

Politechnika Wrocławska

Wydział Elektroniki, Fotoniki i Mikrosystemów

KIERUNEK: Automatyka i Robotyka (AIR)

PRACA DYPLOMOWA INŻYNIERSKA

TYTUŁ PRACY:
Estetyka chodu robotów koczujących

AUTOR:
Kamil Witoń

PROMOTOR:
Dr inż. Robert Muszyński,
Katedra Cybernetyki i Robotyki

Streszczenie

Praca dotyczyła badania estetyki poruszania się robotów kroczących i podjęcia próby jej ilościowej oceny. Postawiono cel analizy i poprawy estetyki chodu sześcionożnego robota kroczącego poprzez modelowanie i optymalizację parametrów ruchu, w celu osiągnięcia bardziej płynnego i naturalnego sposobu poruszania się. Opracowano dwa modele robota: pierwszy – kinematyczny i analityczny, oraz drugi – dynamiczny i numeryczny. Zaprojektowano i zaimplementowano dla robota układ sterowania. Następnie przedstawiono sposób oceny estetyki, wskaźniki ilościowe oraz zdefiniowano struktury służące do optymalizacji chodu pod względem estetyki. Za pomocą algorytmów heurystycznych przeszukano przestrzeń parametrów chodu w celu znalezienia tych jak najbliższych optymalnym. Cel uznano za osiągnięty. Wyniki pracy wskazują na to, że możliwa jest ocena ilościowa estetyki chodu, jednak wymaga ona dalszych badań.

Słowa kluczowe: robot kroczący, estetyka, chód, gracjan, przestrzeń gracji

Abstract

This thesis investigates the aesthetics of walking robot locomotion and is an attempt to its quantitative assessment. The goal was to analyze and improve the aesthetics of a hexapod walking robot's gait by modeling and optimizing movement parameters to achieve a smoother and more natural gait. Two models of the robot were developed: the first one – kinematic and analytical, and the second one – dynamic and numerical. A control system was designed and implemented for the robot. Next, a method for aesthetic assessment, quantitative indicators, and structures for aesthetically optimizing gait were presented. Heuristic algorithms were used to search the gait parameter space to find those closest to optimal. The goal was achieved. The results indicate that quantitative assessment of gait aesthetics is possible, but this requires further research.

Keywords: walking robot, aesthetics, gait, gratian, grace space

#PROW12ORKA

Spis treści

1	Wstęp	3
2	Model robota	5
2.1	Model 3D robota w symulatorze MuJoCo	6
2.2	Model matematyczny robota	9
2.3	Model matematyczny nogi robota	10
3	Sterowanie	12
3.1	Zadanie sterowania	12
3.2	Struktura układu sterowania	15
3.3	Implementacja algorytmu sterowania robotem	17
3.3.1	Warstwa generowania trajektorii robota	17
3.3.2	Warstwa generowania chodu	17
3.3.3	Warstwa generowania trajektorii ruchu końców nóg	22
3.3.4	Warstwa generowania wartości nastaw w przegubach	28
3.3.5	Warstwa sterowania na poziomie wykonawczym	29
4	Estetyka ruchu	31
4.1	Ocena estetyki chodu	33
4.2	Wskaźniki estetyki	34
4.2.1	Zgrabność	34
4.2.2	Odporność na błędy	35
4.2.3	Zużycie energii	35
4.2.4	Finezja	35
4.2.5	Subtelność	36
4.2.6	Piękno ogólne	36
5	Badania	37
5.1	Sformułowanie zadania	37
5.2	Własności ilościowego opisu piękna	38
5.3	Sposób poszukiwań	39
5.4	Poszukiwanie piękna	40
5.4.1	Wpływ parametrów na gracjany	41
5.4.2	Wyniki poszukiwań algorytmów	47
5.5	Wnioski	52

Spis treści	2
6 Podsumowanie	53
6.1 Rola gracjanów w ocenie estetyki	54
6.2 Kierunki dalszych badań	54
6.3 Słowo końcowe	56
Literatura	57
Spis rysunków	60

Do składu pracy wykorzystano system przygotowania dokumentów $\LaTeX$, opracowany przez L. Lamport [Lam94], będący nakładką systemu $\TeX$, [Knu86a,Knu86b]. Matematyczne czcionki o nazwie AMS Euler, których używamy w tej pracy, zostały przygotowane przez H. Zapfa [KZ86], przy współpracy z D. Knuthem i jego studentami, na zlecenie Amerykańskiego Towarzystwa Matematycznego. Wybrane czcionki składu tekstu, Antykwa Toruńska [Now97] – jeden z nielicznych krojów pisma zaprojektowany specjalnie dla języka polskiego w sposób uwzględniający jego rytm – w odczuciu autora doskonale współgrają z kształtem czcionki AMS Euler, pozwalając na uzyskanie harmonijnej całości. Składu bezszeryfowego tekstu maszynowego dokonano z użyciem opracowanej przez R. Leviena czcionki o nazwie Inconsolata [Lev15].

- [Knu86a] D. E. Knuth, The $\TeX$ book, volume A of Computers and Typesetting. Addison-Wesley, Reading, 1986.
- [Knu86b] D. E. Knuth, $\TeX$: The Program, volume B of Computers and Typesetting. Addison-Wesley, Reading, 1986.
- [KZ86] D. E. Knuth i H. Zapf, AMS Euler — A new typeface for mathematics. Scholarly Publishing, 20:131–157, 1986.
- [Lam94] L. Lamport, $\LaTeX$: A Document Preparation System. Addison-Wesley, Reading, 1994.
- [Lev15] R. Levien, Inconsolata. <https://levien.com/type/myfonts/inconsolata.html>, 2015.
- [Now97] J. Nowacki, Antykwa Toruńska — od początku do końca polska czcionka. *Biuletyn Polskiej Grupy Użytkowników Systemu $\TeX$* , 9:26–27, 1997.

Rozdział 1

Wstęp

Ewolucja przez miliony lat wykształciła organizmy, które zachwycają nas swoim wyglądem, rozmiarem, wydawanymi odgłosami czy sposobem poruszania. Od geparda potrafiącego prześcignąć samochód terenowy po pajaka precyzyjnie tkającego sieć. Jeśli podczas ich obserwacji zastanowimy się dłużej, dojdziemy do wniosku, że cały proces stawiania kroku przez zwierzę został dopracowany i przetestowany przez niezliczone istnienia będące początkowo nieudolnymi drapieżnikami czy też ofiarami. W ten sposób najlepsze geny były przekazywane dalej a te, które nie zdały egzaminu – brutalnie eliminowane.

