad (8.16)$$

Także i w tym przykładzie macierz $\bar{\mathbf{G}}$ jest ortogonalna, zatem $\bar{\mathbf{G}}^{(-1)} = \bar{\mathbf{G}}^\top$. W postaci rozwiniętej zależność (8.16) jest następująca

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} (q_2 - q_{a2})c_{qa3}c_{qa4} - (q_1 - q_{a1})s_{qa3}c_{qa4} + (q_3 - q_{a3})s_{qa4} \\ q_4 - q_{a4} \\ -(q_3 - q_{a3})c_{qa4} + (q_2 - q_{a2})c_{qa3}s_{qa4} - (q_1 - q_{a1})s_{qa3}s_{qa4} \\ -(q_1 - q_{a1})c_{qa3} - (q_2 - q_{a2})s_{qa3} \end{bmatrix}. \quad (8.17)$$

Następnie wykorzystując wzór (8.6) otrzymujemy

$$\begin{cases} z_i = y_i, & i \in \{1, 2, 3\}, \\ z_4 = y_4 - \frac{1}{2}y_1^2\mathbf{g}_1^2(y_4(\mathbf{q}_a)) - y_1y_2\mathbf{g}_1(\mathbf{g}_2(y_4(\mathbf{q}_a))) - \frac{1}{2}y_2^2\mathbf{g}_2^2(y_4(\mathbf{q}_a)). \end{cases} \quad (8.18)$$

Za trzy składowe elementu h_2 ze wzoru (8.7), są odpowiedzialne trzy wektory dopuszczalne α : $(2, 0, 0, 0)^T$, $(1, 1, 0, 0)^T$, $(0, 2, 0, 0)^T$. Generatory układu zapisane w postaci operatorów różniczkowych jako

$$\mathbf{g}_1 = -s_{q_3}c_{q_4}\frac{\partial}{\partial q_1} + c_{q_3}c_{q_4}\frac{\partial}{\partial q_2} + s_{q_4}\frac{\partial}{\partial q_3}, \quad \mathbf{g}_2 = \frac{\partial}{\partial q_4},$$

umożliwiają wyliczenie ich akcji na funkcjach – argumentach. Korzystając dodatkowo z (8.17) można wyliczyć $\mathbf{g}_1(\mathbf{g}_2(y_4)) = \mathbf{g}_2(\mathbf{g}_1(y_4)) = 0$ oraz $\mathbf{g}_1(\mathbf{g}_1(y_4)) = (c_{q_3}c_{q_4} + s_{q_3}c_{q_4}s_{q_3})s_{q_4}$. Po uwzględnieniu powyższych zależności i wartościowaniu (8.18) w punkcie aproksymacji otrzymujemy

$$\begin{cases} z_i = y_i, & i \in \{1, 2, 3\}, \\ z_4 = y_4 - \frac{1}{2}y_1^2c_{q_4}s_{q_4}, \end{cases} \quad (8.19)$$

Układ (A.4) we współrzędnych uprzywilejowanych jest następujący

$$\dot{\mathbf{z}} = \begin{bmatrix} F_1(z_1, z_2, z_3) \\ 0 \\ F_2(z_1, z_2, z_3) \\ F_3(z_1, z_2, z_3) \end{bmatrix} \mathbf{u}_1 + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \mathbf{u}_2 = \tilde{\mathbf{g}}_1(\mathbf{z})\mathbf{u}_1 + \tilde{\mathbf{g}}_2(\mathbf{z})\mathbf{u}_2 \quad (8.20)$$

gdzie

$$\begin{aligned} F_1(z_1, z_2, z_3) &= c_{q_3}c_{q_4}c_{q_4+z_2}c_F + s_{q_4}c_{q_4+z_2} + c_{q_4}c_{q_4+z_2}s_{q_3}s_F, \\ F_2(z_1, z_2, z_3) &= c_{q_3}c_{q_4+z_2}c_Fs_{q_4} - c_{q_4}c_{q_4+z_2} + c_{q_4+z_2}s_{q_3}s_{q_4}s_F, \\ F_3(z_1, z_2, z_3) &= -z_1c_{q_4}s_{q_4}^2s_{q_4+z_2} - c_{q_4+z_2}(c_F(s_{q_3} + z_1c_{q_3}c_{q_4}^2s_{q_4}) \\ &\quad + (-c_{q_3} + z_1c_{q_4}^2s_{q_3}s_{q_4})s_F), \end{aligned}$$

z następującymi oznaczeniami $F = (q_3 - z_3c_{q_4} + z_1s_{q_4})$, $c_F = \cos(F)$, $s_F = \sin(F)$ oraz $c_{q_4+z_2} = \cos(q_4 + z_2)$. Współrzędne z_1 oraz z_2 mają stopień 1, zatem stopień pierwszej współrzędnej pola $\hat{\mathbf{g}}_1(\mathbf{z})$ musi być równy $\omega_1 - 1 = 0$. Warunek ten w rozwinięciu Taylora w punkcie 0 jest spełniony wyłącznie dla pierwszej składowej F_1 . Zatem pierwsza współrzędna $\hat{\mathbf{g}}_1(\mathbf{z})$ wynosi $F_1(z_1, z_2, z_3)|_0 = c_{q_3}^2c_{q_4}^2 + s_{q_3}^2c_{q_4}^2 + s_{q_4}^2 = 1$. Podobnie druga składowa $\hat{\mathbf{g}}_1(\mathbf{z})$ ma wartość 0. Trzecia składowa $\hat{\mathbf{g}}_1(\mathbf{z})$ ma mieć stopień równy 1 i jest opisana za pomocą pochodnych współrzędnych z , których stopień jest równy 1. Ostatecznie trzecia składowa jest równa $\frac{\partial F_2(z_1, z_2, z_3)}{\partial z_1}|_0z_1 + \frac{\partial F_2(z_1, z_2, z_3)}{\partial z_2}|_0z_2 = -z_2$. Ostatnia składowa $\hat{\mathbf{g}}_1(\mathbf{z})$ musi mieć stopień równy 2, więc jest równa $\frac{\partial F_3(z_1, z_2, z_3)}{\partial z_3}|_0z_3 +$

$\frac{\partial^2 F_3(z_1, z_2, z_3)}{\partial z_2^2} \Big|_0 z_2^2 + \frac{\partial^2 F_3(z_1, z_2, z_3)}{\partial z_1^2} \Big|_0 z_1^2 + \frac{\partial^2 F_3(z_1, z_2, z_3)}{\partial z_1 z_2} \Big|_0 z_1 z_2 = -z_3 c_{q\alpha}^2 - z_1 z_2 s_{q\alpha}^2$. Wylczenie pola $\hat{\mathbf{g}}_2$ jest trywialne, a układ aproksymujący przyjmuje postać

$$\dot{\mathbf{z}} = \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -z_2 \\ -z_3 c_{q\alpha}^2 - z_1 z_2 s_{q\alpha}^2 \end{bmatrix} \mathbf{u}_1 + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \mathbf{u}_2 = \hat{\mathbf{g}}_1(\mathbf{z}) \mathbf{u}_1 + \hat{\mathbf{g}}_2(\mathbf{z}) \mathbf{u}_2. \quad (8.21)$$

8.2. Miary dokładności nilpotentnej aproksymacji

Do oceny skuteczności nilpotentnej aproksymacji układu oryginalnego należy wprowadzić pewne miary. W tym celu obydwie układy (aproksymowany we współrzędnych \mathbf{q} i aproksymujący – we współrzędnych \mathbf{z}) powinny być wyrażone w tym samym układzie współrzędnych. Konsekwentnie pola wektorowe układu oryginalnego (2.36) powinny być przekształcone dyfeomorfizmem $\mathbf{q} = \varphi_{\mathbf{q}\alpha}^{-1}(\mathbf{z})$ do przestrzeni \mathbb{Z} i wtedy porównywane z polami wektorowymi zdefiniowanymi przez (8.10). Alternatywnie, pola wektorowe układu (8.10) przekształcone dyfeomorfizmem $\mathbf{z} = F_2(F_1(\mathbf{q})) = \varphi_{\mathbf{q}\alpha}(\mathbf{q})$ do \mathbb{Q} . Powstają zatem dwie lokalne miary, wokół punktu $\mathbf{q}_\alpha \in \mathbb{Q}$

$$m_{\mathbf{q}\alpha}(\mathbf{q}) = \sum_{i=1}^m \|\mathbf{g}_i(\mathbf{q}) - \tilde{\mathbf{g}}_i(\mathbf{q})\|, \quad \tilde{\mathbf{g}}_i(\mathbf{q}) = \left(\frac{\partial \varphi_{\mathbf{q}\alpha}(\mathbf{q})}{\partial \mathbf{q}} \right)^{-1} \hat{\mathbf{g}}_i(\varphi_{\mathbf{q}\alpha}(\mathbf{q})), \quad (8.22)$$

lub jego obrazu w przestrzeni \mathbb{Z}

$$m_{\varphi_{\mathbf{q}\alpha}(\mathbf{q}\alpha)}(\mathbf{z}) = \sum_{i=1}^m \|\tilde{\mathbf{g}}_i(\mathbf{z}) - \hat{\mathbf{g}}_i(\mathbf{z})\|, \quad \tilde{\mathbf{g}}_i(\mathbf{z}) = \left(\frac{\partial \varphi_{\mathbf{q}\alpha}^{-1}(\mathbf{z})}{\partial \mathbf{z}} \right)^{-1} \mathbf{g}_i(\varphi_{\mathbf{q}\alpha}^{-1}(\mathbf{z})). \quad (8.23)$$

Dowolna norma może być wykorzystana w zależnościach (8.22), (8.23), lecz preferowaną jest ważona norma euklidesowa. Dla wektora dodatnich wag $\mathbf{w} = (w_1, \dots, w_m)^T$ normę tę zadaje wzór $\|\mathbf{a}\| = (a_1, \dots, a_m)^T = \sqrt{\sum_{i=1}^m w_i a_i^2}$. Dla (8.22) wagi odzwierciedlają różnice w zakresie bądź typie współrzędnych (położeniowe, kątowe) wektora \mathbf{q} . Natura ważonej normy euklidesowej w (8.23) jest nieco inna. Kolejne współrzędne w wektorze \mathbf{z} odpowiadają warstwom pól wektorowych z bazy Halla. I tak pierwszych m współrzędnych odpowiada ruchom w pierwszej warstwie – energetycznie efektywnych przemieszczeń. Dla przykładu wygenerowanie ruchu $T\hat{\mathbf{g}}_1$ na interwale długości T wymaga włączenia sterowania pierwszego i wyłączenia pozostałych, a zużycie energii wynosi $\int_0^T 1 dt = T$ ($u_1 = 1$). Dla następnych $m(m-1)/2$ współrzędnych (odpowiadających drugiej warstwie) scenariusz zastosowanych sterowań jest bardziej skomplikowany. Przykładowo, do generacji ruchu $T[\hat{\mathbf{g}}_1, \hat{\mathbf{g}}_2]$ z wykorzystaniem formuły CBHD [86] przykładowe sterowania złożone z czterech interwałów długości \sqrt{T} są następujące: $\mathbf{u}_1 = (+1, 0, -1, 0)^T$, $\mathbf{u}_2 = (0, +1, 0, -1)^T$, a całkowita energia sterowań wynosi

$4\sqrt{T}$. Ponieważ formuła CBHD jest ważna dla krótkich interwałów czasowych, zatem $T \ll 4\sqrt{T}$.

Miary (8.22), (8.23) spełniają minimalne wymagania dla dobrze zdefiniowanych miar, bowiem osiągają wartość zero, gdy oryginalny układ bezdryfowy jest nilpotentny oraz wartość dodatnią dla układów nie będących nilpotentnymi. W definicjach miar (8.22), (8.23), porównywane są jedynie generatory (lub ich obrazy). Oczywiście miary mogą być rozszerzone na pola wyższych stopni. W tym przypadku sumowanie w zależnościach (8.22), (8.23) powinno zostać rozszerzone z m do r , por. układ rozszerzony (3.1). Nie można jednoznacznie określić, która z miar jest lepsza, gdyż obie posiadają zalety i wady. Miara (8.22) określa jakość aproksymacji w naturalnej przestrzeni stanu, zatem posiada fizyczną interpretację zmiennych. Z drugiej strony, miara (8.23), opisana w przestrzeni \mathbb{Z} posiada współrzędne odzwierciedlające trudność ruchu w ich kierunku. Obie miary współdzielą pewien mankament, nie biorą pod uwagę kierunku planowanego ruchu z punktu $\mathbf{q}_a \in \mathbb{Q}$. Ta własność jest szczególnie ważna w zadaniach planowania ruchu, gdy interesujące są nie wszystkie, ale jeden (lub niewiele) wyróżniony kierunek ruchu. Z tego powodu konieczna jest kolejna miara unikająca powyższej wady.

Niech wyróżnionym będzie kierunek ruchu ku zadanemu punktowi docelowemu \mathbf{q}_f , a bieżącą konfiguracją (w której odbywa się aproksymacja) jest \mathbf{q}_a . Po nilpotentnej aproksymacji układ (2.36) ma swój nilpotentny odpowiednik dany przez (8.10). Jeśli teraz, w ustalonym przedziale czasu równym T , zastosujemy algorytm LS do przeprowadzenia układu (8.10) między punktem $\mathbf{z}_a = \varphi_{\mathbf{q}_a}(\mathbf{q}_a) = \mathbf{0}$ i $\mathbf{z}_f = \varphi_{\mathbf{q}_a}(\mathbf{q}_f)$, a wynikowe sterowania $\mathbf{u}(\cdot)$ zastosujemy do układu (2.36) zainicjowanego w punkcie \mathbf{q}_a , otrzymujemy pewną konfigurację końcową $\mathbf{q}(T)$. Na podstawie tej konfiguracji definiujemy następującą miarę pozycyjną

$$M(\mathbf{q}_a, \mathbf{q}_f) = \frac{\|\mathbf{q}(T) - \mathbf{q}_f\|}{\|\mathbf{q}_f - \mathbf{q}_a\|} \cdot 100\%, \quad (8.24)$$

która opisuje efektywność aproksymacji w planowaniu ruchu w kierunku od \mathbf{q}_a ku \mathbf{q}_f . Miara (8.24) zależy od przyjętej reprezentacji ruchu $\mathbf{S}(t)$ (4.1) w algorytmie LS oraz metody generacji sterowań. Jeśli wartość miary jest nieakceptowalnie wysoka, punkt docelowy \mathbf{q}_f powinien być przesunięty w kierunku \mathbf{q}_a . Ewidentną zaletą miary jest ocena rzeczywistego ruchu, a wadą – spore nakłady obliczeniowe na jej wyliczenie i nieuwzględnienie relacji kątowych między punktami \mathbf{q}_a , \mathbf{q}_f , $\mathbf{q}(T)$. Wartość miary dąży do zera, gdy $\mathbf{q}_f \rightarrow \mathbf{q}_a$, ponieważ dokładność aproksymacji rośnie.

8.3. Symulacje

Wyliczenie nilpotentnej aproksymacji i miar jej jakości przeprowadzono dla dwóch modeli. Pierwszym z nich jest jednokołowiec o modelu (A.3), a punkt aproksymacji to $\mathbf{q}_a = (q_{a1}, q_{a2}, q_{a3})^T \in \mathbb{Q}$. Wykorzystując oznaczenie $\mathbf{q} - \mathbf{q}_a = \mathbf{\Delta} = (\Delta_1, \Delta_2, \Delta_3)^T$

i stosując nilpotentną aproksymację, opisaną w podrozdziale 8.1, odwzorowanie $\mathbf{z} = \varphi_{\mathbf{q}_a}(\mathbf{q})$, oraz odwrotne przekształcenie $\mathbf{q} = \varphi_{\mathbf{q}_a}^{-1}(\mathbf{z})$ opisują

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} \Delta_1 \cos(q_{a3}) + \Delta_2 \sin(q_{a3}) \\ \Delta_3 \\ -\Delta_2 \cos(q_{a3}) + \Delta_1 \sin(q_{a3}) \end{bmatrix}, \quad \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} q_{a1} + z_1 \cos(q_{a3}) + z_3 \sin(q_{a3}) \\ q_{a2} - z_3 \cos(q_{a3}) + z_1 \sin(q_{a3}) \\ q_{a3} + z_2 \end{bmatrix}. \quad (8.25)$$

Nilpotenta aproksymacja dana jest wzorem (8.14). Wykorzystując odwrotne przekształcenie (8.25), układ aproksymowany jest przekształcany do przestrzeni \mathbb{Q}

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} \cos(q_{a3}) - (q_3 - q_{a3}) \sin(q_{a3}) \\ (q_3 - q_{a3}) \cos(q_{a3}) + \sin(q_{a3}) \\ 0 \end{bmatrix} \mathbf{u}_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \mathbf{u}_2 = \tilde{\mathbf{g}}_1(\mathbf{q}) \mathbf{u}_1 + \tilde{\mathbf{g}}_2(\mathbf{q}) \mathbf{u}_2, \quad (8.26)$$

oraz oryginalny układ (A.3) do przestrzeni \mathbb{Z}

$$\dot{\mathbf{z}} = \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} = \begin{bmatrix} \cos(z_2) \\ 0 \\ -\sin(z_2) \end{bmatrix} \mathbf{u}_1 + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \mathbf{u}_2 = \tilde{\mathbf{g}}_1(\mathbf{z}) \mathbf{u}_1 + \tilde{\mathbf{g}}_2(\mathbf{z}) \mathbf{u}_2. \quad (8.27)$$