Wraz z efektywniejszym sposobem poruszania idzie w parze jego płynność i estetyka. Najłatwiej jest zauważyć to u młodych osobników, które uczą się chodzić. Żrebię chwilę po narodzinach stawia kroki nieporadnie, trzęsąc się i upadając raz po raz. Niewłaściwie umieszcza środek ciężkości swojego ciała, którego ruch przypomina bardziej serię zrywów niż płynne przejście z punktu do punktu. Jednak po dobie jest już w pełni gotowe do kroczenia za matką, a chód w pełni zasługuje na utrwalenie go na obrazie. Patrząc na dorosłego konia w pełnym galopie, trudno oprzeć się wrażeniu, że natura, kształtując jego dynamikę, kierowała się czystą estetyką. Zachwyt ten od wieków inspirował artystów, usiłujących uwiecznić majestatyczne piękno tego ruchu na statycznym płótnie czy w martwej skale.

W starożytności, kiedy ludzkość stawiała swoje pierwsze kroki na ścieżce nauki, najwięksi filozofie odkrywali fundamentalne prawa rządzące światem. Próbowali opisać istniejący świat, znaleźć reguły, które pozwolą im na przewidywania przyszłości czy zrozumieć tak prozaiczne sprawy jak wschody i zachody Słońca. Mimo że ich obserwacje i koncepcje naukowe nie były tak złożone i skomplikowane jak te tworzone współcześnie, nie ma wątpliwości, że stanowią one fundamentalne podwaliny dzisiejszych badań. To, co warto zauważyć to to, że już tak dawno temu wiązano ze sobą trzy rzeczy – piękno, prawdę i prostotę [Pla].

Tak też i dziś naukowcy tworząc hipotezy i załączki nowych teorii wychodzą od pięknych i prostych twierdzeń jak np. takich, że „żaden z układów odniesienia nie jest wyszczególniony” [Ein05b] czy „masa jest miarą energii” [Ein05a]. Idąc śladami największych myślicieli, mamy nadzieję, że w pogoni za pięknem uczynimy postęp w nauce i opiszemy jak ocenić w sposób możliwie obiektywny estetykę chodu robotów – tych sterowanych płynnie i tych bardziej impulsowo.

Celem pracy jest analiza i poprawa estetyki chodu sześcionożnego robota kroczącego poprzez modelowanie i optymalizację parametrów ruchu, w celu osiągnię-

cia bardziej płynnego i naturalnego sposobu poruszania się. W rozdziale drugim omówimy opracowane modele robota konieczne do zaimplementowania układu sterowania, którego schemat przedstawimy w rozdziale trzecim. Następnie w rozdziale czwartym przytoczymy historyczne próby opisu piękna i zdefiniujemy własne wskaźniki gracji chodu robota. W rozdziale piątym zostaną przytoczone własności ilościowego opisu piękna oraz opisane przeprowadzone badania. Rozdział szósty podsumowuje całość oraz przedstawia kierunki dalszych badań.

Rozdział 2

Model robota

Początki modelowania możemy odnaleźć już w czasach Arystotelesa [[Aryb](#)], kiedy powstawały proste modele rzeczywistości. Platońska koncepcja form i idei [[Pla](#)] idealnie oddaje sens modelowania. Świat rzeczywisty próbujemy opisać językiem idealnych idei – językiem matematyki. Dzięki temu problem możemy rozwiązać rygorystycznie i ściśle. Trzeba jednak pamiętać, że uzyskany w ten sposób model nie będzie mimo wszystko idealny. W 1976 roku George Box użył jakże trafnego stwierdzenia [[Box79](#)]

„Wszystkie modele są błędne, ale niektóre są użyteczne”.

Parafrazując, wszystkie modele to uproszczenia reprezentujące rzeczywistość, więc z natury są obarczone błędem pewnego uogólnienia.

Ze względu na sposób wybranego opisu, wykonania możemy wyszczególnić pewne rodzaje modeli. I tak, pierwszym z nich są modele analityczne, czyli takie, które przybliżają obiekt, relacje między jego częściami składowymi poprzez równania matematyczne. Zazwyczaj są to równania różniczkowe opisujące zmianę stanu układu. Ze względu na postać równań jesteśmy w stanie wyróżnić trzy najpowszechniejsze metody ich otrzymania, tj. poprzez:

1. opis sił, bazując na zasadach dynamiki Newtona [[New87](#)],
2. rachunek wariacyjny, wykorzystując podstawowe równania Eulera-Lagrange’a [[Wikc](#)],
3. badanie energii układu, w oparciu o zasadę najmniejszego działania Hamiltona [[Wikd](#)].

Idąc dalej, możemy wyszczególnić modele numeryczne. Wykorzystują one argument siły oraz w sposób skrajny czy nawet karykaturalny zasadę „dziel i zwyciężaj” zaprzęgając do pracy znaczące moce obliczeniowe komputerów. Bazują one na podziale złożonego układu na często wręcz atomowe części, które mogą być łatwiej zamodelowane. W tym miejscu możemy przytoczyć Metodę Elementów Skończonych (MES) [[Wikb](#)], która polega na rozczłonkowaniu większego tworzywa na dyskretny zbiór mas elementarnych i opisie relacji między nimi poprzez interpolacje prostymi funkcjami np. liniowymi.

Oczywiście obie te techniki modelowania możemy wykorzystać równolegle, opracowując model hybrydowy. Na przykład, dla bardziej złożonych, nieliniowych

obszarów badanej konstrukcji możemy zastosować MES, a dla prostszych, liniowych – równania analityczne. W ten sposób redukujemy potrzebną moc obliczeniową i omijamy konieczność opisu zmian nieliniowych.

Istnieje również możliwość modelowania fizycznego poprzez wierne, rzeczywiste odwzorowanie obiektu. Jest to częstą praktyką w przemyśle kosmicznym, gdzie buduje się bliźniaki maszyn wysyłanych w przestrzeń gwiazdną, aby móc symulować ich zachowania na Ziemi. Dzięki temu zmniejszamy ryzyko uszkodzenia lub spowodowania nieodwracalnych zmian na obiekcie oddalonym od nas nawet o wiele set tysięcy kilometrów, ponieważ skutki naszych decyzji możemy obserwować na kopii.

Niekiedy odwzorowanie 1:1 może być nieopłacalne lub nawet niemożliwe. Ze względu na łatwiejsze wprowadzanie zmian czy manewrowanie obiektem tworzy się rzeczywisty model w powiększeniu lub pomniejszeniu. Pozwala to np. na badanie właściwości aerodynamicznych turbin wiatrowych w zamkniętych komorach laboratoryjnych, testowanie odporności budowli architektonicznych na wstrząsy czy przestrzenne wizualizacje bardzo małych struktur jak jądra komórkowe.

Możliwości modelu różnią się w zależności od celu, w jakim ten został stworzony. W naszym przypadku wykonamy dwa rodzaje modeli tego samego robota. Pierwszy model, symulacyjny wykorzystamy do testowania sterowania i badania zmian parametrów chodu na jego estetykę. Będzie on wymagał zaprojektowania kształtów brył składowych oraz nadania im właściwości fizycznych. Dodatkowo konieczne będzie zamodelowanie silników napędowych oraz czujników, z których będą pobierane dane o stanie robota. Dzięki takiemu podejściu otrzymamy obiekt badań bez konieczności jego fizycznej realizacji, zyskując jednocześnie możliwość łatwej modyfikacji konstrukcji. Zwolni nas to od potencjalnych problemów związanych z realizacją warstwy sprzętowej.

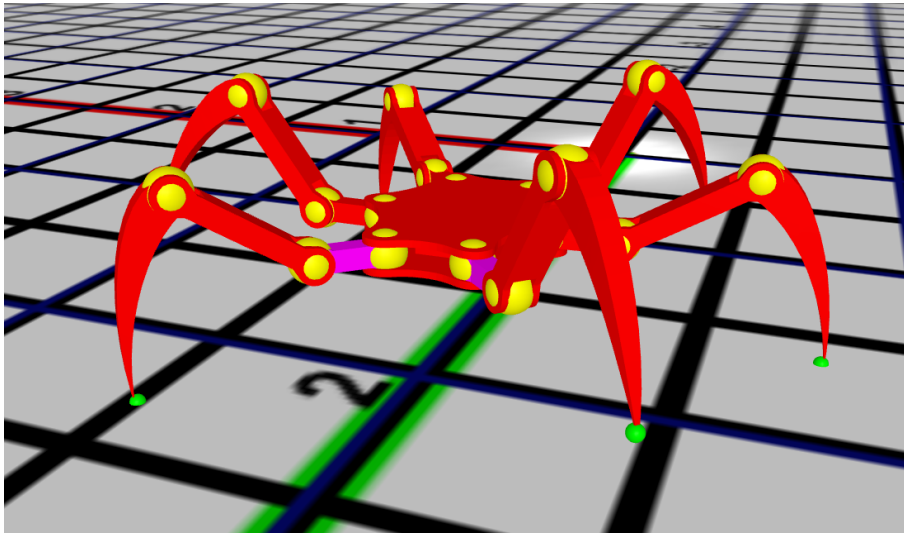
Drugi model, matematyczny, kinematyczny posłuży do opisu zależności pomiędzy konkretnymi częściami składowymi robota – członami nóg, korpusem – w zależności od aktualnych pozycji w przegubach. Całość będzie bazować na opisie poprzez transformacje między układami współrzędnych umieszczonych w punktach szczególnych robota. W rozdziale opiszemy wspomniane wcześniej modele.

2.1 Model 3D robota w symulatorze MuJoCo

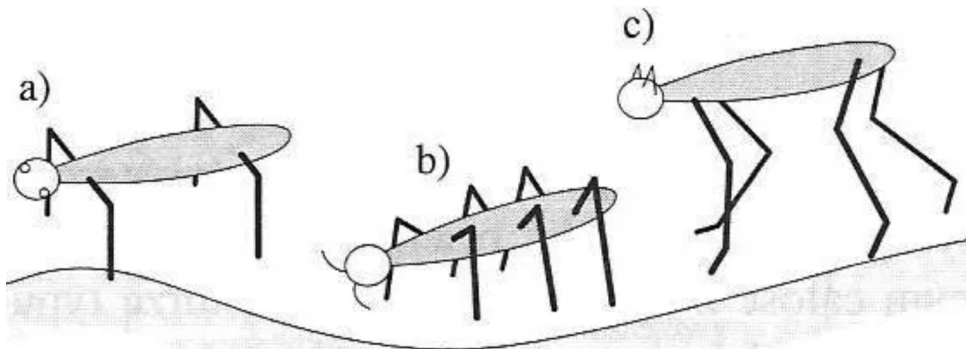
Robota (rys. 2.1) zaprojektowano jako heksapoda o układzie nóg owada (rys. 2.2b), gdzie staw kolanowy znajduje się powyżej płaszczyzny tułowia. Jego korpus ma kształt sześciokąta o wymiarach 3,5 na 3,0 m. Zakładamy, że robot będzie poruszać się tak, aby korpus był usytuowany równolegle do podłoża.

Jako wirtualne środowisko symulacyjne wybrano uniwersalny silnik fizyczny MuJoCo [Muj], który poprzez język Python [Phy] pozwala na proste modelowanie robota, sterowanie nim i odczyt czujników. Do zamodelowania części składowych robota użyto oprogramowania Blender [Ble], które umożliwia projektowanie brył trójwymiarowych.

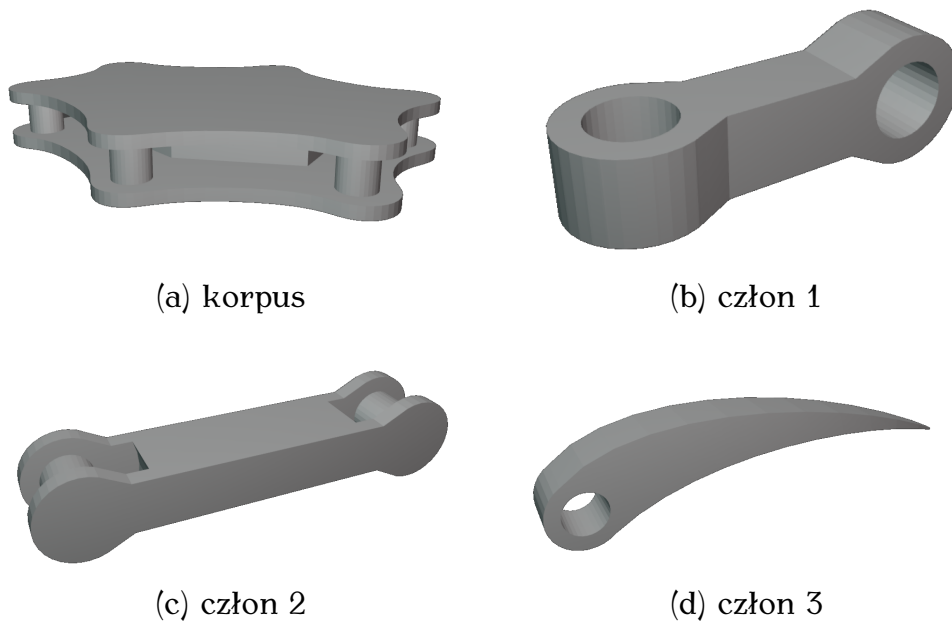
Zaprogramowanie modelu robota wymagało wcześniejszego zaprojektowania jego kształtu. W tym celu opracowano poszczególne elementy konstrukcyjne – korpus i trzy człony nogi – przedstawione na rys. 2.3. Na korpusie umieszczono sześć miejsc montowania nóg, usytuowanych symetrycznie. Współrzędne ich położenia