Miara (8.22) bazująca na (A.3), (8.26) z wektorem wag $(1, 1, 1)^\top$ osiąga wartość

$$m_{\mathbf{q}_a}(\mathbf{q}) = \sqrt{2 + \Delta_3^2 - 2 \cos(\Delta_3) - 2\Delta_3 \sin(\Delta_3)}, \quad (8.28)$$

lub, gdy funkcje trygonometryczne w równaniu (8.28) zostaną rozwinięte w szereg Taylora wokół punktu \mathbf{q}_a (wielomiany do stopnia 4), powstaje

$$m_{\mathbf{q}_a}(\mathbf{q}) \simeq \frac{1}{2} \Delta_3^2. \quad (8.29)$$

Równania (8.28), (8.29) zależą jedynie od Δ . Podobnie miara (8.23) bazująca na (8.14), (8.27) z wektorem wag $(1, 1, 1)^\top$ jest równa

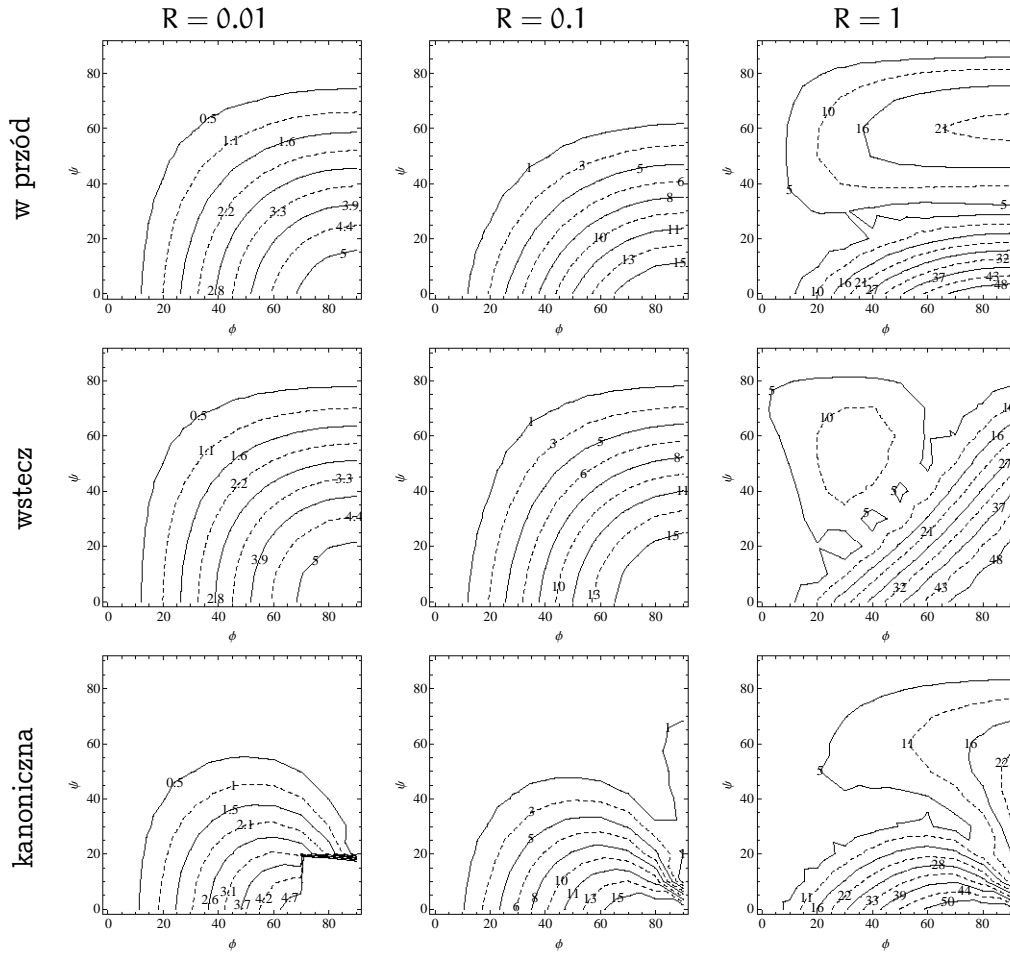
$$m_{\varphi_{\mathbf{q}_a}(\mathbf{q}_a)}(\mathbf{z}) = \sqrt{2 + z_2^2 - 2 \cos(z_2) - 2z_2 \sin(z_2)} \simeq \frac{1}{2} z_2^2. \quad (8.30)$$

By przetestować miarę (8.24) wybrano reprezentacje ruchu „wstecz”, „w przód” i kanoniczną. Dla każdej z reprezentacji wyliczono odpowiadające równania CFS postaci

$$\begin{aligned} e^{h_1 \mathbf{B}_1} e^{h_2 \mathbf{B}_2} e^{h_3 [\mathbf{B}_1, \mathbf{B}_2]} &\Rightarrow (\dot{h}_1, \dot{h}_2, \dot{h}_3)^\top = (v_1, v_2, -h_2 v_1 + v_3)^\top \\ e^{h_3 [\mathbf{B}_1, \mathbf{B}_2]} e^{h_2 \mathbf{B}_2} e^{h_1 \mathbf{B}_1} &\Rightarrow (\dot{h}_1, \dot{h}_2, \dot{h}_3)^\top = (v_1, v_2, h_1 v_2 + v_3)^\top \\ e^{h_1 \mathbf{B}_1 + h_2 \mathbf{B}_2 + h_3 [\mathbf{B}_1, \mathbf{B}_2]} &\Rightarrow (\dot{h}_1, \dot{h}_2, \dot{h}_3)^\top = (v_1, v_2, (h_1 v_2 - h_2 v_1)/2 + v_3)^\top \end{aligned} \quad (8.31)$$

zainicjalizowane w $\mathbf{h}(0) = \mathbf{0}$. Postawiono trzy zadania z różnymi punktami aproksymacji \mathbf{q}_a

$$1: \mathbf{q}_a = (5, 2, 0^\circ)^\top, \quad 2: \mathbf{q}_a = (5, 2, 45^\circ)^\top, \quad 3: \mathbf{q}_a = (5, 2, 90^\circ)^\top. \quad (8.32)$$

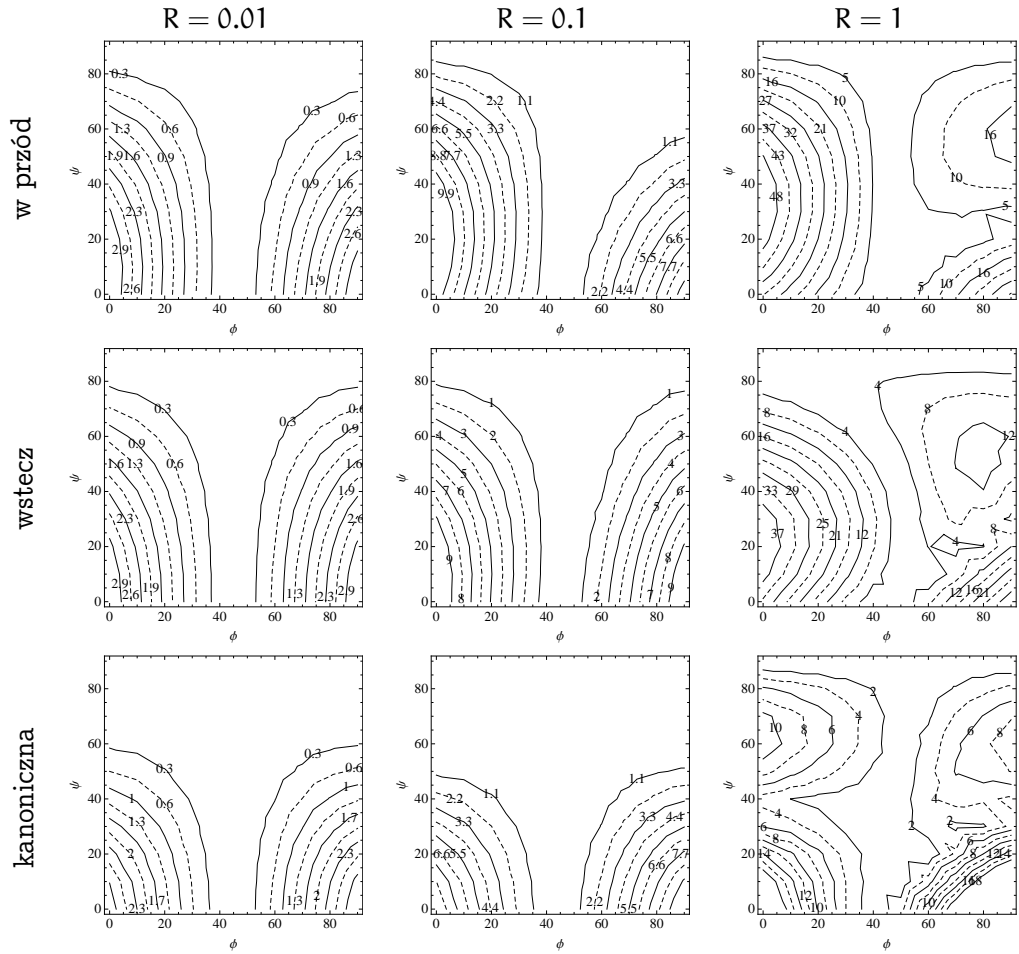


Rysunek 8.1. Zadanie 1, miara $M(\mathbf{q}_\alpha, \mathbf{q}_f)$ dla różnych reprezentacji i odległości R oraz rodziny punktów docelowych generowanych przez $\phi, \psi \in (0, 90^\circ)$, ze skokiem co 10°

Pierwsze dwie współrzędne q_1, q_2 punktu aproksymacji pozostają niezmiennie, gdyż spodziewamy się istotnej zależności jedynie od trzeciej współrzędnej, zob. (8.28). Do ustalenia punktów docelowych \mathbf{q}_f wykorzystano współrzędne sferyczne, a zbiór konfiguracji docelowych zadaje zależność $\mathbf{q}_f = \mathbf{q}_\alpha + \Delta\mathbf{q}$, gdzie

$$\Delta\mathbf{q} = (R \cos(\phi) \cos(\psi), R \cos(\phi) \sin(\psi), R \sin(\phi))^T, \quad (8.33)$$

z zasięgiem ruchu ustalonym na $R = \|\Delta\mathbf{q}\|$, oraz zmiennymi kątami ϕ, ψ . Czas ruchu ustalono na $T = 1$, a jako trajektorię referencyjną wybrano odcinek $\lambda(t) = (t/T) \cdot \mathbf{z}_f = (t/T) \cdot \varphi_{\mathbf{q}_0}(\mathbf{q}_f)$, bowiem $\mathbf{z}_0 = \varphi_{\mathbf{q}_0}(\mathbf{q}_0) = \mathbf{0}$. Do realizacji sterowań wykorzystano uogólnioną formułę gCBHD, zobacz rozdział 7. Po rozwiązaniu (8.31), ze znanymi sterowaniami rozszerzonymi $\mathbf{v}(\cdot)$, sterowania generujące ruch wzdłuż $\mathbf{B}_1 \rightarrow \hat{\mathbf{g}}_1(\mathbf{z}), \mathbf{B}_2 \rightarrow$

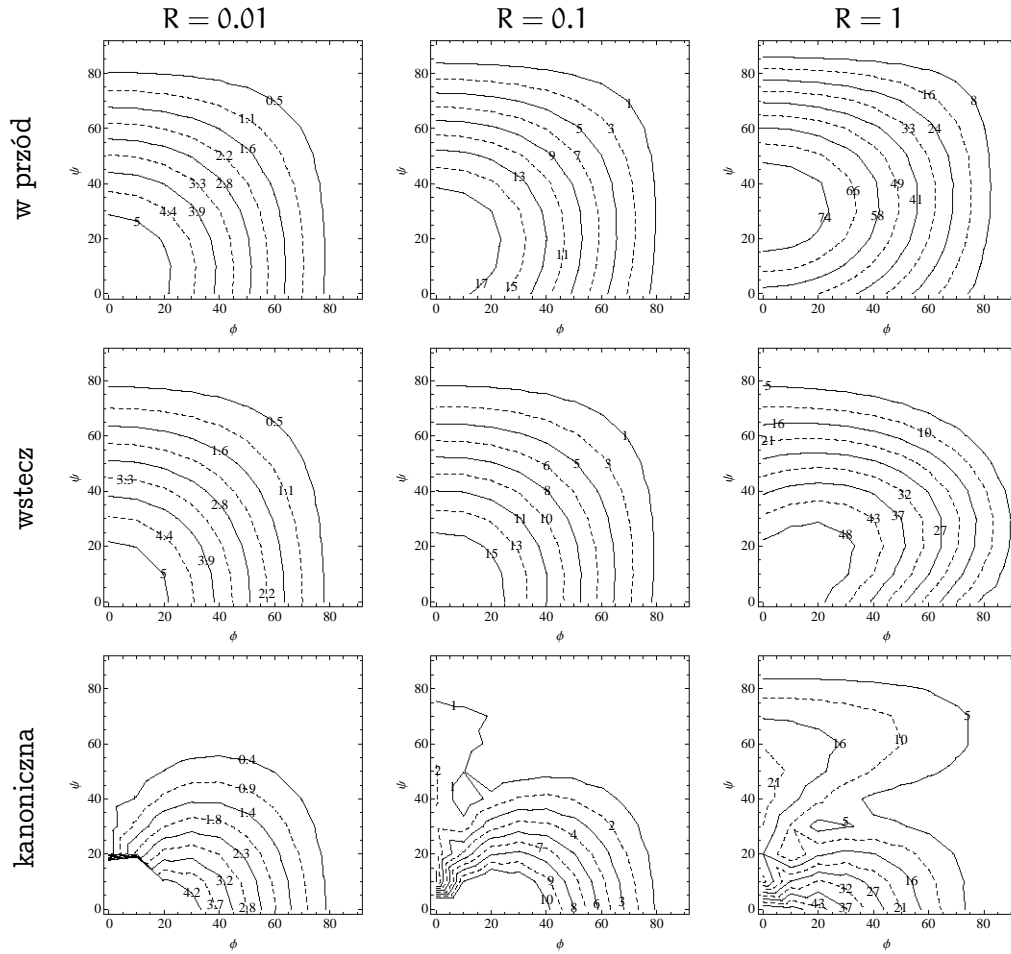


Rysunek 8.2. Zadanie 2, miara $M(\mathbf{q}_\alpha, \mathbf{q}_f)$ dla różnych reprezentacji i odległości R oraz rodziny punktów docelowych generowanych przez $\phi, \psi \in (0, 90^\circ)$, ze skokiem $\text{co } 10^\circ$

$\hat{\mathbf{g}}_2(\mathbf{z}), \mathbf{B}_3 \rightarrow [\hat{\mathbf{g}}_1, \hat{\mathbf{g}}_2](\mathbf{z})$, wybrano jako

$$\begin{aligned}
 e^{h_1(T)\mathbf{B}_1} &\Rightarrow \mathbf{u}_1(t) = h_1(T)/T', \quad \mathbf{u}_2(t) = 0, \\
 e^{h_2(T)\mathbf{B}_2} &\Rightarrow \mathbf{u}_1(t) = 0, \quad \mathbf{u}_2(t) = h_2(T)/T', \\
 e^{h_3(T)\mathbf{B}_3} &\Rightarrow \mathbf{u}_1(t) = \text{sgn}(h_3(T))\xi \cos(2\pi t/T' + \alpha), \quad \mathbf{u}_2(t) = \xi \sin(2\pi t/T' + \alpha),
 \end{aligned}
 \tag{8.34}$$

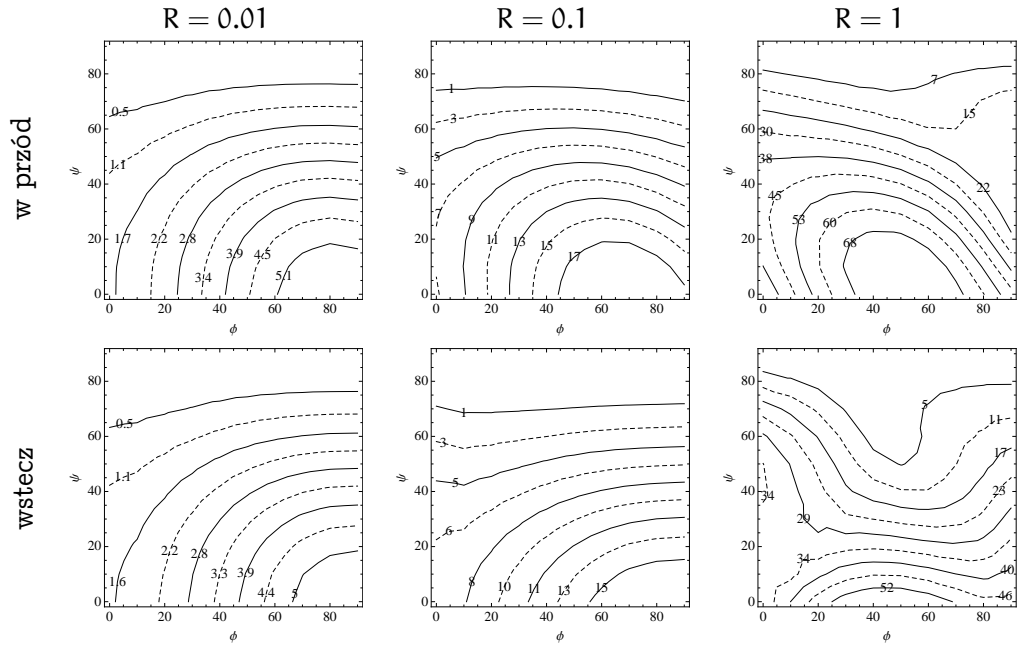
gdzie $\xi = (2/T')\sqrt{\pi|h_3(T)|}$ i $\alpha = 0$. By zapewnić wykonanie ruchu w czasie równym T , każdy segment $e^{h_1(T)\mathbf{B}_1}, e^{h_2(T)\mathbf{B}_2}, e^{h_3(T)\mathbf{B}_3}$ trwa $T' = T/3$. Na rys. 8.1-8.3 zaprezentowano symulacje dla zadań 1-3. Warstwy (linie ciągłe i kreskowane) odpowiadające tej samej wartości $M(\mathbf{q}_\alpha, \mathbf{q}_f)$ opatrzone odpowiednią wartością wyrażoną w [%]. Wydaje się, że reprezentacja tylko w niewielkim stopniu wpływa na tę miarę, natomiast istotnie jej



Rysunek 8.3. Zadanie 3, miara $M(\mathbf{p}, \mathbf{q}_f)$ dla różnych reprezentacji i odległości R , dla rodziny punktów docelowych z przedziału $\phi, \psi \in (0, 90^\circ)$, ze skokiem co 10°

wartość zależy od celu ruchu. Kierunek najmniejszego $M(\mathbf{q}_a, \mathbf{q}_f)$ jest zgodny z orientacją q_{a3} punktu \mathbf{q}_a (najmniejsze wartości miary występują dla $\phi = 0^\circ$ (rys. 8.1), dla $\phi = 45^\circ$ (rys. 8.2) i $\phi = 90^\circ$ (rys. 8.3). Dla małego zasięgu ruchu, $R = 0.01$, jakość planowania bazująca na nilpotentnej aproksymacji jest zadowalająca. Dla większych wartości R tylko ruch w wybranych kierunkach jest realizowany z względnie małym błędem. We wszystkich symulacjach (por. rys. 8.1-8.3) można zauważyć, że dla $\psi \simeq 90^\circ$ błąd jest relatywnie mały. Dla $\psi \simeq \pm 90^\circ$ ruch jednokołowego robota jest realizowany praktycznie w kierunku $[\mathbf{g}_1(\mathbf{q}_a), \mathbf{g}_2(\mathbf{q}_a)]$. Wartości współczynników $h_1(T), h_2(T)$ są bliskie zeru i ruch w kierunkach $\mathbf{g}_1(\mathbf{q}_a), \mathbf{g}_2(\mathbf{q}_a)$ nie zakłóca ruchu w kierunku $[\mathbf{g}_1(\mathbf{q}_a), \mathbf{g}_2(\mathbf{q}_a)]$.