Rysunek 2.1: Model robota w symulacji MuJoCo



Rysunek 2.2: Układy nóg (a) gada – staw kolanowy zgięty pod kątem $\frac{\pi}{2}$, (b) owada – staw kolanowy ponad płaszczyzną tułowia, (c) ssaka – płaszczyzna ruchu nogi równoległa do grzbietu [Zie14]



Rysunek 2.3: Części składowe robota

Tabela 2.1: Współrzędne przyłączeń nóg

i	r_i [m]	θ_i [deg]	x_i [m]	y_i [m]
1	1,200	-90,00	0,00	-1,20
2	1,569	-22,48	1,45	-0,60
3	1,569	22,48	1,45	0,60
4	1,200	90,00	0,00	1,20
5	1,569	157,52	-1,45	0,60
6	1,569	-157,52	-1,45	-0,60

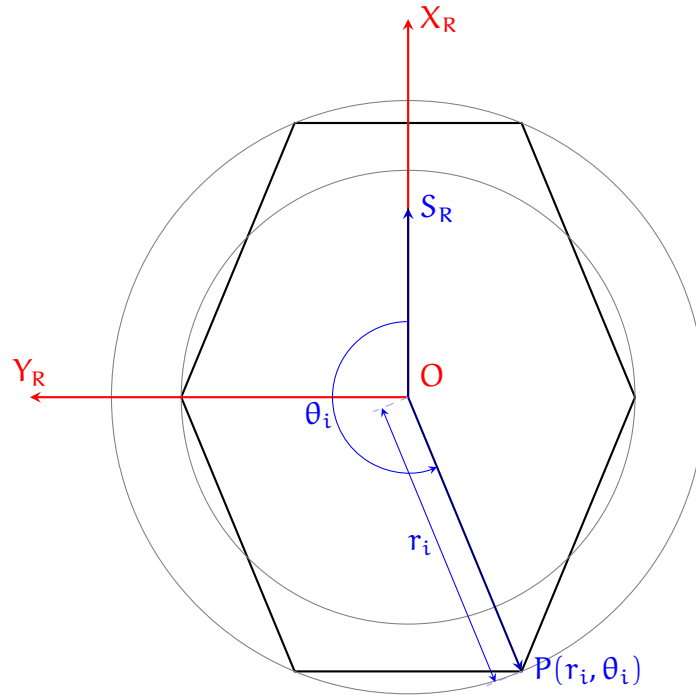
Tabela 2.2: Długości członów robota

Nr członu	Długość [m]
1	1,0
2	2,0
3	2,7

dla układu biegunowego i kartezjańskiego robota przedstawionych na rys. 2.4 zebrano w tabeli 2.1.

Elementy składowe zestawione są ze sobą za pomocą połączeń sworzniowych, które umożliwiają ruch obrotowy dwóch sąsiadujących części względem siebie. W tabeli 2.2 zebrano odległości pomiędzy środkami połączeń dla poszczególnych członów (dla członu trzeciego – środkiem połączenia a końcem członu). Wszystkim częściom składowym przypisano masę $m = 0,2$ kg.

Do napędu przegubów robota wykorzystano serwomechanizmy pozycyjne zaimplementowane w środowisku MuJoCo [Muj]. Na pracę serwomechanizmów można wpływać przez zmianę ich parametrów: wzmocnienia błędu k_p , tłumienie k_v , zakres dopuszczalnych pozycji $ctrlrange$ oraz zakres dopuszczalnych sił $forcerange$.



Rysunek 2.4: Współrzędne biegunowe dla korpusu robota

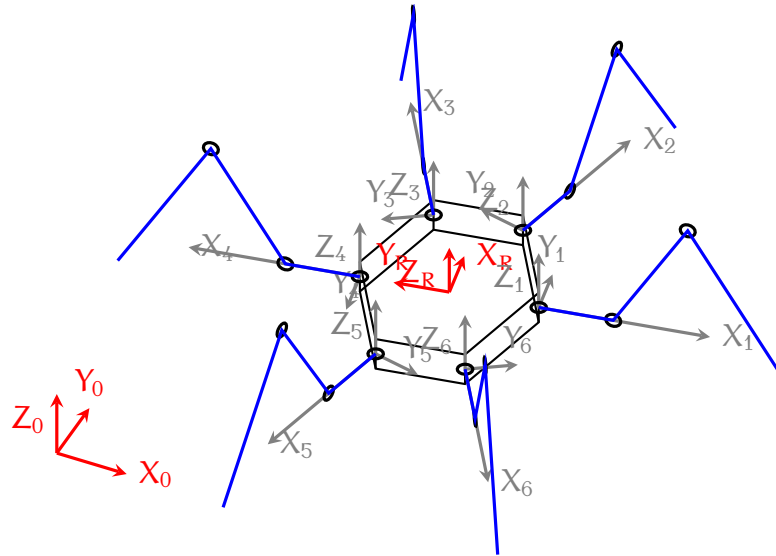
Tabela 2.3: Wartości parametrów serwomechanizmów w przegubach

i	kp	kv	ctrlrange	forcerange
1	150	30	(-0,785; 0,785)	(-50; 50)
2	100	20	(0,000; 2,618)	(-40; 40)
3	80	16	(-2,618; 0,000)	(-30; 30)

Wartości tych parametrów dla poszczególnych serwomechanizmów zastosowanych w przegubach robota zebrano w tabeli 2.3. Model robota w symulacji przedstawiono na rys. 2.1.

2.2 Model matematyczny robota

Z korpusem robota związany układ współrzędnych robota $X_R Y_R Z_R$, w taki sposób, że jego początek znajduje się w geometrycznym środku korpusu, oś $O_R X_R$, skierowana jest w przód, a oś $O_R Z_R$ – w górę. Z każdą nogą powiązano lokalny układ współrzędnych nogi $X_i Y_i Z_i$, $i \in 1, \dots, 6$. Na rys. 2.5 pokazano w sposób schematyczny rozmieszczenie układów współrzędnych wykorzystanych do przygotowania modelu matematycznego robota. Układ współrzędnych świata $X_0 Y_0 Z_0$ umieszczamy tak, że wektor grawitacji działa wzdłuż osi OZ_0 ze zwrotem do niej przeciwnym.



Rysunek 2.5: Schematyczny model robota z przypisanymi układami współrzędnych

Macierz przejścia z układu świata do układu robota wygląda następująco

$$A_0^R = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 & x \\ \sin(\psi) & \cos(\psi) & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.1)$$

gdzie ψ to orientacja robota względem osi Z_0 (kąt pomiędzy osiami OX_0 a OX_R). Macierz przejścia z układu robota do układu i -tej nogi ma postać

$$A_R^i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & r_i \cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) & 0 & r_i \sin(\theta_i) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.2)$$

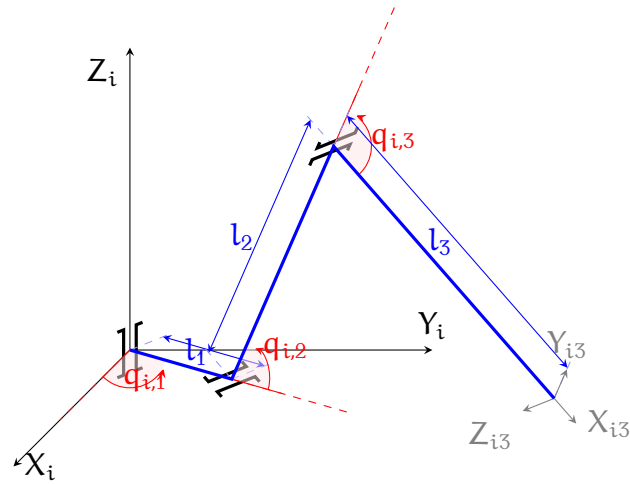
gdzie r_i oraz θ_i to współrzędne punktu przyłączenia nogi do korpusu w układzie biegunowym robota (rys. 2.4).

2.3 Model matematyczny nogi robota

Każdą z nóg robota potraktowano jako manipulator o trzech przegubach obrotowych, opisanych współzrzednymi $\mathbf{q} = (q_1, q_2, q_3)^T$, połączonych członami o długościach odpowiednio l_1, l_2, l_3 (rys. 2.6). Kinematykę nogi można opisać za pomocą notacji Denavita-Hartenberga (tab. 2.4) [HD64]. Pozwala to wyliczyć kinematykę pojedynczej nogi robota opisanej we współzrzednych kartezjańskich układu nogi $X_i Y_i Z_i$

$$\mathbf{k}(\mathbf{q}) = \begin{bmatrix} -l_3 s_2 s_3 c_1 + l_3 c_1 c_2 c_3 + l_2 c_1 c_2 + l_1 c_1 \\ -l_3 s_1 s_2 s_3 + l_3 s_1 c_2 c_3 + l_2 s_1 c_2 + l_1 s_1 \\ l_3 s_2 c_3 + l_2 s_2 + l_3 s_3 c_2 \end{bmatrix}, \quad (2.3)$$

gdzie s_i to $\sin(q_i)$ a $c_i = \cos(q_i)$.



Rysunek 2.6: Schemat kinematyczny nogi robota

Tabela 2.4: Parametry Denavita-Hartenberga dla nogi robota

i	θ_i	d_i	a_i	α_i
1	q_1	0	l_1	$\frac{\pi}{2}$
2	q_2	0	l_2	0
3	q_3	0	l_3	0

Zadanie kinematyki odwrotnej dla nogi rozwiązano przez zastosowanie metody Newtona-Raphsona [Ypm95]. Pojedynczy krok k algorytmu można opisać zależnością

$$q_{k+1} = q_k + J^{-1}(q_k)e, \quad (2.4)$$

gdzie $e = k(q) - x_d$ – błąd położenia końca odnóża, z x_d oznaczającym jego pozycję zadaną, zaś $J^{-1}(q)$ jest odwrotnością macierzy Jakobiego $J(q) = \frac{\partial k(q)}{\partial q}$ nogi robota.

Rozdział 3

Sterowanie

Okolo 2600 r. p.n.e. rządy w starożytnym Egipcie objęła IV dynastia faraonów, z których drugim, jednym z najbardziej znanych, był Cheops. Przyjmuje się okres jego panowania za czasy największej świetności państwa i to właśnie wtedy powstała najwyższa piramida na świecie nazwana na jego cześć. Za jej budowę miał być odpowiedzialny bezpośrednio wezyr Hemiunu. Jak podaje Herodot z Halikarnasu [Her59], przy budowie piramidy pracowało do 100 tysięcy ludzi. Jest to bez wątpienia przykład potęgi ludzkiej współpracy, która nie mogła się obyć bez równie wspaniałego zarządzania zasobami, czyli z naszego punktu widzenia sterowania. Sterowania obiektem o wieluset tysiącach stopni swobody i dość dużej stałej czasowej. Jednak pomimo tak wielkich osiągnięć, dopiero po 4000 lat powstała dziedzina opisująca od strony naukowej takie zjawiska – cybernetyka – która została zdefiniowana przez Norberta Wienera w następujący sposób [Wie48]:

„Cybernetyka czyli sterowanie i komunikacja w zwierzęciu i maszynie”.

W rozdziale opiszemy układ sterowania kinematycznego sześcionożnego robota kroczącego, łączącego cechy zwierzęcia i maszyny.

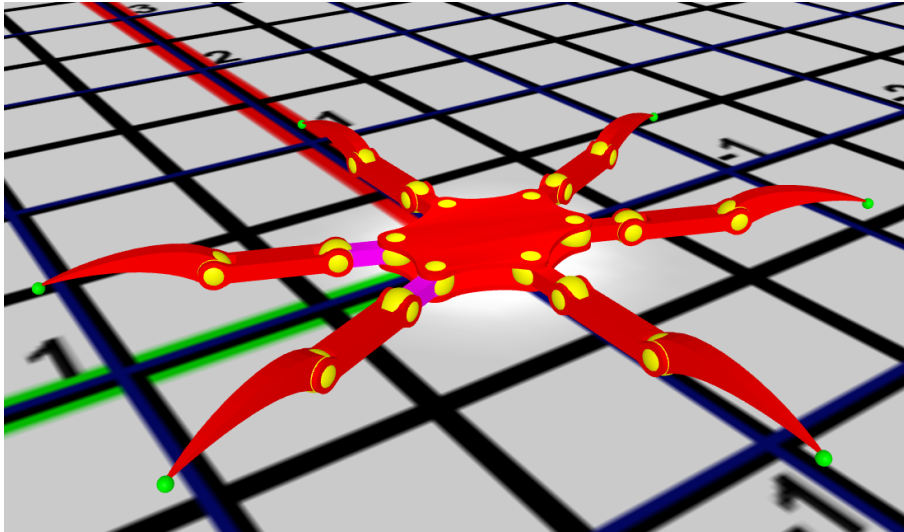
3.1 Zadanie sterowania

Naszym celem jest, aby ruch korpusu odbywał się z zadaną prędkością, która w układzie świata ma postać