Drugim testowanym modelem jest trójkołowy robot mobilny zadany równaniem (A.4). Nilpotentną aproksymację przeprowadzono w punkcie $\mathbf{q}_a = (q_{a1}, q_{a2}, q_{a3}, q_{a4})^T$. Transformacja zmiennych $\mathbf{z} = \varphi_{\mathbf{q}_a}(\mathbf{q})$, uzyskana na podstawie zależności (8.17), (8.19), jest



Rysunek 8.4. Zadanie 4, miara $M(\mathbf{q}_\alpha, \mathbf{q}_f)$, dla różnych reprezentacji i odległości R , dla rodziny punktów docelowych z przedziału $\phi, \psi \in (0, 90^\circ)$, ze skokiem co 10°

następująca

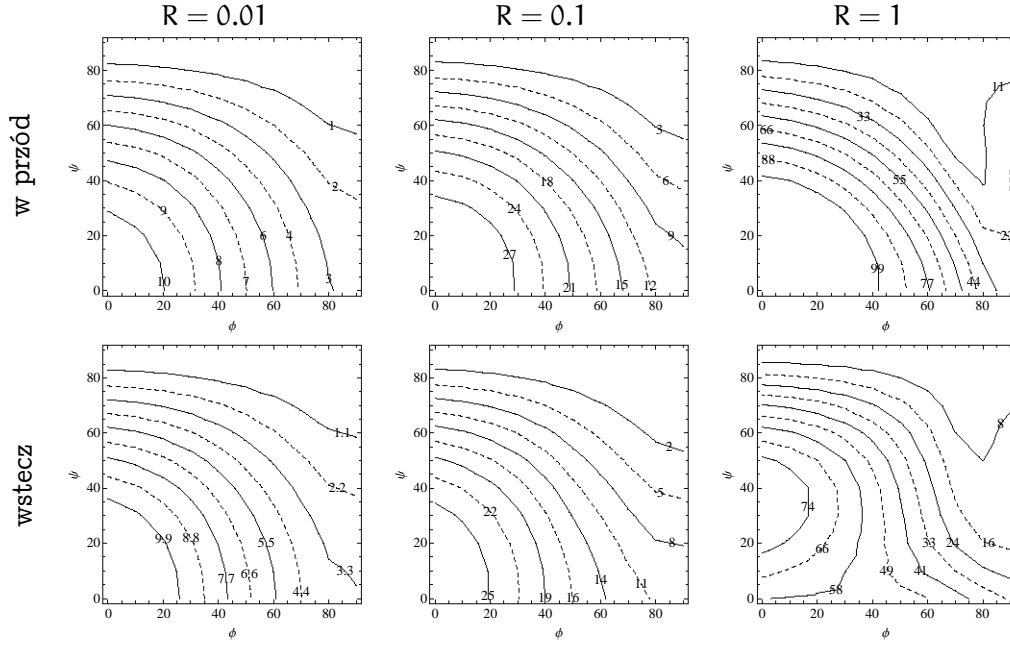
$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} c_{qa4}(\Delta_2 c_{qa3} - \Delta_1 s_{qa3}) + \Delta_3 s_{qa4} \\ \Delta_4 \\ s_{qa4}(\Delta_2 c_{qa3} - \Delta_1 s_{qa3}) - \Delta_3 c_{qa4} \\ -\Delta_1 c_{qa3} - \Delta_2 s_{qa3} - \frac{1}{4}(c_{qa4}(\Delta_2 c_{qa3} - \Delta_1 s_{qa3}) + \Delta_3 s_{qa4})^2 s_{2 \cdot qa4} \end{bmatrix}, \quad (8.35)$$

a transformacja odwrotna $\mathbf{q} = \varphi_{\mathbf{q}_\alpha}^{-1}(\mathbf{z})$ dana jest równaniem

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} q_{a1} - s_{qa3}(z_1 c_{qa4} + z_3 s_{qa4}) - c_{qa3}(z_4 + \frac{1}{4}z_1^2 s_{2 \cdot qa4}) \\ q_{a2} + c_{qa3}(z_1 c_{qa4} + z_3 s_{qa4}) - s_{qa3}(z_4 + \frac{1}{4}z_1^2 s_{2 \cdot qa4}) \\ q_{a3} + z_1 s_{qa4} - z_3 c_{qa4} \\ q_{a4} + z_2 \end{bmatrix}, \quad (8.36)$$

gdzie $s_{qa3} = \sin(q_{a3})$, $c_{qa3} = \cos(q_{a3})$, itd. Nilpotentną aproksymację trójkołowca o modelu (A.4) opisują równania (8.21), a generatory układu (A.4) przetransformowanego za pomocą (8.35) opisują równania (8.20). Natomiast model układu nilpotentnego (8.21) przekształcony do przestrzeni \mathbb{Q} posiada następujące generatory

$$\tilde{\mathbf{g}}_1(\mathbf{q}) = (w_1, w_2, s_{qa4} + \Delta_4 c_{qa4}, 0)^\top, \quad \tilde{\mathbf{g}}_2(\mathbf{q}) = (0, 0, 0, 1)^\top,$$



Rysunek 8.5. Zadanie 5, miara $M(\mathbf{q}_\alpha, \mathbf{q}_f)$, dla różnych reprezentacji i odległości R , dla rodziny punktów docelowych z przedziału $\phi, \psi \in (0, 90^\circ)$, ze skokiem co 10°

gdzie

$$\begin{aligned}
 w_1 = & -s_{q_3}c_{q_4} + \Delta_4 s_{q_3} s_{q_4} + \Delta_2 \Delta_4 c_{q_3}^2 c_{q_4} s_{q_4}^2 \\
 & + c_{q_3}(-\Delta_3 c_{q_4}^3 + c_{q_4}(-\Delta_3 + \Delta_1 \Delta_4 s_{q_3})s_{q_4}^2 + \Delta_3 \Delta_4 s_{q_4}^3), \\
 w_2 = & c_{q_3}c_{q_4} - \Delta_3 c_{q_4}^3 s_{q_3} - c_{q_4} s_{q_3}(-\Delta_3 + \Delta_1 \Delta_4 s_{q_3}) \\
 & s_{q_4}^2 + \Delta_3 \Delta_4 s_{q_3}^3 s_{q_4} + c_{q_3}(-\Delta_4 s_{q_4} + c_{q_4} \Delta_2 \Delta_4 s_{q_3} s_{q_4}^2).
 \end{aligned}$$

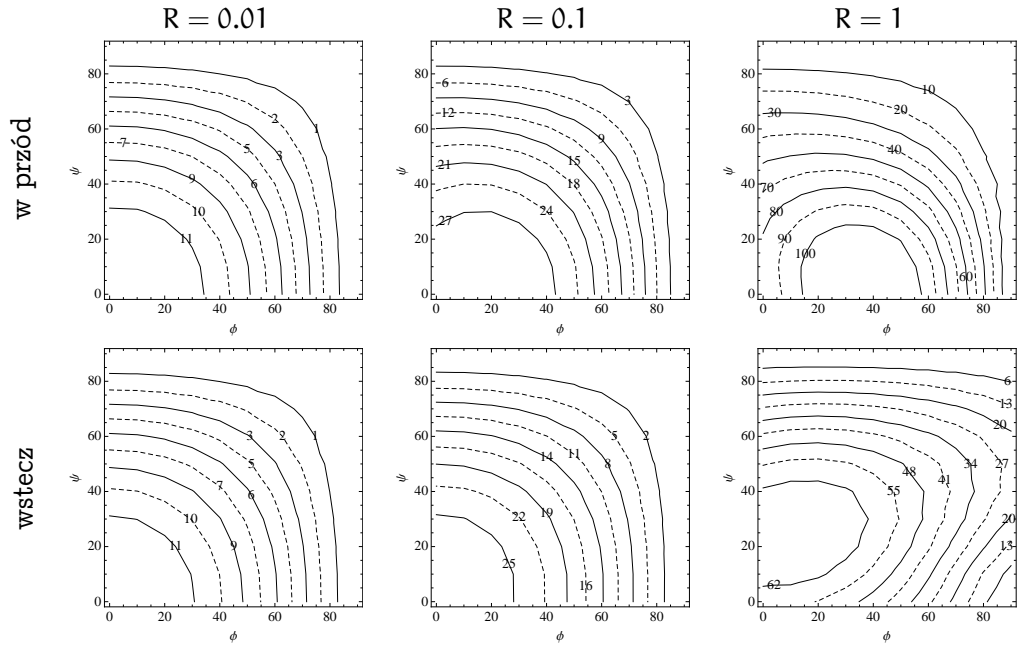
Symulacje dla trójkątowca przeprowadzono dla trzech konfiguracji \mathbf{q}_α

$$4 : \mathbf{q}_\alpha = (5, 2, 0, 0)^\top, \quad 5 : \mathbf{q}_\alpha = (5, 2, 0, \pi/4)^\top, \quad 6 : \mathbf{q}_\alpha = (5, 2, 0, \pi/2)^\top. \quad (8.37)$$

Dwie pierwsze zmienne (położeniowe) dobrano identycznie jak dla jednokołowca, natomiast zmiennej q_3 opisującej orientację przedniego koła (nie niosącej informacji o położeniu i orientacji pojazdu) przypisano wartość 0. Konfiguracje docelowe dobrano zgodnie z zależnością $\mathbf{q}_f = \mathbf{q}_\alpha + \Delta \mathbf{q}$, gdzie

$$\Delta \mathbf{q} = (R \cos(\phi) \cos(\psi), R \cos(\phi) \sin(\psi), R \sin(\phi) \cos(\psi), R \sin(\phi) \sin(\psi))^\top \quad (8.38)$$

z zadaniem $R \in \{0.01, 0.1, 1\}$. Dwie pierwsze składowe celowo wybrano identyczne, aby przedstawić system czterowymiarowy na wykresach trzywymiarowych. Rozpatrujemy tylko reprezentację „w przód” i „wstecz”, reprezentacja kanoniczna została pominięta, gdyż do wyliczenia sterowań należy użyć procedury numerycznej. Do generacji



Rysunek 8.6. Zadanie 6, miara $M(\mathbf{q}_\alpha, \mathbf{q}_f)$, dla różnych reprezentacji i odległości R , dla rodziny punktów docelowych z przedziału $\phi, \psi \in (0, 90^\circ)$, ze skokiem co 10°

$\mathbf{B}_1 \rightarrow \hat{\mathbf{g}}_1(\mathbf{z}), \mathbf{B}_2 \rightarrow \hat{\mathbf{g}}_2(\mathbf{z}), \mathbf{B}_3 \rightarrow [\hat{\mathbf{g}}_1, \hat{\mathbf{g}}_2](\mathbf{z})$, wykorzystano sterowania generujące ruch jednokołowca (8.34), ponieważ w trzeciej warstwie ruch jest generowany przez pięciosegmentowe reprezentacje, zatem należy zmodyfikować czas trwania poszczególnych sterowań: $T' = T/5$. Do przemieszczenia układu zgodnie z $\mathbf{B}_4 \rightarrow [\hat{\mathbf{g}}_1, [\hat{\mathbf{g}}_1, \hat{\mathbf{g}}_2]](\mathbf{z})$ wykorzystamy zależność (7.1), z parametrami (7.5), a dla $\mathbf{B}_5 \rightarrow [\hat{\mathbf{g}}_2, [\hat{\mathbf{g}}_1, \hat{\mathbf{g}}_2]](\mathbf{z})$, sterowania (7.6) z parametrami (7.8) (czas także należy przeskalować do $T/5$). Na rys. 8.4, 8.5, 8.6 przedstawiono wyniki symulacji. Spostrzeżenia sformułowane dla jednokołowca pozostają w mocy także dla trójkołowca.

8.4. Nilpotentne sprzężenie zwrotne

Zawartość podrozdziału bazuje na pracy Struempersa [94]. Nilpotentyzacja polega na znalezieniu bazy $\{\tilde{\mathbf{g}}_1, \dots, \tilde{\mathbf{g}}_m\}$ dla dystrybucji $\Delta = \text{span}(\mathbf{g}_1, \dots, \mathbf{g}_m)$ skojarzonej z układem bezdryfowym (2.36) takiej, że algebra Liego $L(\tilde{\mathbf{g}}_1, \dots, \tilde{\mathbf{g}}_m)$ generowana przez te pola wektorowe była nilpotentą bazą dla układu (2.36). A więc istnieje lokalnie odwracalna macierz $\mathbf{F}(\mathbf{q})$ rozmiaru $(m \times m)$, taka że transformacja $\mathbf{u} = \mathbf{F}(\mathbf{q})\mathbf{v}$ przekształca układ podstawowy (2.36) do następującej postaci

$$\dot{\mathbf{q}} = \mathbf{G}(\mathbf{q})\mathbf{F}(\mathbf{q})\mathbf{v} = \tilde{\mathbf{G}}(\mathbf{q})\mathbf{v}, \quad (8.39)$$

gdzie kolumny macierzy $\tilde{\mathbf{G}}(\mathbf{q})$ dane są przez $\tilde{\mathbf{g}}_1(\mathbf{q}), \dots, \tilde{\mathbf{g}}_m(\mathbf{q})$. Układ (8.39) jest równoważny układowi bezdryfowemu (2.36) w takim sensie, że każda trajektoria $\mathbf{q}_{\text{zad}}(\cdot)$ odpowiadająca sterowaniu $\mathbf{u}_{\text{zad}}(\cdot)$ jest lokalnie odtwarzalna za pomocą sterowania $\mathbf{v}_{\text{zad}}(\cdot) = \mathbf{F}^{-1}(\mathbf{q}_{\text{zad}}(\cdot))\mathbf{u}_{\text{zad}}(\cdot)$. Obszar zastosowania transformacji zdeterminowany jest możliwością znalezienia i odwrócenia macierzy $\mathbf{F}(\mathbf{q})$. Rozważmy dwie nilpotentyzacje jednokołowego robota mobilnego (A.3). Pierwsza zaproponowana przez Lafferriere i Sussmanna [54] o sprzężeniu

$$\begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} = \begin{bmatrix} \cos q_3 & 0 \\ 0 & \sec^2 q_3 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix},$$

które prowadzi do układu nilpotentnego postaci

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{tg } q_3 \\ 0 \end{bmatrix} \mathbf{v}_1 + \begin{bmatrix} 0 \\ 0 \\ \cos^2 q_3 \end{bmatrix} \mathbf{v}_2.$$

Drugie sprzężenie zwrotne, zaproponowane przez Giamberardino [14], jest postaci

$$\begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} = \begin{bmatrix} 1 & q_1 \sin q_3 - q_2 \cos q_3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}$$

i transformuje układ jednokołowca do postaci nilpotentnej

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} \cos q_3 \\ \sin q_3 \\ 0 \end{bmatrix} \mathbf{v}_1 + \begin{bmatrix} q_1 \sin q_3 \cos q_3 - q_2 \cos^2 q_3 \\ q_1 \sin^2 q_3 - q_2 \sin q_3 \cos q_3 \\ 1 \end{bmatrix} \mathbf{v}_2.$$

Zauważmy, że pierwsza z transformacji jest lokalna i wprowadza osobliwości do układu sterowania. Natomiast wersja Giamberardino jest globalna, ale prowadzi do bardziej skomplikowanego układu sterowania.

Nilpotentna aproksymacja może być wyliczona dla dowolnego bezdryfowego układu sterowania postaci (2.36), jednak problem wyznaczenia nilpotentnej bazy generującej dystrybucję Δ układu jest równoważny znalezieniu rozwiązania powiązanego cząstkowego równania różniczkowego. Warunki na istnienie rozwiązań cząstkowych równań różniczkowych nie są znane w ogólności, a jedynie dla szczególnych przypadków. Dla przypadku dwuwęściowego, z wykorzystaniem układu łańcuchowego jako modelu docelowego układu nilpotentnego, niezbędne warunki zostały określone za pomocą normalnej postaci Goursata [73]. Dla $m > 2$ odpowiednie warunki istnieją tylko dla $m = n - 1$, wykorzystujące twierdzenie Darboux [39], oraz pewien szczególny przypadek transformacji do układu łańcuchowego z pojedynczym łańcuchem i generatorem [10]. Jak pokazał Struemper istnieją również pewne warunki wykorzystujące grupę Liego [94], jednak by je zastosować układ bezdryfowy musi posiadać szczególne własności.

Problem znalezienia nilpotentego sprzężenia zwrotnego formułowany jest również w terminologii form różniczkowych. Dla zadanego zbioru 1-form $\{\alpha_1(\mathbf{q}), \dots, \alpha_{n-m}(\mathbf{q})\}$

anihilujących Δ należy znaleźć taką zmianę współrzędnych, że α_i gdzie $i \in \{1, \dots, n - m\}$, przybiera postać 1-form β_i , $i \in \{1, \dots, n - m\}$, charakteryzujących odpowiedni układ nilpotentny [95]. W tej notacji zostały podane warunki na nilpotentne sprzężenie zwrotne dla układu w pięciowymiarowej przestrzeni stanu o dwóch wejściach [89]. Konkludując: zadanie globalnej nilpotentyzacji dowolnego układu bezdryfowego pozostaje nadal otwarte.

9. Algorytmy dla układów nienilpotentnych

Jednym z głównych celów dysertacji jest rozszerzenie działania algorytmu LS na układy nienilpotentne. W literaturze można znaleźć wiele prób rozwiązania tego zagadnienia. Pierwszym z podejść jest wykorzystanie algorytmów nilpotentnej aproksymacji oraz nilpotentnych sprzężeń zwrotnych prezentowanych w poprzednim rozdziale. Alternatywnym rozwiązaniem jest pomysł iteracyjnego wykorzystania algorytmu LS pochodzącego z pracy [97] (przykład iteracyjnego LS [51]). We wspomnianych pracach punkty pośrednie wybierane są z linii prostej łączącej punkt startowy z końcowym. Możliwe są też inne podejścia, optymalizujące dodatkowe kryteria ruchu. Wykorzystanie w algorytmie LS jednej z formuł gCBHD lub CBHD do generacji sterowań dla układów nienilpotentnych powoduje błąd w osiągnięciu celu, ponieważ pojawiają się niezerowe pola wektorowe z wyższych warstw, które powodując dryf przeszkadzają w osiągnięciu zamierzonego celu. Aby zniwelować wpływ niepożądanych nawiasów Liego warto wybierać podcele w uprzywilejowanych kierunkach, w których jest mały dryf (wykorzystując np. sfery nieholonomiczne [91]), zerują się pewne pola, lub minimalizuje się pewne dodatkowe kryterium [19, 29]. Uwagi oraz literaturowe rozwiązania dotyczące problemu znajdują się w książce [101].

W dalszej części rozdziału zaprezentowano dwa iteracyjne algorytmy planowania ruchu. Pierwszy algorytm bazujący na metodzie LS jest wspomagany nilpotentną aproksymacją, czyli w każdym kroku (wokół kolejnej bieżącej konfiguracji) wyliczane jest nilpotentne przybliżenie układu sterowanego, a sterowanie jest generowane metodą LS. Drugi algorytm stanowi modyfikację algorytmu Lie algebraicznego [101].

9.1. Iteracyjny algorytm LS wspomagany nilpotentną aproksymacją

Algorytm planowania ruchu układów bezdryfowych z wykorzystaniem nilpotentnej aproksymacji składa się z następującej sekwencji kroków:

Algorytm 7. *wspomagany nilpotentną aproksymacją*

Krok 1 Wczytać dane wejściowe: sterowalny układ bezdryfowy (2.36), dokładność osiągnięcia celu ϵ , konfigurację początkową i końcową ruchu \mathbf{q}_0 i \mathbf{q}_f , maksymalna akceptowalna niedokładności aproksymacji δ_{\max} . Zainicjować początkową listę sterowań jako pustą i wyzerować licznik iteracji $i \leftarrow 0$. Podstawić za punkt bieżący planowania konfigurację początkową $\mathbf{q}_c \leftarrow \mathbf{q}_0$.

Krok 2 Wyznaczyć aproksymację nilpotentną układu dla dowolnego punktu przestrzeni konfiguracyjnej $\mathbf{p} \in \mathbb{Q}$. Efektem jest układ nilpotentny w przestrzeni \mathbb{Z} oraz dyfeomorfizm $\mathbf{z} = \varphi_{\mathbf{p}}(\mathbf{q})$.

Krok 3 Sprawdzić warunek końca obliczeń:

jeśli $\|\mathbf{q}_c - \mathbf{q}_f\| < \epsilon$ zakończyć obliczenia i wyprowadzić wyniki (sterowania, trajektorię, charakterystyki liczbowe: liczbę iteracji, czas ruchu, energię sterowań). W przeciwnym przypadku kontynuować krokiem 4.

Krok 4 Za dowolny punkt aproksymacji podstawić konfigurację bieżącą $\mathbf{p} \leftarrow \mathbf{q}_c$. Przetrasformować dyfeomorfizmem $\mathbf{z} = \varphi_{\mathbf{p}}(\mathbf{q})$ punkt \mathbf{q}_c do przestrzeni \mathbb{Z} , czyli $\mathbf{z}_c = \varphi_{\mathbf{q}_c}(\mathbf{q}) = \mathbf{0}$.

Krok 5 Wybrać bieżący cel ruchu \mathbf{q}_p na odcinku $\mathbf{q}_c \mathbf{q}_f$ i wyzerować licznik prób (w jednej iteracji) $j \leftarrow 0$.

Krok 6 Przetrasformować dyfeomorfizmem $\mathbf{z} = \varphi_{\mathbf{p}}(\mathbf{q})$ punkt \mathbf{q}_p do przestrzeni \mathbb{Z} otrzymując, punkt $\mathbf{z}_p = \varphi_{\mathbf{p}}(\mathbf{q}_p)$ i przeprowadzić algorytmem LS planowanie ruchu układu nilpotentnego między punktami \mathbf{z}_c i \mathbf{z}_p . Wynikiem jest sterowanie $\mathbf{u}(\cdot)$ na horyzoncie czasu T skorelowanym z długością wektora $\mathbf{q}_p - \mathbf{q}_c$.

Krok 7 Zastosować wyznaczone sterowanie do układu (2.36) inicjując stan konfiguracją \mathbf{q}_c . W wyniku całkowania numerycznego otrzymujemy punkt $\tilde{\mathbf{q}}_p$.

Krok 8 Sprawdzić, czy jakość aproksymacji jest odpowiednia: jeśli

$$\frac{\|\tilde{\mathbf{q}}_p - \mathbf{q}_p\|}{\|\mathbf{q}_c - \mathbf{q}_p\|} \in f(\delta_{\max}, \|\mathbf{q}_c - \mathbf{q}_f\|), \quad (9.1)$$

gdzie $f(\cdot, \cdot)$ jest funkcją projektanta algorytmu ustalająca przedział dopuszczalny wartości błędu aproksymacji (opisanego lewą stroną zależności (9.1)), dopisać sterowania do listy sterowań, podstawić $\mathbf{q}_c \leftarrow \tilde{\mathbf{q}}_p$, zwiększyć licznik iteracji $i \leftarrow i + 1$ i przejść do kroku 3. W przeciwnym przypadku, zmodyfikować \mathbf{q}_p , zwiększyć licznik prób $j \leftarrow j + 1$ i kontynuować krokiem 6.

Warto zauważyć, że jeśli nie istnieje jedna uniwersalna aproksymacja dla wszystkich punktów przestrzeni konfiguracyjnej (czyli w różnych punktach przestrzeni inne pola decydują o postaci aproksymacji), to procedurę nilpotentnej aproksymacji należy umieścić wewnątrz pętli (zamienić krok 3. z krokiem 2.) W trybie off-line łatwo ustalić, w jakich obszarach przestrzeni konfiguracyjnej, która z aproksymacji jest właściwa. Następnie przełączać aproksymację w zależności od obszaru do którego należy punkt bieżący \mathbf{q}_c . Modyfikację punktu próbnego \mathbf{q}_p (krok 8.) i wybór pierwszej jego wartości (krok 5.) należy traktować jako jednowymiarową optymalizację położenia punktu próbnego. Czym bliżej punkt \mathbf{q}_p znajduje się \mathbf{q}_c , tym dokładniejsza jest aproksymacja, jednak w tym przypadku liczba iteracji (a zatem złożoność) algorytmu rośnie, gdyż przemieszczenie $\|\mathbf{q}_c - \tilde{\mathbf{q}}_p\|$ w jednej pętli algorytmu jest niewielkie. Jeśli punkt \mathbf{q}_p oddala się od \mathbf{q}_c obserwujemy efekt przeciwny. Warunek akceptowalności aproksymacji (9.1) powinien dopuszczać możliwie duże wartości błędu (ograniczone od góry przez założony

parametr δ_{\max}), gdy \mathbf{q}_c jest odległe od \mathbf{q}_f i maleć do zera, gdy $\mathbf{q}_c \rightarrow \mathbf{q}_f$. Przykładowa realizacja kroku 5:

$$\mathbf{q}_p = \mathbf{q}_c + \alpha(\mathbf{q}_f - \mathbf{q}_c), \quad (9.2)$$

gdzie współczynnik $\alpha = \min(\Delta/\|\mathbf{q}_f - \mathbf{q}_c\|, 1)$, a Δ jest założoną długością przemieszczenia w kierunku celu. Przykładowa modyfikacja \mathbf{q}_p w kroku 8.: niech α_{\min} , α_{\max} oznaczają minimalną i maksymalną wartość współczynnika α o początkowych wartościach równych, odpowiednio, 0 i 1. Jeśli wartość lewej strony zależności (9.1) jest mniejsza od dolnej granicy dopuszczalnej wartości błędu aproksymacji, to $\alpha_{\min} = \alpha$. Jeśli natomiast jest większa od górnej granicy, to $\alpha_{\max} = \alpha$. Następnie wyliczana jest nowa wartość $\alpha = (\alpha_{\min} + \alpha_{\max})/2$, która determinuje, w myśl zależności (9.2), konfigurację próbną \mathbf{q}_p . Złożoność obliczeniowa algorytmu zależy od złożoności obliczeniowej realizacji planowania ruchu dla układu nilpotentnego (warto przyjąć tę wielkość jako jednostkę) pomnożoną przez liczbę wywołań tego algorytmu (zależącą od liczby zmian \mathbf{q}_p w pojedynczej iteracji (licznik j) oraz liczby iteracji (licznik i), która, z kolei, zależy od przyjętego parametru δ_{\max}). Zarówno jednokrokowy jak i całkowity czas ruchu może być przeskalowany (np. gdy sterowania mają być ograniczone). Funkcja oceny z Kroku 8 może być zdefiniowana np. w następujący sposób $f(\delta_{\max}, \|\mathbf{q}_c - \mathbf{q}_f\|) = [(1 - \kappa)\delta_{\max}, \delta_{\max}]$, gdzie współczynnik $\kappa \in [0, 1)$ determinuje dolną granicę dopuszczalnego przedziału – dopuszczalną rozbieżność od δ_{\max} zadaną w procentach ($\kappa = 10[\%]$ ustala przedział na $[90[\%]\delta_{\max}, 100[\%]\delta_{\max}]$). Drugi argument funkcji pozwala na uzależnienie wyboru przedziału od odległości od celu.

9.2. Metoda Lie algebraiczna z iteracyjnym doбором podcelów

Metoda wpisuje się w szereg metod Lie algebraicznych, a jej zapis zawiera następujące kroki.

Algorytm 8. *Lie algebraiczny z iteracyjnym doбором podcelów*

Krok 1 Wczytać dane wejściowe: sterowalny układ bezdryfowy (2.36), dokładność osiągnięcia celu ϵ , konfigurację początkową i końcową ruchu \mathbf{q}_0 i \mathbf{q}_f , maksymalna akceptowalna niedokładności ruchu jednokrokowego δ_{\max} . Zainicjować początkową listę sterowań jako pustą i licznik iteracji $i \leftarrow 0$. Podstawić za punkt bieżący planowania konfigurację początkową $\mathbf{q}_c \leftarrow \mathbf{q}_0$.

Krok 2 Wybrać reprezentację sterowań w wybranej bazie $\phi_i(t)$, $t \in [0, T]$, gdzie T jest jednokrokovym horyzontem czasu ruchu oraz rozkład sterowań $u_i(\cdot)$ w tej bazie) kodowaną wektorem, na razie, nieoznaczonych parametrów \mathbf{p}

$$u_i(t) = p_{ij}\phi_i(t), \quad i = 1, \dots, m, \quad j = 1, \dots \quad (9.3)$$

$\mathbf{p} = (p_{11}, p_{12}, \dots, p_{21}, p_{22}, \dots, p_{m1}, p_{m2}, \dots)^T$. Wyliczyć odwzorowanie typu kinematycznego $\mathbf{v} = \mathbf{k}(\mathbf{p})$, korzystając z formuły gCBHD (2.33).

Krok 3 Sprawdzić warunek końca obliczeń:

jeśli $\|\mathbf{q}_c - \mathbf{q}_f\| < \epsilon$ zakończyć obliczenia i wyprowadzić wyniki (sterowania, trajektorię, charakterystyki liczbowe: liczbę iteracji, czas ruchu, energię sterowań).
W przeciwnym przypadku kontynuować krokiem 4.

Krok 4 Wybrać bieżący cel ruchu \mathbf{q}_p na odcinku $\mathbf{q}_c\mathbf{q}_f$ i wyzerować licznik prób (w jednej iteracji) $j \leftarrow 0$.

Krok 5 Wyrazić kierunek do celu jako kombinację pól bazowych zwartościowanych w konfiguracji bieżącej

$$\mathbf{q}_p - \mathbf{q}_c = \mathbf{g}_1(\mathbf{q}_c)\mathbf{v}_1 + \cdots + \mathbf{g}_m(\mathbf{q}_c)\mathbf{v}_m + \mathbf{g}_{m+1}(\mathbf{q}_c)\mathbf{v}_{m+1} + \cdots + \mathbf{g}_r(\mathbf{q}_c)\mathbf{v}_r = \mathbf{G}_{\text{ext}}(\mathbf{q}_c)\mathbf{v}_c \quad (9.4)$$

gdzie użyto oznaczenia \mathbf{v}_c dla podkreślenia, że współczynniki są istotnie zależne od bieżącej konfiguracji. Wyznaczyć współczynniki kombinacji \mathbf{v}_c korzystając z pseudoodwrotności Moora-Penrose'a.

$$\mathbf{v}_c = \mathbf{G}_{\text{ext}}^{\#}(\mathbf{q}_c)(\mathbf{q}_p - \mathbf{q}_c). \quad (9.5)$$

Krok 6 Korzystając z odwzorowania typu kinematycznego, wyliczyć algorytmem Newtona rozwiązania zadania kinematyki odwrotnej wartość parametrów

$$\mathbf{p}_{i+1} = \mathbf{p}_i + \left(\frac{\partial \mathbf{k}}{\partial \mathbf{p}}\right)^{\#}(\mathbf{p}_i)\mathbf{v}_c. \quad (9.6)$$

gdzie \mathbf{p}_0 jest wybranym, początkowym zestawem parametrów. Wynikiem jest zestaw parametrów \mathbf{p}_c , który determinuje sterowanie $\mathbf{u}(\cdot)$ na horyzoncie czasu T.

Krok 7 Zastosować wyznaczone sterowanie do układu (2.36) inicjując stan konfiguracją \mathbf{q}_c . W wyniku całkowania numerycznego otrzymujemy punkt $\tilde{\mathbf{q}}_p$.

Krok 8 Sprawdzić, czy jakość uzyskanego wyniku jest odpowiednia: jeśli

$$\frac{\|\tilde{\mathbf{q}}_p - \mathbf{q}_p\|}{\|\mathbf{q}_c - \mathbf{q}_p\|} \in f(\delta_{\text{max}}, \|\mathbf{q}_c - \mathbf{q}_f\|), \quad (9.7)$$

gdzie $f(\cdot, \cdot)$ jest funkcją projektanta algorytmu ustalającą przedział dopuszczalny wartości błędu jakości przybliżenia (opisanego lewą stroną zależności (9.1)), dopisać sterowania do listy sterowań, podstawić $\mathbf{q}_c \leftarrow \tilde{\mathbf{q}}_p$, zwiększyć licznik iteracji $i \leftarrow i+1$ i przejść do kroku 3. W przeciwnym przypadku, zmodyfikować \mathbf{q}_p , zwiększyć licznik prób $j \leftarrow j+1$ i kontynuować od kroku 5.