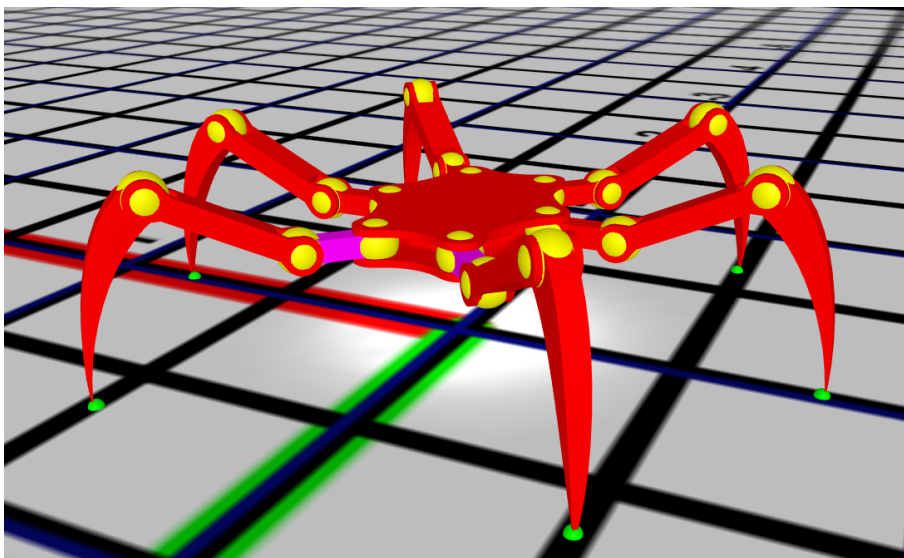
$$v_d = \begin{pmatrix} v_x \\ 0 \\ 0 \end{pmatrix}, \quad (3.1)$$

gdzie $v_x = \text{const}$ i $v_x > 0$. Zakładamy, że robot w chwili początkowej $t = t_0$ znajduje się w pozycji leżącej na powierzchni X_0Y_0 o orientacji $\psi = 0$ z przegubami każdej z nóg w pozycjach odniesienia $q = (0, 0, 0)^T$. Pozycję tę będziemy nazywać pozycją zerową, przedstawiono ją na rys. 3.1. Robot, aby rozpocząć chód, powinien podnieść korpus na zadaną wysokość h_{rob} i przejść tym samym do pozycji stojącej. Zdefiniujemy ją jako pozycję startową (rys. 3.2), z której robot rozpoczyna chód. Przyjęte współrzędne końców nóg w układzie robota dla tej pozycji zebrano w tab. 3.1.

Chód jest to sposób poruszania się poprzez stawianie kroków nie tracąc przy tym kontaktu z podłożem. Dodatkowo, aby nie przewrócić się, powinniśmy stale utrzymywać stabilność. Ze względu na jej rodzaj możemy wyróżnić chody [Zie14]:



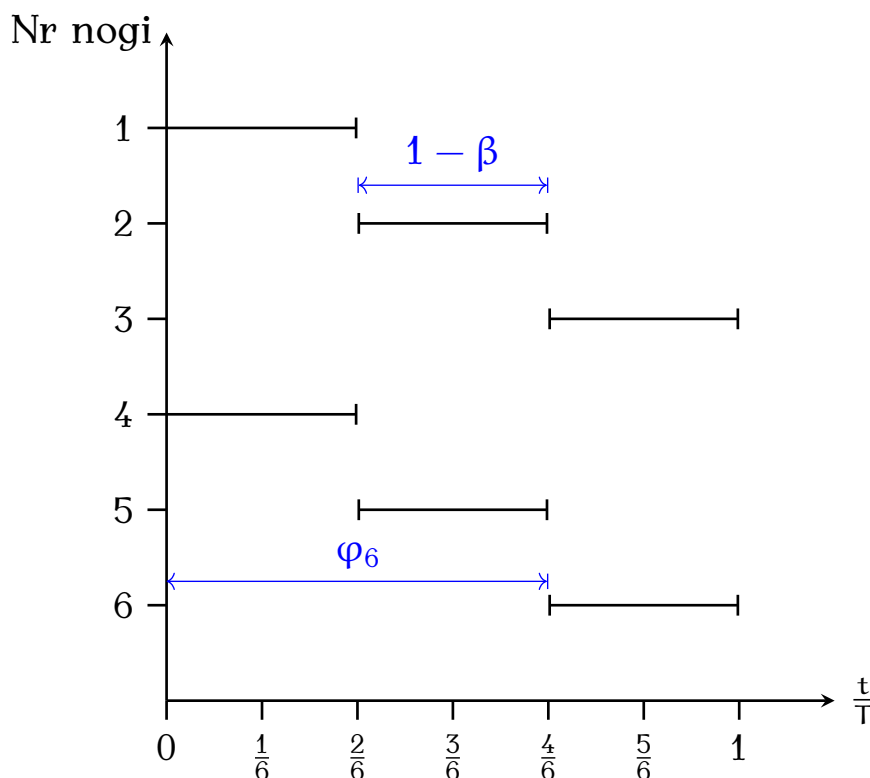
Rysunek 3.1: Robot w pozycji zerowej



Rysunek 3.2: Robot w pozycji startowej

Tabela 3.1: Współrzędne końców nóg w układzie $X_R Y_R Z_R$

Nr nogi	Współrzędne punktu (x_i, y_i, z_i)		
1	$(0, 0;$	$-4, 5;$	$-h_{rob})^T$
2	$(3, 0;$	$-3, 0;$	$-h_{rob})^T$
3	$(3, 7;$	$3, 0;$	$-h_{rob})^T$
4	$(0, 0;$	$4, 5;$	$-h_{rob})^T$
5	$(-3, 7;$	$3, 0;$	$-h_{rob})^T$
6	$(-3, 7;$	$-3, 0;$	$-h_{rob})^T$



Rysunek 3.3: Diagram chodu dla chodu czwórpodporowego

- **stabilne statycznie** – kiedy rzut środka ciężkości robota na wielokąt podparcia (rozpięty poprzez punkty kontaktu nóg z podłożem) leży w jego wnętrzu w trakcie całego chodu,
- **stabilne dynamicznie** – które nie spełniają warunku powyżej, a mimo to pozwalają na ruch bez upadku.