Zauważmy formalne podobieństwo Algorytmu 8 do Algorytmu 7. Podobny jest sposób poprawy punktu próbnego, główna idea polegająca na wyznaczaniu pożądaných parametrów ruchu (współczynniki Halla lub \mathbf{v}_c) oraz realizacja tych współczynników przy pomocy sterowań. Natomiast jedną z różnic jest fakt, że w Algorytmie 8 nie otrzymuje się sterowań rozszerzonych $\mathbf{v}(\cdot)$, lecz wektor współczynników \mathbf{v}_c . Przedstawiona wersja algorytmu odpowiada przyjęciu reprezentacji kanonicznej w algorytmie

LS, bowiem z definicji gCBHD generowany jest ruch postaci: $e^{v_1 \mathbf{g}_1 + v_2 \mathbf{g}_2 + \dots + v_1 \mathbf{g}_r}$. Złożoność obliczeniowa Algorytmu 8 jest porównywalna z wersją reprezentacji kanonicznej Algorytmu 7, bowiem w jednym i drugim przypadku zachodzi konieczność dodatkowego procesu optymalizacyjnego (algorytm Newtona) w celu wyznaczenia parametrów sterowań. Wyznaczenie kinematyki $\mathbf{k}(\mathbf{p})$ dla wybranej reprezentacji sterowań jest dokonywane w trybie off-line. Istnieje możliwość przyjęcia innej niż kanoniczna reprezentacji ruchu, gdyż istnieje dyfeomorfizm między reprezentacjami. Przykładową zamianę zilustrowano przykładem 21.

Przykład 21. Niech zadaniem będzie wyznaczenie dla układu dwuwęściowego o generatorach $\mathbf{g}_1, \mathbf{g}_2$ rozpatrywanego do warstwy drugiej włącznie i zadanej reprezentacji kanonicznej ruchu $e^{h_1 \mathbf{g}_1 + h_2 \mathbf{g}_2 + h_3 \mathbf{g}_3}$ ze znanymi współczynnikami $\mathbf{h} = (h_1, h_2, h_3)^T$ (w chwili końcowej planowania $t = T$) równoważnych (do warstwy drugiej włącznie) reprezentacji „w przód” $e^{\bar{h}_1 \mathbf{g}_1} e^{\bar{h}_2 \mathbf{g}_2} e^{\bar{h}_3 \mathbf{g}_3}$ i „wstecz” $e^{\underline{h}_3 \mathbf{g}_3} e^{\underline{h}_2 \mathbf{g}_2} e^{\underline{h}_1 \mathbf{g}_1}$, gdzie $\mathbf{g}_3 = [\mathbf{g}_1, \mathbf{g}_2]$.

Dla reprezentacji „w przód” korzystamy dwukrotnie z formuły CBHD (4.7) obcinając pola wyższych stopni niż 2:

$$\begin{aligned} e^{\bar{h}_1 \mathbf{g}_1} e^{\bar{h}_2 \mathbf{g}_2} e^{\bar{h}_3 \mathbf{g}_3} &= e^{\bar{h}_1 \mathbf{g}_1 + \bar{h}_2 \mathbf{g}_2 + \frac{1}{2} \bar{h}_1 \bar{h}_2 [\mathbf{g}_1, \mathbf{g}_2]} e^{\bar{h}_3 \mathbf{g}_3} = \\ e^{\bar{h}_1 \mathbf{g}_1 + \bar{h}_2 \mathbf{g}_2 + \frac{1}{2} \bar{h}_1 \bar{h}_2 [\mathbf{g}_1, \mathbf{g}_2] + \bar{h}_3 \mathbf{g}_3 + \frac{1}{2} [\bar{h}_1 \mathbf{g}_1 + \bar{h}_2 \mathbf{g}_2 + \frac{1}{2} \bar{h}_1 \bar{h}_2 [\mathbf{g}_1, \mathbf{g}_2], \bar{h}_3 \mathbf{g}_3]} &= \\ e^{\bar{h}_1 \mathbf{g}_1 + \bar{h}_2 \mathbf{g}_2 + \frac{1}{2} \bar{h}_1 \bar{h}_2 [\mathbf{g}_1, \mathbf{g}_2] + \bar{h}_3 \mathbf{g}_3} &= e^{\bar{h}_1 \mathbf{g}_1 + \bar{h}_2 \mathbf{g}_2 + (\frac{1}{2} \bar{h}_1 \bar{h}_2 + \bar{h}_3) \mathbf{g}_3}, \end{aligned}$$

a po porównaniu współczynników pól $\mathbf{g}_1, \mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]$ reprezentacji kanonicznej i „w przód” otrzymujemy

$$\bar{h}_1 = h_1, \quad \bar{h}_2 = h_2, \quad \bar{h}_3 = h_3 - \frac{1}{2} h_1 h_2.$$

Podobnie dla reprezentacji „wstecz” (z pominięciem szczegółów obliczeniowych)

$$\underline{h}_1 = h_1, \quad \underline{h}_2 = h_2, \quad \underline{h}_3 = h_3 + \frac{1}{2} h_1 h_2.$$

9.3. Symulacje

Aby zbadać wpływ maksymalnej akceptowalnej niedokładności δ_{\max} osiągnięcia kolejnych podcelów na generowane trajektorie algorytmem 7 oraz 8, wykonano serie symulacji dla jednokołowca (A.3) o następujących danych wejściowych: dopuszczalny błąd osiągnięcia celu $\epsilon = 0.01$, czas pojedynczego planowania osiągnięcia podcelu $T = 1$, punkty brzegowe planowania $\mathbf{q}_0 = (5.0, 2.0, 0)^T$, $\mathbf{q}_f = (10.0, 20.0, \pi/2)^T$. Rozważono trzy reprezentacje ruchu: „w przód”, „wstecz” i kanoniczną, a parametr δ_{\max} zmieniano w zakresie od 10[%] do 100[%] z krokiem 5[%]. Rejestrowanymi charakterystykami wyników algorytmów są: całkowita liczba iteracji realizacji zadania planowania ruchu i, wraz z sumaryczną liczbą iteracji próbnych j (dla jednej iteracji

Tablica 9.1. Wartości energii (E), energii znormalizowanej ($E_{\tilde{T}=100}$) względem ustalonego czasu ruchu $\tilde{T} = 100$ oraz liczba iteracji algorytmu 7 dla różnych wartości δ_{\max} i reprezentacji dla modelu jednokołowca

δ_{\max} [%]	„w przód”			„wstecz”			kanoniczna		
	energia		iteracji	energia		iteracji	energia		iteracji
	E	$E_{\tilde{T}=100}$	$i_{\sum j}$	E	$E_{\tilde{T}=100}$	$i_{\sum j}$	E	$E_{\tilde{T}=100}$	$i_{\sum j}$
10	430.57	710.37	165 ₆₆₁	416.54	683.13	164 ₆₂₇	194.10	357.14	184 ₇₅₅
15	504.28	393.34	78 ₂₈₉	405.56	304.17	75 ₃₅₅	201.13	181.02	90 ₄₄₇
20	565.85	265.95	47 ₃₆₇	436.84	183.47	42 ₃₆₉	208.92	114.90	55 ₄₆₁
25	534.87	160.46	30 ₂₀₄	455.89	127.65	28 ₂₁₅	219.68	83.48	38 ₂₇₆
30	613.06	141.00	23 ₁₃₉	475.63	99.88	21 ₁₄₈	230.13	66.74	29 ₁₉₈
35	590.39	100.37	17 ₉₇	513.99	92.52	18 ₁₀₈	239.59	55.11	23 ₁₅₁
40	608.88	91.33	15 ₇₄	552.04	82.81	15 ₈₁	246.10	49.22	20 ₁₁₄
45	645.36	83.90	13 ₅₇	484.29	48.43	10 ₅₈	256.29	46.13	18 ₉₉
50	708.35	85.00	12 ₅₁	661.14	79.34	12 ₅₈	270.47	43.28	16 ₈₃
55	736.39	81.00	11 ₄₂	711.20	78.23	11 ₅₂	285.72	42.86	15 ₇₀
60	797.16	87.69	11 ₃₈	783.66	86.20	11 ₄₆	307.12	43.00	14 ₇₁
65	910.76	100.18	11 ₄₃	864.51	86.45	10 ₄₅	326.02	42.38	13 ₆₃
70	926.98	92.70	10 ₃₅	722.57	50.58	7 ₃₅	344.02	44.72	13 ₅₈
75	1081.77	108.18	10 ₃₈	970.71	87.36	9 ₃₆	380.43	49.46	13 ₅₅
80	1461.56	160.77	11 ₄₄	899.74	62.98	7 ₃₆	440.19	57.22	13 ₅₈
85	1945.89	155.67	8 ₄₀	971.46	68.00	7 ₃₄	426.80	46.95	11 ₅₀
90	2001.55	160.12	8 ₄₁	1099.95	77.00	7 ₃₄	541.02	70.33	13 ₅₂
95	2034.16	162.73	8 ₄₅	1110.02	77.70	7 ₃₅	677.68	81.32	12 ₅₇
100	2051.67	164.13	8 ₄₆	1132.11	56.61	5 ₂₄	789.91	118.49	15 ₅₉

pętli głównych algorytmów może być wykonywanych wiele iteracji próbnych dobierających każdorazowo \mathbf{q}_p). Ze względu na ustalony czas realizacji pojedynczego podcelu ($T = 1$), liczba iteracji i opisuje całkowity czas trwania ruchu. Monitorowana jest także energia sterowań $E(\mathbf{u}(\cdot)) = \int_0^{i \cdot T} (u_1^2(t) + u_2^2(t)) dt$. Ponieważ sterowania różnią się czasem trwania zależnym od iloczynu czasu T i liczby iteracji (i), dlatego dla lepszego uwydatnienia różnic między generowanymi sterowaniami definiuje się przeskalowaną energię sterowań rozciągniętych bądź skróconych do arbitralnie ustalonego czasu \tilde{T} . Pamiętając, że układ bezdryfowy (2.36) jest autonomiczny (w jawnej postaci nie zależy od czasu) zatem można przeskalować sterowania. Skalowana energia wyraża się wzorem $E_{\tilde{T}}(\mathbf{u}(\cdot)) = \int_0^{\tilde{T}} ((cu_1(ct))^2 + (cu_2(ct))^2) dt$, gdzie $c = (T \cdot i) / \tilde{T}$. Porównanie nakładów energetycznych na sterowania o równym czasie trwania dla zadań planowania o identycznych parametrach z wyjątkiem uzmiennionej δ_{\max} niesie bardziej istotną informację.

W algorytmach 7 i 8, kolejne podcele \mathbf{q}_p spełniające warunki (9.1), bądź (9.7)

Tablica 9.2. Wartości energii (E), energii znormalizowanej ($E_{\bar{T}=100}$) względem ustalonego czasu ruchu $\bar{T} = 100$ oraz liczba iteracji algorytmu 8 dla różnych wartości δ_{\max} i reprezentacji dla modelu jednokołowca

δ_{\max} [%]	„w przód”			„wstecz”			kanoniczna		
	energia		iteracji $i_{\sum j}$	energia		iteracji $i_{\sum j}$	energia		iteracji $i_{\sum j}$
	E	$E_{\bar{T}=100}$		E	$E_{\bar{T}=100}$		E	$E_{\bar{T}=100}$	
10	414.42	712.80	172 ₆₉₇	388.89	711.67	183 ₇₆₁	164.51	312.56	190 ₇₈₀
15	466.39	363.78	78 ₄₀₉	437.27	376.06	86 ₄₁₆	195.06	177.51	91 ₄₅₈
20	493.77	227.14	46 ₃₆₃	467.00	238.17	51 ₄₁₀	204.66	114.61	56 ₄₆₆
25	506.82	152.05	30 ₂₁₂	468.60	149.95	32 ₂₄₁	212.65	80.81	38 ₂₇₇
30	514.51	108.05	21 ₁₃₂	497.70	114.47	23 ₁₆₁	222.89	64.64	29 ₂₀₅
35	573.87	97.56	17 ₁₀₄	544.05	103.37	19 ₁₂₂	234.58	56.30	24 ₁₅₆
40	607.86	91.18	15 ₈₃	565.56	90.49	16 ₉₂	245.86	51.63	21 ₁₁₈
45	615.23	79.98	13 ₆₀	626.09	93.91	15 ₈₄	252.70	45.49	18 ₁₀₃
50	681.67	81.80	12 ₅₃	598.53	83.79	14 ₆₂	267.97	42.88	16 ₈₉
55	706.96	77.77	11 ₄₆	634.70	82.51	13 ₅₅	275.43	41.31	15 ₇₁
60	776.29	85.39	11 ₄₀	688.14	89.46	13 ₅₃	298.60	41.80	14 ₇₄
65	815.62	81.56	10 ₃₆	761.11	91.33	12 ₅₁	327.11	42.52	13 ₆₃
70	927.67	92.77	10 ₃₇	836.46	100.38	12 ₅₀	354.65	46.10	13 ₅₉
75	1070.01	107.00	10 ₃₇	976.61	107.43	11 ₄₆	375.33	48.79	13 ₅₀
80	1206.19	120.62	10 ₄₀	1074.82	128.98	12 ₄₅	426.62	55.46	13 ₆₀
85	1444.84	158.93	11 ₄₁	1557.95	186.95	12 ₅₁	453.62	49.90	11 ₅₅
90	1962.65	235.52	12 ₄₈	1779.98	178.00	10 ₄₈	543.04	70.59	13 ₅₆
95	2793.69	279.37	10 ₅₂	1852.34	185.23	10 ₄₉	526.56	52.66	10 ₄₇
100	2793.69	279.37	10 ₅₂	1799.02	179.90	10 ₄₅	732.16	109.82	15 ₆₀

wyznaczają łamaną $\mathbf{f}_l(t)$ w przestrzeni konfiguracyjnej. Przebieg trajektorii ocenimy podając błąd określony jako odległość euklidesowa między rzeczywistą trajektorią ruchu $\mathbf{f}_r(t)$, a łamaną $\mathbf{f}_l(t)$, czyli $\text{Err}(t) = \|\mathbf{f}_r(t) - \mathbf{f}_l(t)\|$. Wybrano dwie cechy opisujące charakterystykę przebiegu błędu: maksymalne odchylenie od łamanej, $\max_t(\text{Err}(t))$, oraz skumulowane odchylenie (błąd całkowity) $\int_0^T \text{Err}(t) dt$.

Wyniki symulacji zestawiono w tab. 9.1, 9.2 oraz 9.3. Wyniki zebrane w tab. 9.1, 9.2 wskazują, że energie sterowań w obydwu algorytmach dla identycznych zadań są porównywalne, energia E rośnie wraz ze wzrostem maksymalnej dopuszczalnej niedokładności δ_{\max} . Dla małych δ_{\max} bardzo gwałtownie wzrasta liczba iteracji, a zatem czas ruchu wydłuża się, a amplitudy sterowań są coraz mniejsze, co przekłada się na małą energię sterowań (pozornie, gdyż kosztem wydłużenia czasu ruchu). Natomiast dla dużych δ_{\max} liczba iteracji jest mała, ale dzieje się to kosztem dużych błędów poszczególnych planowań, zatem energia sterowań również rośnie. Na podstawie symulacji okazuje się, że skalowana energia $E_{\bar{T}=100}$ przyjmuje minimum dla pewnej wartości

Tablica 9.3. Wartości miar błędu dla różnych wartości δ_{\max} i reprezentacji dla modelu jedno-kołowca

rep.	algorytm 7						algorytm 8					
	„w przód”		„wstecz”		kanoniczna		„w przód”		„wstecz”		kanoniczna	
	błąd		błąd		błąd		błąd		błąd		błąd	
δ_{\max} [%]	całka	max	całka	max	całka	max	całka	max	całka	max	całka	max
10	14.67	1.98	12.85	3.22	31.09	1.55	15.09	1.33	13.79	1.35	31.85	0.93
15	12.59	3.41	10.14	3.33	23.01	1.49	12.36	1.87	10.34	1.59	23.54	1.30
20	11.83	3.69	8.55	3.28	19.01	1.47	11.26	2.32	8.55	1.61	19.31	1.30
25	11.09	3.22	7.57	3.10	16.78	1.52	10.55	2.53	7.28	1.66	16.76	1.53
30	11.70	3.82	7.13	2.84	15.56	1.57	10.33	2.01	6.98	1.97	15.50	1.49
35	11.46	2.77	7.59	2.76	14.91	1.83	10.73	2.58	7.89	2.39	14.84	1.52
40	11.83	3.03	8.38	2.86	14.57	1.73	11.19	2.68	8.55	2.46	14.57	1.64
45	12.51	3.16	8.81	5.45	14.67	1.71	11.58	2.46	8.83	2.95	14.55	1.62
50	13.65	3.73	11.67	3.88	15.07	1.91	12.69	3.15	10.09	2.57	14.86	1.84
55	14.46	4.06	13.34	4.28	15.53	2.16	13.43	3.28	11.44	2.73	15.09	1.97
60	15.83	4.55	15.93	4.81	16.48	2.21	14.87	3.76	13.35	3.12	16.04	2.16
65	18.17	5.41	18.57	5.68	17.36	2.45	15.88	3.59	15.96	3.70	17.41	2.36
70	18.72	5.15	14.46	8.08	18.32	2.60	18.34	4.29	18.59	4.54	18.82	2.67
75	22.07	6.36	21.74	7.44	20.20	3.01	21.55	5.28	23.63	5.51	19.87	3.01
80	28.70	8.72	19.10	10.02	23.44	3.58	24.24	6.57	27.12	5.63	21.31	3.31
85	27.88	14.19	21.24	9.96	21.91	3.96	28.96	7.78	40.79	9.66	23.03	4.06
90	29.02	14.19	23.93	10.01	28.41	4.59	38.23	10.1	42.75	12.69	27.98	4.18
95	29.71	14.19	24.19	10.01	30.23	5.24	40.94	16.11	45.07	12.69	25.05	5.42
100	30.08	14.19	21.95	14.14	41.46	6.12	40.94	16.11	44.79	12.45	38.47	5.43

δ_{\max} z przedziału pomiędzy około 40[%], a 70[%] (np. dla reprezentacji kanonicznej w algorytmie 8 (tab. 9.2) minimalna $E_{\dot{\tau}=100} = 41.31$ występuje dla $\delta_{\max} = 55$ [%]). Należy się spodziewać, że zależność $E_{\dot{\tau}=100}(\delta_{\max})$ nie będzie ciągła, gdyż sterowania są mocno podporządkowane liczbie iteracji, która ulega gwałtownym zmianom zależnym od położenia podcelów w przestrzeni stanu. Zatem położenie minimum energetycznego istotnie, zależy od zadania, gdyż pewne kierunki ruchu są łatwiejsze do generacji niż inne [91]. Generacja nawiasu Liego za pomocą sekwencji kroków jest droższa energetycznie niż za pomocą jednego ruchu [29], dlatego przy wzroście liczby iteracji wydatki energetyczne wzrastają. Jeszcze raz potwierdza się fakt, iż reprezentacja kanoniczna jest mniej kosztowna energetycznie niż reprezentacje „w przód” i „wstecz”.