W pracy ograniczymy się jedynie do chodu stabilnego statycznie, który pozwala przemieszczać korpus robota w płaszczyźnie równoległej do płaszczyzny X_0Y_0 przy zachowaniu jego stałej orientacji.

Analizując chód wyodrębnić możemy w nim pojedynczy cykl, który należy powtarzać. W zależności od liczby nóg n , które jednocześnie podpierają korpus, wyróżniamy chody n -podporowe. Każdy z tych chodów możemy reprezentować poprzez diagram przedstawiający okres przenoszenia każdej nogi w funkcji upływającego czasu. Przykładowy diagram dla chodu czwórpodporowego przedstawiono na rys. 3.3. Charakteryzuje się on dwoma parametrami:

- **współczynnikiem wypełnienia chodu** β – który jest stosunkiem czasu protrakcji do retrakcji,
- **przesunięciem fazowym** φ_i – czyli czasem pomiędzy rozpoczęciem fazy protrakcji i -tej a pierwszej nogi.

W pracy ograniczymy się do chodu trój-, czwór- i pięciopodporowego.

Podczas stawiania kroku możemy wyszczególnić dwie główne fazy [Zie14]:

- **retrakcji** – kiedy noga podpira korpus,
- **protrakcji** – kiedy przenosimy ją nad powierzchnią ruchu.

W pierwszej z faz zadanie polega na śledzeniu punktu podparcia, którego współrzędne są stałe w układzie świata $X_0Y_0Z_0$ (rys. 2.5) natomiast ze względu na przemieszczanie korpusu zmienne w układzie robota $X_RY_RZ_R$. W celu wyznaczenia nastaw w przegubach nóg będziemy dokonywać serii transformacji położenia z układu świata do układu robota, dalej do układu nogi i ostatecznie do układu przestrzeni przegubowej.

W drugiej z faz należy wyznaczyć trajektorię przenoszenia końca nogi z aktualnej pozycji do oddalonej o długość kroku. Tutaj pojawia się pewna dowolność, co do kształtu trajektorii ruchu końca nogi. Zasadniczo można wybrać dowolną funkcję, która będzie zaczynać się i kończyć w punktach podniesienia i opuszczenia nogi oraz zwracać na tym przedziale (przenoszenia końca nogi) punkty, których współrzędna z jest większa od 0.

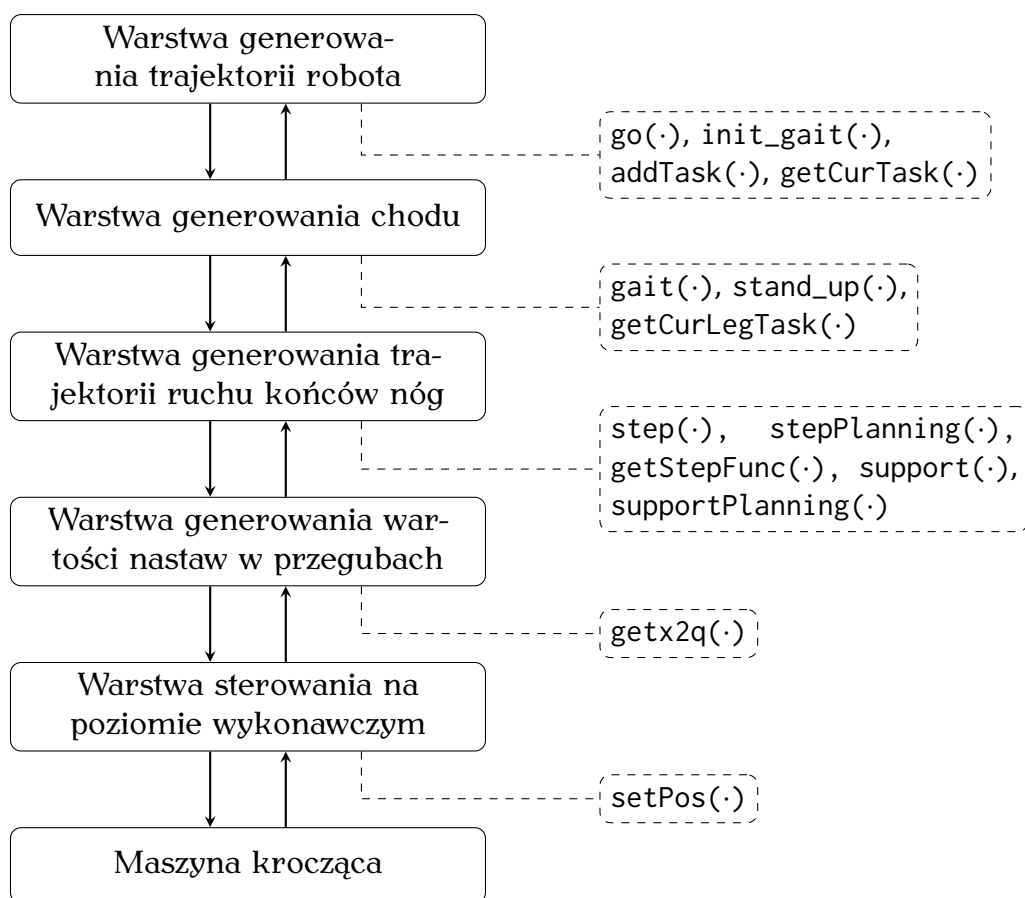
Zatem, kompletne zadanie sterowania będzie polegać na przejściu z pozycji zerowej do startowej – zadanie wstępne, a następnie poruszaniu się z zadaną prędkością liniową v_d w płaszczyźnie X_0Y_0 bez zmiany orientacji $\psi = 0$ – zadanie właściwe.

3.2 Struktura układu sterowania

W układzie sterowania można wyróżnić warstwy zgodnie z podziałem przedstawionym w [Zie14]. Są to, poczynając od najwyższej (rys. 3.4):

1. **Warstwa generowania trajektorii ruchu robota.** Obliczana jest tu pozycja zadana korpusu robota dla aktualnego czasu, konieczna dla realizacji zadań niższych warstw.
2. **Warstwa generowania chodu.** Warstwa dysponuje zadaniami dla poszczególnych nóg robota – „przenieś nogę”, „podtrzymuj korpus”.
3. **Warstwa generowania trajektorii ruchu końców nóg.** W zależności od zleconego dla nogi zadania z warstwy wyższej, wyliczana jest bieżąca pozycja w jakiej powinien znajdować się koniec nogi robota.
4. **Warstwa generowania wartości nastaw w przegubach.** Za pomocą algorytmu Newtona-Raphsona i modelu kinematycznego nogi robota (przedstawionego w rozdz. 2) dla zadanych pozycji końców nóg wyliczane są nastawy w ich przegubach.
5. **Warstwa sterowania na poziomie wykonawczym.** Wartości zadane dla przegubów przekazywane są do lokalnych układów sterowania (aktuatorów) realizowanych jako sterowniki PID.
6. **Maszyna krocząca.** Zsynchronizowany ruch napędów w przegubach nadaje prędkość korpusu robota.