Błąd całkowity dla algorytmów 7 i 8 jest zbliżony, przyjmuje minimum dla maksymalnej dopuszczalnej niedokładności δ_{\max} z przedziału (20, 40)[%] i jest uzależniony od wyboru reprezentacji ruchu. Błąd całkowity rośnie dla małych wartości δ_{\max} , wprawdzie kroki algorytmu są bardzo małe, jednak duża liczba iteracji powoduje narastanie

Tablica 9.4. Wartości energii (E), energii znormalizowanej ($E_{\bar{T}=100}$) względem ustalonego czasu ruchu $\bar{T} = 100$, liczba iteracji algorytmu 7 oraz miar błędu dla różnych wartości δ_{\max} i reprezentacji dla modelu trójkolowca

rep.	„w przód”					„wstecz”				
	energia		iteracji. $i_{\sum j}$	błąd		energia		iteracji $i_{\sum j}$	błąd	
	E	$E_{\bar{T}=100}$		całka	max	E	$E_{\bar{T}=100}$		całka	max
10	5628.85	65463.5	1163 ₇₃₉₇	50.60	1.04	4226.42	46532.9	1101 ₆₆₇₃	48.31	1.07
15	3675.22	14370.1	391 ₂₀₄₂	26.86	1.28	3991.40	15047.6	377 ₁₉₄₆	25.44	1.28
20	3321.08	6243.63	188 ₇₇₇	18.14	1.46	1586.68	2697.35	170 ₆₄₀	16.3	1.43
25	3079.42	3325.77	108 ₄₀₁	13.93	1.70	3369.91	3235.11	96 ₃₂₅	12.17	1.63
30	3095.59	2105.00	68 ₂₅₉	11.48	1.83	2922.31	1724.16	59 ₂₀₆	9.84	1.78
35	2709.22	1327.52	49 ₂₀₆	10.24	1.96	1876.55	750.62	40 ₂₁₄	8.57	1.86
40	2611.05	966.09	37 ₂₂₃	9.46	2.19	2418.65	677.22	28 ₁₅₂	7.92	2.00
45	2539.77	761.93	30 ₁₆₈	9.10	2.39	2232.38	446.48	20 ₉₄	7.70	2.24
50	2372.16	569.32	24 ₁₁₇	8.91	2.57	2101.22	315.18	15 ₇₂	7.79	2.42
55	2526.93	530.65	21 ₁₁₁	9.00	2.57	1999.91	239.99	12 ₄₇	8.20	2.72
60	2573.60	488.98	19 ₉₀	9.38	2.90	1873.00	168.57	9 ₃₃	8.85	3.11
65	2655.34	451.41	17 ₇₅	9.91	2.90	1914.08	153.13	8 ₃₀	9.96	3.69
70	2807.97	449.28	16 ₇₂	10.80	3.19	2103.64	189.33	9 ₃₂	12.01	4.21
75	3082.31	462.35	15 ₇₁	12.41	3.44	2270.03	181.60	8 ₃₅	13.39	5.01
80	3371.17	505.67	15 ₇₁	13.92	3.44	2747.28	247.26	9 ₃₆	16.51	5.74
85	4161.20	665.79	16 ₇₄	18.18	3.91	3118.50	280.66	9 ₄₀	19.97	6.29
90	5082.85	914.91	18 ₇₃	22.65	3.91	3417.36	307.56	9 ₃₉	21.77	6.42
95	9150.91	2470.75	27 ₁₂₄	43.86	5.84	3895.57	389.56	10 ₄₅	26.86	6.96
100	16982.50	7811.95	46 ₁₉₀	79.16	5.84	5433.07	651.97	12 ₅₆	40.75	7.61

błędu. Błąd maksymalny natomiast wykazuje tendencję do zmniejszania się wraz δ_{\max} . Warto też zauważyć, że błąd maksymalny jest uzależniony od kierunku.

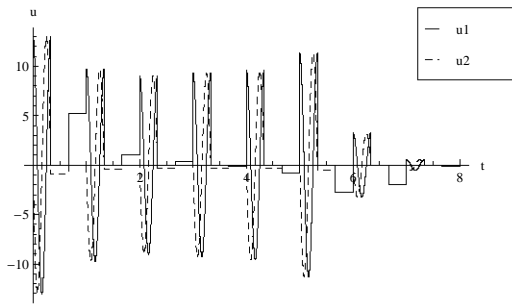
Dla układu (A.3) wyniki przedstawione w tab. 9.1, 9.2 wskazują, że pod względem energetycznym, wartości błędu oraz liczby iteracji algorytmu 7 i 8, są również bardzo do siebie zbliżone. Mimo, iż algorytm Lie algebraiczny jest mniej kosztowny obliczeniowo, unika skomplikowanej procedury nilpotentnej aproksymacji, to jednak dla wymiaru wektora stanu $\dim(\mathbf{q}) = 3$ przybliżanie systemu nilpotentnym odpowiednikiem jest bardzo proste, zatem jakość algorytmów 9.1, 9.2 jest porównywalna. Na rys. 9.1 przedstawiono wykresy sterowań, błędu oraz rzutu trajektorii rzeczywistej na powierzchnię wyznaczaną przez dwie pierwsze współrzędne wektora \mathbf{q} dla algorytmów 9.1, 9.2 generujących sterowania dla układu (A.3) – obie metody działają niemalże identycznie. W symulacjach rys. 9.1 celowo zwiększono dokładność osiągnięcia celu. Warto zauważyć fakt, iż zwiększenie współczynnika ϵ powoduje zwiększenie liczby iteracji. Algorytm wspomagany nilpotentą aproksymacją i Lie algebraiczny najpierw doprowadzają układ

Tablica 9.5. Wartości energii (E), energii znormalizowanej ($E_{\bar{T}=100}$) względem ustalonego czasu ruchu $\bar{T} = 100$, liczba iteracji algorytmu 8 oraz miar błędu dla różnych wartości δ_{\max} i reprezentacji dla modelu trójkółowca

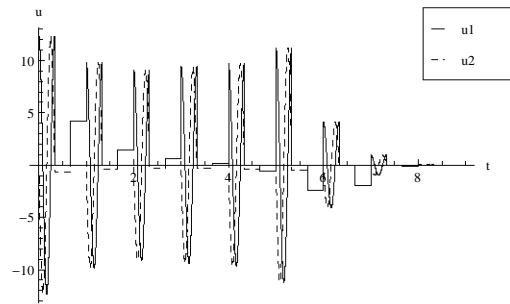
rep.	„w przód”						„wstecz”					
	energia		iteracji $i_{\sum j}$	błąd		energia		iteracji $i_{\sum j}$	błąd			
	E	$E_{\bar{T}=100}$		całka	max	E	$E_{\bar{T}=100}$		całka	max		
10	5275.59	65311.80	1238 ₈₂₉₀	69.02	0.78	6564.05	80081.5	1220 ₈₀₉₀	68.31	0.83		
15	5521.89	22418.90	406 ₂₁₅₅	35.61	1.05	4223.35	18759.5	400 ₂₁₀₃	35.23	1.12		
20	4396.36	9100.46	207 ₉₀₈	24.35	1.26	4515.82	8896.16	197 ₈₂₅	23.62	1.38		
25	3985.38	4782.45	120 ₄₅₄	18.34	1.49	2357.36	2640.25	112 ₄₀₆	17.59	1.58		
30	3593.56	2838.91	79 ₃₁₆	15.04	1.65	3618.82	2496.99	69 ₂₄₄	14.00	1.76		
35	4021.57	2332.51	58 ₂₂₁	13.21	1.74	3404.01	1667.96	49 ₂₂₇	12.18	1.92		
40	3570.64	1606.79	45 ₂₈₂	12.12	1.94	3543.74	1240.31	35 ₁₈₇	10.88	2.02		
45	3789.28	1402.03	37 ₂₀₈	11.51	2.03	3335.17	867.15	26 ₁₂₀	10.15	2.23		
50	3702.46	1147.76	31 ₁₇₃	11.19	2.13	3239.21	647.84	20 ₁₀₀	9.84	2.37		
55	3664.90	989.52	27 ₁₃₅	11.20	2.31	3164.37	537.94	17 ₆₆	9.96	2.50		
60	3864.73	927.54	24 ₁₂₈	11.65	2.48	3133.45	438.68	14 ₅₈	10.51	2.74		
65	3917.60	861.87	22 ₁₀₄	12.15	2.65	3215.28	385.83	12 ₄₀	11.38	3.09		
70	4162.49	832.50	20 ₉₁	13.01	2.82	3386.25	372.49	11 ₃₉	12.36	3.36		
75	4350.89	826.67	19 ₈₃	13.94	2.97	3140.94	345.50	11 ₄₆	11.08	3.25		
80	4484.35	807.18	18 ₈₅	15.13	3.13	3052.47	274.72	9 ₄₁	9.46	3.01		
85	4587.93	734.07	16 ₇₄	16.46	3.51	3052.47	274.72	9 ₄₁	9.46	3.01		
90	3793.73	417.31	11 ₅₃	14.23	4.98	3356.29	335.63	10 ₄₇	12.14	3.59		
95	3345.76	334.58	10 ₄₂	13.52	4.38	4382.75	482.10	11 ₅₃	17.63	4.37		
100	3466.75	312.01	9 ₃₇	14.34	4.38	4621.36	554.56	12 ₅₅	20.34	4.74		

w pobliże celu, następnie za pomocą sekwencji sterowań o bardzo małej amplitudzie tak długo poprawiają pozycję końcową, aż zostanie osiągnięty zadany błąd.

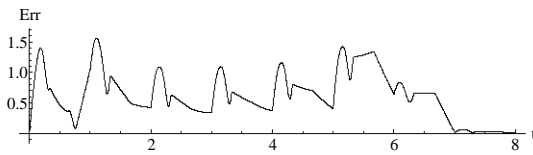
Przeprowadzono również serię symulacji dla trójkółowca (A_4) z punktem początkowym $q_0 = (5.0, 2.0, 0.0, 0.0)^T$ i końcowym $q_f = (10.0, 7.5, 1.0, 1.0)^T$, przy identycznych parametrach ϵ i T jak dla jednokołowca dla reprezentacji „w przód” oraz „wstecz” wyniki zebrano w tab. 9.4 i 9.5. Obserwacje poczynione dla jednokołowca pozostają prawdziwe dla trójkółowca. Algorytm wspomagany nilpotentną aproksymacją w testach (w tym konkretnym przypadku) okazał się minimalnie lepszy. Można również zauważyć, że przy dużej maksymalnej akceptowalnej niedokładności maleje liczba iteracji, a zarazem trajektoria ruchu jest bardziej chaotyczna (por. rys. 9.2), natomiast przy zmniejszeniu δ_{\max} trajektoria zbliża się do trajektorii wyznaczonej przez podcele.



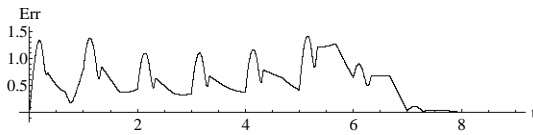
alg. 8 a



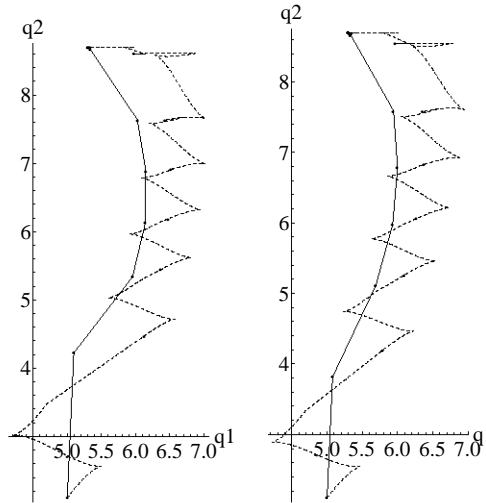
alg. 7 a



alg. 8 b



alg. 7 b



alg. 8 c

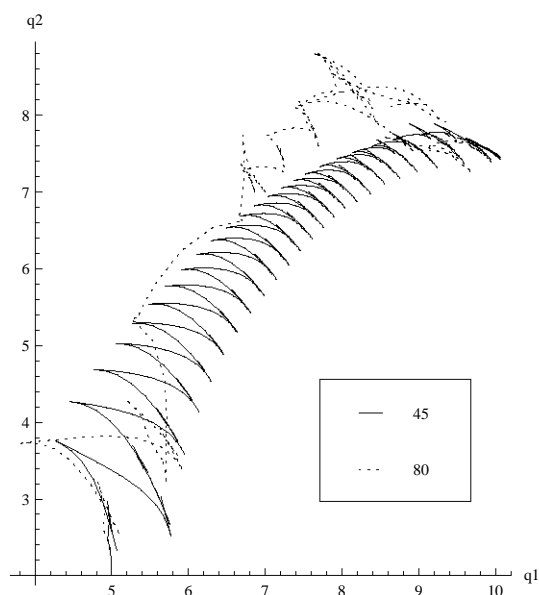
alg. 7 c

Rysunek 9.1. Symulacja algorytmów 7, 8 dla układu jednokołowca przy danych wejściowych $\epsilon = 10^{-4}$, $T = 1$, $q_0 = (5.0, 2.1, 0.9)^T$; $q_f = (5.3, 8.7, 0.0)^T$; $\delta_{\max} = 50$, a – sterowania, b – błąd od łamanej wyznaczonej przez pod cele q_f , c – trajektoria ruchu (linia przerywana) na płaszczyźnie q_1, q_2 , prosta łamana wyznaczająca podcele (linia ciągła)

9.4. Podsumowanie

W wyniku przeprowadzonych symulacji stwierdzono, że algorytm Lie algebraiczny i wspomagany nilpotentną aproksymacją generują podobne wyniki. Zatem wykorzystanie skomplikowanej procedury nilpotentnej aproksymacji nie wpływa na polepszenie iteracyjnej metody LS.

Przeprowadzono także badania w celu ustalenia optymalnej wartości δ_{\max} (maksymalnej dopuszczalnej niedokładności). Gdy δ_{\max} jest małe wtedy dopuszcza się sytuację generacji kierunków, przy pomocy sekwencji coraz krótszych kawałków ruchu, a jest to kosztowne energetycznie [29]. Natomiast dopuszczenie dużego błędu δ_{\max} , powoduje duże skoki przy generacji podcelów, dzięki czemu maleją koszty obliczeniowe skom-



Rysunek 9.2. Trajektoria ruchu trójkołowca na płaszczyźnie (q_1, q_2) , algorytm 8 wygenerowany dla danych wejściowych: $\epsilon = 0.01$, $T = 1$, $q_0 = (5.0, 2.0, 0.0, 0.0)^T$, $q_f = (10.0, 7.5, 1.0, 1.0)^T$, oraz dwóch wartości $\delta_{\max} = 45$ i $\delta_{\max} = 80$

plikowanej procedury LS, ale zarazem traci się energię na ruch w złych kierunkach (powoli zbliżających się do celu). Wybór δ_{\max} powinien podyktowany być wyważonym balansem pomiędzy tymi dwiema zależnościami (liczbą iteracji i minimum energetycznym). Bardzo duża liczba iteracji powoduje także bardzo długi czas realizacji ruchu, a przeskalowanie (skrócenie) takich sterowań powoduje pojawienie się sygnałów wysokoczęstotliwościowych, często nie realizowalnych w praktyce.

Wybór δ_{\max} wpływa również na maksymalny błąd, oraz całkę z błędu. Dobór maksymalnej dopuszczalnej niedokładności może podyktowany być minimalizacją tych parametrów. Odchylenia od toru ruchu wyznaczanego przez podcele, przeważnie maleją gdy zwiększa się liczba iteracji (δ_{\max} małe). Należy jednak pamiętać, że algorytmy 9.1, 9.2 nie gwarantują, że odległości pomiędzy poszczególnymi planowaniami ruchu będą małe. Odległość pomiędzy dwoma zadanymi podcelami może okazać się duża pomimo małego δ_{\max} , zatem też błąd okazałby się większy. Podsumowując zbytne zmniejszanie δ_{\max} nie gwarantuje zmniejszenia błędu, natomiast skutkuje negatywnie poprzez zwiększanie się liczby iteracji, a zarazem czasu ruchu.

10. Podsumowanie

W dysertacji wszechstronnie zbadano metodę Lafferriera-Sussmanna pierwotnie wykorzystywaną do planowania ruchu układów nilpotentnych będących podklasą bezdryfowych układów nieholonomicznych. Metodę rozszerzono przez wprowadzenie znacznie większej niż w algorytmie oryginalnym liczby reprezentacji ruchu, przez co uzyskano możliwość jej zastosowania w środowiskach kolizyjnych. Z każdą reprezentacją ruchu związane jest równanie Chena-Fliessa-Sussmanna. Wykazano jaką strukturę mogą mieć równania CFS oraz zaproponowano dwa efektywniejsze niż literaturowe algorytmy generacji równania. Podczas wyprowadzenia równań CFS należy przyjąć pewną bazę algebry Liego. W zadaniach robotycznych wymagających takiej bazy najczęściej (jedynie) stosowana jest baza Halla. W dysertacji przedstawiono także inne literaturowe bazy (Lyndona, Shirshova, Chibrikova), wraz ze sposobem ich generacji. Jakkolwiek zastosowanie alternatywnych baz może mieć pewne zalety, to jednak mogą się one uwidocznić dopiero dla układów o dużej liczbie zmiennych stanu, gdyż dopiero wtedy elementy baz różnią się istotnie. Jednym z etapów algorytmu LS jest dobór trajektorii referencyjnej. Pokazano symulacyjnie, analitycznie dla wybranych klas układów nilpotentnych oraz sformułowano ogólne twierdzenie o niezależności działania planera ruchu od trajektorii referencyjnej. Wnioskiem praktycznym jest przyjęcie odcinka jako trajektorii referencyjnej. W klasie układów łańcuchowych wykazano równoważność metody LS z metodą sterowań sinusoidalnych Murraya-Sastry'ego. Rozszerzając stosowalność algorytmu LS na układy bezdryfowe zaimplementowano pomysł Sussmanna stosowania algorytmu nilpotentnej aproksymacji w kolejnych punktach przestrzeni konfiguracyjnej. Do sprawdzenia jakości nilpotentnej aproksymacji zaproponowano autorskie miary. Dwie z nich porównują pola wektorowe oryginalnego układu i jego aproksymowanej wersji. Miary nie dają wskazówki o rzeczywistej przydatności aproksymacji w planowaniu ruchu, bowiem nie oceniają wyróżnionego kierunku ruchu. Dlatego zdefiniowano, zaimplementowano i przetestowano na wybranych układach bezdryfowych miarę, która ocenia efektywność aproksymacji w zadanym kierunku ruchu. Wykorzystując tę miarę zauważono, że efektywność aproksymacji zależy istotnie od wybranego kierunku ruchu. Algorytm LS wspomagany nilpotentną aproksymacją oraz (w celach porównawczych) algorytm metody Lie-algebraicznej zaimplementowano i przetestowano dla dwóch układów bezdryfowych. Wyniki symulacyjne wskazują, że możliwe jest efektywne rozwiązanie zadania planowania dla układów bezdryfowych obiema metodami, jednak liczba punktów, w których należy ponawiać procedurę nilpotentnej aproksymacji może być znaczna, natomiast zyski w porównaniu z metodą Lie algebraiczną są znikome.

Bibliografia

- [1] A. A. Agrachev, R. V. Gamkrelidze. The exponential representation of flows and the chronological calculus. *Matematicheskii Sbornik*, 107(4):467–532, 1978.
- [2] G. B. Airy. On the regulator of the clock-work for effecting uniform movement of equatorials. *Memoirs of the Royal Astronomical Society*, 11:249–267, 1840.
- [3] A. Astolfi. *Asymptotic stabilization of nonholonomic systems with discontinuous control*. Rozprawa doktorska, Swiss Federal Institute of Technology, Zurich, 1996.
- [4] A. Bellaïche, J. P. Laumond, M. Chyba. Canonical nilpotent approximation of control systems: application to nonholonomic motion planning. *IEEE Conf. on Decision and Control*, strony 2694–2699, 1993.
- [5] L. Bokut, Y. Chen, Y. Li. Anti-commutative Gröbner-Shirshov basis of a free Lie algebra. *Science in China Series A: Mathematics*, 52:244–253, 2009.
- [6] L. A. Bokut, Y. Chen. Gröbner-Shirshov bases for Lie algebras: after A. I. Shirshov. *ArXiv e-prints*, 2008.
- [7] L. A. Bokut, Y. Chen, Y. Chen. Gröbner-Shirshov bases for Lie algebras over a commutative algebra. *ArXiv e-prints*, 2010.
- [8] R. W. Brockett. Control theory and singular Riemannian geometry. P. J. Hilton, G. S. Young, redaktorzy, *New Directions in Applied Mathematics*, strony 11–27. Springer, New York, 1981.
- [9] I. N. Bronsztejn, K. A. Siemiendiajew, G. Musiol, H. Muhlig. *Nowoczesne Kompendium matematyki*. Wydawnictwo Naukowe PWN, Warszawa, 2004.
- [10] G. Bushnell, D. M. Tilbury, S. Sastry. Steering three-input nonholonomic systems - the firetruck example. *Int. Journal of Robotics Research*, 14(4):366–381, 1995.
- [11] G. Casalino, M. Aicardi, A. Bicchi, A. Balestrino. Closed loop steering and path following for unicycle-like vehicles: a simple Lapunov function based approach. *IEEE Robotics and Automation Magazine*, 2(1):27–35, 1995.
- [12] E. S. Chibrikov. A right normed basis for free Lie algebras and Lyndon-Shirshov words. *Journal of Algebra*, 302(2):593–612, 2006.
- [13] W. L. Chow. Über systeme von linearen partiellen Differentialgleichungen erster Ordnung. *Math. Ann.*, 117(1):85–105, 1939.
- [14] P. di Giamberardino. Exact steering with arbitrarily bounded input for a class of driftless dynamics. *IEEE Conf. on Decision and Control*, wolumen 4, strony 4170–4175, 1998.
- [15] A. W. Divelbiss, J. Wen. Nonholonomic path planning with inequality constraints. *IEEE Conf. on Robotics and Automation*, wolumen 1, strony 52–57, 1994.
- [16] L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79:497–516, 1957.
- [17] I. Dułęba. *Algorithms of motion planning for nonholonomic robots*. Oficyna Wydawnicza PWr, Wrocław, 1998.

- [18] I. Dułęba. On a computationally simple form of the generalized Campbell-Baker-Hausdorff-Dynkin formula. *Systems & Control Letters*, 34:191–202, 1998.
- [19] I. Dułęba. Lie-bracket based dexterity measure for driftless nonholonomic systems. *Proc. of Int. Symp. of Mathematical Theory of Networks and Systems*, 2000.
- [20] I. Dułęba. How many P. Hall representations there are for motion planning of nilpotent nonholonomic systems? *Proc. of Int. Conf. on Methods and Models in Automation and Robotics*, 14, 2009.
- [21] I. Dułęba, J. Jagodziński. On the structure of Chen-Fliess-Sussmann equation for Ph. Hall motion representation. K. Tchoń, redaktor, *Progress in Robotics*, strony 9–20. Wydawnictwa Komunikacji i Łączności, Warszawa, 2008.
- [22] I. Dułęba, J. Jagodziński. Computational algebra support for the Chen-Fliess-Sussmann differential equation. *Workshop on Robot Motion and Control*, strony 133–142. Springer Verlag, 2009.
- [23] I. Dułęba, J. Jagodziński. Motion representations for the Lafferriere-Sussmann’s algorithm of control nilpotent systems. *Int. Journal of Applied Mathematics and Computer Science*, 21(3), 2011. w druku.
- [24] I. Dułęba, J. Jagodziński. Practical efficiency evaluation of a nilpotent approximation algorithm for driftless nonholonomic systems. *Workshop on Robot Motion and Control*, 2011. w druku.
- [25] I. Dułęba, W. Khefifi. Pre-control form of the gCBHD formula and its application to nonholonomic motion planning. I. Dułęba, J. Z. Sasiadek, redaktorzy, *IFAC Symposium on Robot Control*, strony 123–128, Wrocław, 2003.
- [26] I. Dułęba, W. Khefifi. Pre-control form of the generalized Campbell-Baker-Hausdorff-Dynkin formula for affine nonholonomic systems. *Systems & Control Letters*, 55(2):146 – 157, 2006.
- [27] I. Dułęba, J. Sówka. Sfery nieholonomiczne dla systemów bezdryfowych. *Prace Naukowe Instytutu Cybernetyki Technicznej Politechniki Wrocławskiej*, wolumen 98, strony 161–169. Oficyna Wydawnicza PWr, 1998.
- [28] I. Duleba. On Ph. Hall expansion of $\log(\exp x \cdot \exp y)$. Raport instytutowy, Inst. Informatyki, Automatyki i Robotyki, Wrocław, 2008.
- [29] I. Duleba, J. Z. Sasiadek. Calibration of controls in steering nonholonomic systems. *Control Engineering Practice*, 9(2):217–225, 2001.
- [30] E. B. Dynkin. Calculation of the coefficients in the Campbell-Hausdorff formula. *Doklady Akad. Nauk SSSR (N. S.)*, 57:323–326, 1947. Angielskie tłumaczenie: A. Kleshchev (1997).
- [31] W. Findeisen, J. Szymanowski, A. Wierzbicki. *Teoria i metody obliczeniowe optymalizacji*. PWN, Warszawa, 1980.
- [32] G. H. Golub, C. F. Van Loan. *Matrix Computations*. Johns Hopkins Press, Baltimore, MD, USA, wydanie 2, 1989.
- [33] B. Goodwine, J. W. Burdick. Motion planning for kinematic stratified systems with application to quasi-static legged locomotion and finger gaiting. *IEEE Trans. on Robotics and Automation*, 12(2):209–222, 2002.
- [34] J. W. Goodwine. *Control of Stratified Systems with Robotic Applications*. Praca doktorska, California Institute of Technology Pasadena, California, 1998.
- [35] M. Hall. A basis for free Lie rings and higher commutators in free groups. *Proc. Amer. Math. Soc.*, 1:575–581, 1950.

- [36] M. Hall. *The theory of groups*. New York: Macmillan, 1959.
- [37] Ph. Hall. A contribution to the theory of groups of prime power order. *Proc. London Math. Soc.*, wolumen 36, strony 29–95, 1934.
- [38] H. Hermes. Nilpotent approximations of control systems. *IEEE Conf. on Decision and Control*, wolumen 24, strony 708–710, 1985.
- [39] H. Hermes, A. Lundell, D. Sullivan. Nilpotent bases for distributions and control systems. *Journal of Differential Equations*, 55(3):385–400, 1984.
- [40] R. Hossa. *Modele i algorytmy sterowania kołowych robotów mobilnych*. Praca doktorska, Politechnika Wrocławska, Instytut Cybernetyki Technicznej, Wrocław, 1996.
- [41] M. Ishikawa, T. Fujino. Control of the double-linked trident snake robot based on the analysis of its oscillatory dynamics. *Proc. IEEE/RSJ Intelligent Robots and Systems*, strony 1314–1319, 2009.
- [42] M. Ishikawa, P. Morin, C. Samson. Tracking control of the trident snake robot with the transverse function approach. *IEEE Conf. on Decision and Control*, 48, strony 4137–4143, 2009.
- [43] G. Jacob. Lyndon discretization and exact motion planning. *European Control Conf.*, strony 1507–1512, Grenoble, 1991.
- [44] J. Jagodziński. Metoda Sussmanna i Lafferriere’a analitycznego planowania ruchu układów nieholonomicznych. Praca magisterska, Politechnika Wrocławska, Instytut Informatyki, Automatyki i Robotyki, Wrocław, 2007.
- [45] J. Jagodziński. Równoważność algorytmu Lafferrierra-Sussmanna z metodą Murray-Sastry’ego w planowaniu ruchu układów łańcuchowych. *Problemy Robotyki, Tom. II, Prace Naukowe, Elektronika, z. 175*, strony 515–524, Warszawa, 2010. Oficyna Wydawnicza Politechniki Warszawskiej.
- [46] M. Kawski. Bases for Lie algebras and a continuous CBH formula. V. Blondel, A. Megretski, redaktorzy, *Unsolved Problems in Mathematical Systems and Control Theory*, strony 97–102. Princeton Univ. Press, 2004.
- [47] M. Kawski. Problem 3.3. Bases for Lie algebras and a continuous CBH formula. V. D. Blondel, A. Megretski, redaktorzy, *Unsolved problems in mathematical systems and control theory*, strony 97–102. e-Book, 2008.
- [48] W. Khefifi. *Lie-algebraiczne metody planowania ruchu układów afinicznych*. Praca doktorska, Instytut Cybernetyki Technicznej Politechniki Wrocławskiej, Wrocław, 2005.
- [49] J. Komorowski. *Od liczb zespolonych do tensorów, spinorów, algebr Liego i kwadryk*. PWN, Warszawa, 1978.
- [50] T. Koralewski. Algorytm nilpotentnej aproksymacji bezdryfowych układów nieholonomicznych. Praca magisterska, Politechnika Wrocławska, Instytut Informatyki, Automatyki i Robotyki, Wrocław, 2009.
- [51] N. T. Koussoulas, P. Skiadas. Motion planning for drift-free nonholonomic system under a discrete levels control constraint. *Journal of Intelligent and Robotic Systems*, 32:55–74, 2001.
- [52] N. T. Koussoulas, P. Skiadas. Symbolic computation for mobile robot path planning. *Journal of Symbolic Computation*, 37:761–775, 2004.
- [53] G. Lafferriere. A general strategy for computing steering controls of systems without drift. *IEEE Conf. on Decision and Control*, strony 1115–1120, 1991.
- [54] G. Lafferriere, H. Sussmann. Motion planning for controllable systems without drift: a preliminary report. Raport instytutowy, Rutgers Center for System and Control, 1990.

- [55] G. Lafferriere, H. Sussmann. Motion planning for controllable systems without drift. *IEEE Conf. on Robotics and Automation*, strony 1148–1153, 1991.
- [56] F. Lamiroux, S. Sekhavat, J. P. Laumond. Motion planning and control for Hilare pulling a trailer. *IEEE Trans. on Robotics and Automation*, 15(4):640–652, 1999.
- [57] J. P. Laumond, P. E. Jacobs, M. Taix, R. M. Murray. A motion planner for nonholonomic mobile robots. *IEEE Trans. on Robotics and Automation*, 10(5):577–593, 1994.
- [58] J. P. Laumond, G. Oriolo. Nilpotent approximation of nonholonomic systems with singularities: A case study. *IFAC Nonlinear Control Systems Design Symp.*, strony 777–782, 1998.
- [59] J. P. Laumond, S. Sekhavat, F. Lamiroux. Guidelines in nonholonomic motion planning for mobile robots. J. P. Laumond, S. Sekhavat, F. Lamiroux, redaktorzy, *Robot Motion Planning and Control*, strony 1–53. Springer Verlag, 1998.
- [60] S.M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.
- [61] W. Liu. Averaging theorems for highly oscillatory differential equations and iterated Lie brackets. *SIAM Journal on Control and Optimization*, 35(6):1989–2020, 1997.
- [62] R.C. Lyndon. On Burnside’s problem I. *Trans. Amer. Math. Soc.*, 77:202–215, 1954.
- [63] W. Magiera. Planowanie ruchu kuli toczącej się: metoda funkcji transwersalnych i metoda endogenicznej przestrzeni konfiguracyjnej. *Problemy Robotyki, Tom. II, Prace Naukowe, Elektronika, z. 175*, strony 537–546, Warszawa, 2010. Oficyna Wydawnicza Politechniki Warszawskiej.
- [64] A. A. Markov. Some examples of the solution of a special kind of problem on greatest and least quantities. *Soobshch. Kharkovsk. Mat. Obshch.*, 1:250–276, 1887.
- [65] G. Melançon. Combinatorics of bases of free Lie algebras. Raport instytutowy LaBRI, U.R.A. CNRS # 1304, Université Bordeaux I, 1996. Technical report.
- [66] M. Michałek. *Sterowanie metodą orientowanych pól wektorowych dla podklasy systemów nieholonomicznych*. Praca doktorska, Politechnika Poznańska, Poznań, 2006.
- [67] S. Monaco, D. Normand-Cyrot. An introduction to motion planning under multirate digital control. *IEEE Conf. on Decision and Control*, wolumen 2, strony 1780–1785, 1992.
- [68] R. Mukherjee, D. P. Anderson. Nonholonomic motion planning using Stoke’s theorem. *IEEE Conf. on Robotics and Automation*, wolumen 3, strony 802–809, 1993.
- [69] H. Munthe-Kaas, B. Owren. Computations in a free Lie algebra. Raport instytutowy 148, Departament of Informatics, University of Bergen, 1999.
- [70] R. Murray, Z. X. Li, S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. Boca FL, CFC Press, 1994.
- [71] R. Murray, S. Sastry. Steering nonholonomic systems in chained form. *IEEE Conf. on Decision and Control*, strony 1121–1126, Brighton, 1991.
- [72] R. M. Murray. *Robotic Control and Nonholonomic Motion Planning*. Praca doktorska, College of Engineering University of California, Berkeley, 1990.
- [73] R. M. Murray. Nilpotent bases for a class of nonintegrable distributions with applications to trajectory generation for nonholonomic systems. *Math. Control Signals Systems*, 7:58–75, 1994.
- [74] Y. Nakamura. *Advanced Robotics: Redundancy and Optimization*. Wesley, New York, 1991.
- [75] Y. Nakamura, W. Chung, O.J. Sordalen. Design and control of the nonholonomic manipulator. *IEEE Trans. on Robotics and Automation*, 17(1):48–59, 2001.

- [76] Y. Nakamura, R. Mukherjee. Nonholonomic path planning of space robots via a bidirectional approach. *IEEE Trans. on Robotics and Automation*, 7(4):500–514, 1991.
- [77] A. Ollero. *Robótica, Manipuladores y Robots Móviles*. Alfaomega-Marcombo, 2001.
- [78] J. A. Reeds, L. A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific J. Math.*, 145(2):367–393, 1990.
- [79] D. B. Reister, F. G. Pin. Time optimal trajectories for mobile robots with two independently driven wheels. *Int. Journal of Robotics Research*, 13(1):38–54, 1994.
- [80] Ch. Reutenauer. *Free Lie Algebras*. Clarendon Press, Oxford, 1993.
- [81] P. Rouchon, M. Fliess, J. Levine, P. Martin. Flatness, motion planning and trailer systems. *IEEE Conf. on Decision and Control*, strony 2700–2705, 1993.
- [82] E. Sachkova. Solution of a control problem for a nilpotent system. *Differential Equations*, 44:1768–1772, 2008.
- [83] E. Sachkova. Approximate solution of a control problem on the basis of the nilpotent approximation. *Differential Equations*, 45:1386–1395, 2009. 10.1134/S0012266109090146.
- [84] J. Sawada, C. R. Miers, F. Ruskey. Generating Lyndon brackets: a basis for the n -th homogeneous component of the free Lie algebra. *Journal of Algorithms*, 46:21–26, 2001.
- [85] M. P. Schützenberger. Sur une propriété combinatoire des algèbres de Lie libres pouvant être utilisée dans un problème de mathématiques appliquées. *Séminaire Dubreil-Pisot, Inst. H. Poincaré, Paris*, 1:1–22, 1958–59.
- [86] J.-P. Serre. *Lie algebras and Lie groups*. New York: W.A. Benjamin, 1965.
- [87] A. I. Shirshov. O svobodnyh kolchah Li. *Matematicheskii Sbornik*, 45(87)(2):113–122, 1958. po rosyjsku.
- [88] A. I. Shirshov. On bases for free Lie algebras. *Algebra i Logika Sem.*, strony 14–19, 1962.
- [89] W. Sluis, W. Shadwick, R. Grossman. Nonlinear normal forms for driftless control systems. *IEEE Conf. on Decision and Control*, wolumen 1, strony 320–325, 1994.
- [90] E. Sontag. Gradient techniques for systems with no drift: a classical idea revisited. *IEEE Conf. on Decision and Control*, wolumen 3, strony 2706–2711, 1993.
- [91] J. Sówka. Lokalne własności metryczne układów nieholonomicznych. Praca magisterska, Politechnika Wrocławska, Instytut Cybernetyki technicznej, Wrocław, 1998.
- [92] M. Spivak. *Analiza na rozmaitościach*. Wydawnictwo Naukowe PWN, Warszawa, wydanie 2, 2005.
- [93] R. S. Strichartz. The Campbell-Baker-Hausdorff-Dynkin formula and solutions of differential equations. *Journal of Functional Analysis*, 72:320–345, 1987.
- [94] H. Struemper. Nilpotent approximation and nilpotentization for under-actuated systems on matrix Lie groups. *IEEE Conf. on Decision and Control*, strony 4188–4193, 1998.
- [95] H. K. Struemper. *Motion Control for Nonholonomic Systems on Matrix Lie Groups*. Rozprawa doktorska, The University of Maryland, 1997.
- [96] H. Sussmann. Thirty years of optimal control: was the path unique? E. O. Roxin, redaktor, *Modern Optimal Control*, strony 359–375. Marcel Dekker, 1989.
- [97] H. Sussmann. Two new methods for motion planning for controllable systems without drift. *European Control Conf.*, strony 1501–1506, 1991.
- [98] H. Sussmann. New differential geometric methods in nonholonomic path finding. A. Isidori, T. J. Tarn, redaktorzy, *Systems, Models, and Feedback: Theory and Applications*, strony 365–384. Birkhäuser, 1992.
- [99] Y. Tan, Z. Jiang, Z. Zhou. A nonholonomic motion planning and control based on

- chained form transformation. *Proc. IEEE/RSJ Intelligent Robots and Systems*, strony 3149–3153, 2006.
- [100] K. Tchoń, J. Jakubiak. Acceleration-driven kinematics of mobile manipulators: an endogenous configuration space approach. J. Lenarčič, C. Galletti, redaktorzy, *On Advances in Robot Kinematics*. The Netherlands, Kluwer Academic Publishers, 2004.
- [101] K. Tchoń, A. Mazur, I. Dulęba, R. Hossa, R. Muszyński. *Manipulatory i roboty mobilne*. Akademicka Oficyna Wydawnicza PLJ, Warszawa, 2000.
- [102] M. Torres-Torriti. Lie tools package, user's guide, version 1.03. Lie Tools Package dostępny na <http://www.cim.mcgill.ca/~migueltt/ltp/ltp.html>, 2004.
- [103] F. ur Rehman. Steering control of a hopping robot model during the flight phase. *Proc. IEE Control Theory and Applications*, wolumen 152, strony 645–653, 2005.
- [104] V. Varadarajan. *Lie Groups, Lie Algebras, and Their Representations*. Springer Verlag, 1984.
- [105] M. Vendittelli, G. Oriolo, F. Jean, J. P. Laumond. Nonhomogeneous nilpotent approximations for nonholonomic system with singularities. *IEEE Trans. on Automatic Control*, 49(2):261–266, 2004.
- [106] M. Vendittelli, G. Oriolo, J. P. Laumond. Steering nonholonomic systems via nilpotent approximations: The general two-trailer system. *IEEE Conf. on Robotics and Automation*, strony 823–329, 1999.
- [107] G. Viennot. *Algebres de Lie libres et monoïdes libres: Bases des algebres de Lie libres et factorisations des monoïdes libres*. Lecture Notes in Mathematics, 691. Springer Verlag, 1978.
- [108] S. A. Wadoo. *Feedback Control and Nonlinear Controllability of Nonholonomic Systems*. Praca doktorska, Blacksburg, Virginia Polytechnic, 2003.

A. Modele

A.1. Integrator Brocketta

Integrator Brocketta jest układem o trójwymiarowej przestrzeni stanu zadany równaniami

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -q_2 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 1 \\ q_1 \end{bmatrix} u_2 = \mathbf{g}_1(\mathbf{q})u_1 + \mathbf{g}_2(\mathbf{q})u_2. \quad (A.1)$$

Jest układem nilpotentnym stopnia 2, gdyż wszystkie pola wektorowe bazy Halla warstwy trzeciej zerują się tożsamościowo $[\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]] = [\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]] = \mathbf{0}$, a niezerowym polem jest $[\mathbf{g}_1, \mathbf{g}_2] = (0, 0, 2)^T$. Jest także sterowalny, w krótkim czasie, spełniając warunek rzędu (2.37), $\text{rank}(\mathbf{g}_1, \mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]) = 3$.

A.2. Czterowymiarowy układ łańcuchowy

Czterowymiarowy układ łańcuchowy jest opisany za pomocą równań

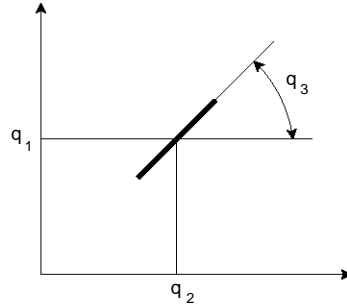
$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ q_2 \\ q_3 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} u_2 = \mathbf{g}_1(\mathbf{q})u_1 + \mathbf{g}_2(\mathbf{q})u_2. \quad (A.2)$$

Oprócz generatorów układu $\mathbf{g}_1, \mathbf{g}_2$, tylko pole wektorowe $[\mathbf{g}_1, \mathbf{g}_2] = (0, 0, -1, 0)^T$ oraz $[\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]] = (0, 0, 0, 1)^T$ nie znikają tożsamościowo ($[\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]] = \mathbf{0}$). Układ zatem jest nilpotentny stopnia 3, oraz sterowalny, gdyż $\text{rank}(\mathbf{g}_1, \mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2], [\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]]) = 4$.

A.3. Jednokołowy robot mobilny

Jednokołowy robot mobilny przedstawiono na rys. A.1, gdzie współrzędne q_1, q_2 oznaczają położenie robota, a q_3 jego orientację. Jeśli sterowaniami jest prędkość ruchu postępowego oraz prędkość zmiany orientacji, wtedy równania układu są postaci

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} \cos q_3 \\ \sin q_3 \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_2 = \mathbf{g}_1(\mathbf{q})u_1 + \mathbf{g}_2(\mathbf{q})u_2. \quad (A.3)$$



Rysunek A.1. Jednokołowy robot mobilny, rzut z góry

Wyliczając kolejne nawiasy Liego otrzymujemy $[\mathbf{g}_1, \mathbf{g}_2] = (\sin q_3, -\cos q_3, 0)^T$ oraz $\text{ad}_{\mathbf{g}_2}^{k+1}[\mathbf{g}_1, \mathbf{g}_2] = \frac{\partial}{\partial q_3} \text{ad}_{\mathbf{g}_2}^k[\mathbf{g}_1, \mathbf{g}_2]$ dla $k \geq 1$. Ostatnią równość dowodzimy w sposób następujący. Załóżmy, że $\text{ad}_{\mathbf{g}_2}^k[\mathbf{g}_1, \mathbf{g}_2] = (f_1(q_3), f_2(q_3), 0)^T$ (jest tak oczywiście dla $k = 0$, a gdy udowodnimy równość, to także dla $k \in \mathbb{N}$), więc

$$\text{ad}_{\mathbf{g}_2}^{k+1}[\mathbf{g}_1, \mathbf{g}_2] = [\mathbf{g}_2, \text{ad}_{\mathbf{g}_2}^k[\mathbf{g}_1, \mathbf{g}_2]] = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} f_1(q_3) \\ f_2(q_3) \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial q_3} f_1(q_3) \\ \frac{\partial}{\partial q_3} f_2(q_3) \\ 0 \end{bmatrix} = \frac{\partial}{\partial q_3} \text{ad}_{\mathbf{g}_2}^k[\mathbf{g}_1, \mathbf{g}_2].$$

Wszystkie pozostałe nawiasy zawierające więcej niż jedno pole \mathbf{g}_1 zerują się tożsamościowo. Układ ten jest sterowalny, bowiem $\text{rank}(\mathbf{g}_1, \mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]) = 3$ oraz nie jest nilpotentny.

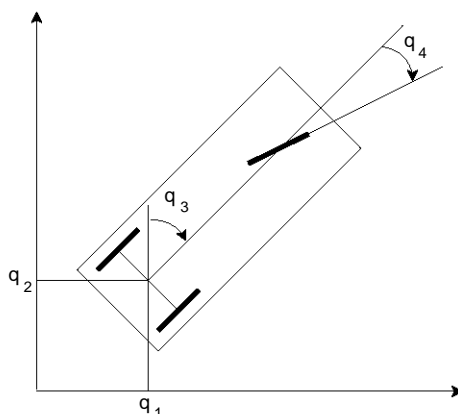
A.4. Trójkołowy robot mobilny

Model kinematyczny trójkołowego pojazdu [77] przedstawionego na rys. A.2, o wektorze stanu $\mathbf{q} = (q_1, q_2, q_3, q_4)^T$, jest postaci

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \begin{bmatrix} -s_{q_3}c_{q_4} \\ c_{q_3}c_{q_4} \\ s_{q_4} \\ 0 \end{bmatrix} \mathbf{u}_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \mathbf{u}_2 = \mathbf{g}_1(\mathbf{q})\mathbf{u}_1 + \mathbf{g}_2(\mathbf{q})\mathbf{u}_2, \quad (A.4)$$

gdzie dla uproszczenia zapisu przyjęto następujące oznaczenia $c_{q_i} = \cos(q_i)$, $s_{q_i} = \sin(q_i)$. Współrzędne q_1, q_2 (rys. A.2) opisują położenie robota (środek osi tylnego koła), q_3 to orientacja pojazdu, a q_4 – orientacja przedniego koła sterującego.

Zakłada się, że napęd trójkołowca znajduje się w przednim kole. W przypadku napędu na tylne koła generatory równania (A.4) byłyby postaci $\mathbf{g}_1 = (\cos q_3, \sin q_3, \tan q_4, 0)^T$ oraz $\mathbf{g}_2 = (0, 0, 0, 1)^T$. Napęd na tylną oś skutkuje osobliwością w przednim kole dla orientacji $q_4 = \pm \frac{\pi}{2}$. Dzięki przeniesieniu napędu na przednią oś zapobiega się osobliwości w tych punktach.



Rysunek A.2. Trójkolowy robot mobilny, rzut z góry.

Pola wyższych stopni dla układu (A.4) są postaci $[\mathbf{g}_1, \mathbf{g}_2] = (-s_{q_3}s_{q_4}, c_{q_3}s_{q_4}, -c_{q_4}, 0)^T$, $[\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]] = (-c_{q_3}, -s_{q_3}, 0, 0)^T$, $[\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]] = \mathbf{g}_1$. Układ jest sterowalny, gdyż spełnia warunek rzędu (2.37) $\text{rank}(\mathbf{g}_1, \mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2], [\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]]) = 4$. Nie jest układem nilpotentnym, gdyż na mocy równości $[\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]] = \mathbf{g}_1 \neq \mathbf{0}$ możliwe jest wygenerowanie niezerowego pola dowolnego stopnia.

B. Podstawowe struktury algebraiczne

Struktura algebraiczna [9, 49] to zbiór złożony ze skończonej liczby zbiorów oraz skończonej liczby odwzorowań iloczynów kartezjańskich tych zbiorów w te zbiory. Odwzorowania wchodzące w skład struktury algebraicznej nazywamy działaniami.

Definicja 4. Grupa

to struktura algebraiczna (G, \cdot) , gdzie G jest zbiorem niepustym, a $\cdot : G \times G \rightarrow G$ odwzorowaniem spełniającym pierwsze trzy aksjomaty:

G.1 $\forall_{g_1, g_2, g_3 \in G} g_1 \cdot (g_2 \cdot g_3) = (g_1 \cdot g_2) \cdot g_3$ własność łączności,

G.2 $\exists_{e \in G} \forall_{g \in G} g \cdot e = e \cdot g = g$ gdzie e to element neutralny grupy,

G.3 $\forall_{g \in G} \exists_{g^{-1} \in G} g \cdot g^{-1} = g^{-1} \cdot g = e$ gdzie g^{-1} to element odwrotny do g ,

* jeśli dodatkowo $\forall_{g_1, g_2 \in G} g_1 \cdot g_2 = g_2 \cdot g_1$, to grupa jest abelowa (przemienna).

Definicja 5. Monoid

jest strukturą spełniająca tylko aksjomaty G.1 i G.2.

Definicja 6. Ciało

to struktura algebraiczna $(K, +, \cdot)$, gdzie $+$: $K \times K \rightarrow K$, \cdot : $K \times K \rightarrow K$ są odwzorowaniami spełniającymi aksjomaty:

K.1 $(K, +)$ jest grupą abelową,

K.2 $(K \setminus \{0\}, +)$ jest grupą abelową, gdzie 0 jest elementem neutralnym grupy $(K, +)$,

K.3 $\forall_{k_1, k_2, k_3 \in K} k_1 \cdot (k_2 + k_3) = (k_1 \cdot k_2) + (k_1 \cdot k_3)$.

Definicja 7. Przestrzeń wektorowa

nad ciałem K to struktura algebraiczna $(V, +; K, +, \cdot; \cdot)$, taka że

PW.1 $(V, +)$ jest grupą abelową,

PW.2 $(K, +, \cdot)$ jest ciałem,

działanie $\cdot : K \times V \rightarrow V$, gdy $k_1, k_2 \in K$, $\mathbf{v}, \mathbf{w} \in V$, oraz 1 jest jednością w K , to

PW.3 $k_1 \cdot (\mathbf{v} + \mathbf{w}) = k_1 \cdot \mathbf{v} + k_1 \cdot \mathbf{w}$, $(k_1 + k_2) \cdot \mathbf{v} = k_1 \cdot \mathbf{v} + k_2 \cdot \mathbf{v}$ (rozdzielność),

PW.4 $k_1 \cdot (k_2 \cdot \mathbf{v}) = (k_1 \cdot k_2) \cdot \mathbf{v}$ (łączność),

PW.5 $1 \cdot \mathbf{v} = \mathbf{v}$.

Definicja 8. Algebra

to przestrzeń wektorowa V nad ciałem K , gdzie oprócz działań V zdefiniowano mnożenie $\cdot : V \times V \rightarrow V$, takie że dla każdego $\mathbf{v}, \mathbf{w}, \mathbf{x} \in V$ i $k \in K$ zachodzi

A.1 $\mathbf{v} \cdot (\mathbf{w} \cdot \mathbf{x}) = (\mathbf{v} \cdot \mathbf{w}) \cdot \mathbf{x}$,

A.2 $\mathbf{v} \cdot (\mathbf{w} + \mathbf{x}) = \mathbf{v} \cdot \mathbf{w} + \mathbf{v} \cdot \mathbf{x}$, $(\mathbf{v} + \mathbf{w}) \cdot \mathbf{x} = \mathbf{v} \cdot \mathbf{x} + \mathbf{w} \cdot \mathbf{x}$,

A.3 $k \cdot (\mathbf{v} \cdot \mathbf{w}) = (k \cdot \mathbf{v}) \cdot \mathbf{w} = \mathbf{v} \cdot (k \cdot \mathbf{w})$.