Na rys. 3.4 umieszczono dodatkowo nazwy funkcji programu realizujących zadania przypisane warstwom.



Rysunek 3.4: Struktura sterowania

3.3 Implementacja algorytmu sterowania robotem

Sterowanie zrealizowano w sposób iteracyjny. Główną funkcję algorytmu przedstawiono na rys. 3.5. Rozpoczynamy od fazy inicjacji, w której ustawiamy wartości zmiennych t , $last_update$, $control_t$, przekazujemy parametry chodu – metoda $init_gait(\cdot)$ oraz planujemy kolejność i czasy wykonywanych zadań – metoda $add_task(\cdot)$. Następnie w pętli głównej pobierany jest czas symulacji t_sym , według którego co okres $control_t$ wykonywana jest iteracyjnie metoda $go(\cdot)$ dla aktualnego czasu wywołująca funkcję z warstwy sterowania. Poniżej przedstawimy implementację algorytmu sterowania z podziałem na wyróżnione warstwy, poczynając od najwyższej.

3.3.1 Warstwa generowania trajektorii robota

Zadania warstwy generowania trajektorii robota realizowane są głównie poprzez funkcję $go(\cdot)$ – rys. 3.6, która nadzoruje wykonywanie zaplanowanych wcześniej dla robota zadań. Na wstępie poprzez wywołanie $getCurTask(\cdot)$ dla aktualnego czasu pobierane jest bieżące zadanie oraz chwila jego rozpoczęcia. W zależności od typu zadania wywoływane jest polecenie $stand_up(\cdot)$ lub $gait(\cdot)$ realizujące odpowiednio podnoszenie robota z płaszczyzny X_0Y_0 lub właściwy chód. Na koniec aktualizowany jest stan wewnętrzny robota – $update(\cdot)$.

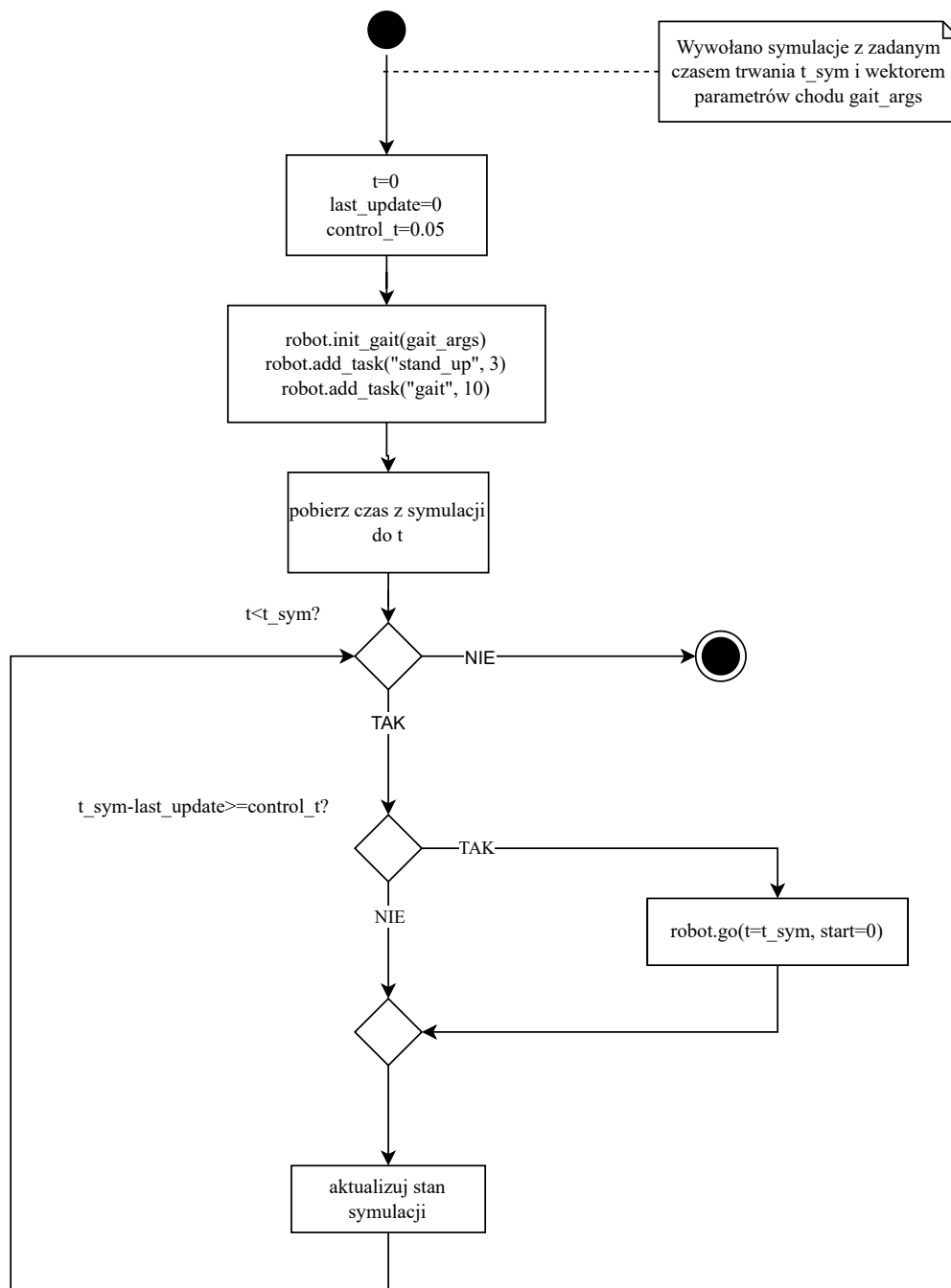
3.3.2 Warstwa generowania chodu

W warstwie generowania chodu możemy wyszczególnić dwie główne funkcje realizujące podnoszenie korpusu nad płaszczyznę X_0Y_0 (traktujemy to jako graniczny przypadek chodu sześciopodporowego) – $stand_up(\cdot)$ oraz generowanie właściwego chodu – $gait(\cdot)$. Schemat pierwszej z wymienionych przedstawiono na rys. 3.7. Jako parametry wejściowe przyjmuje ona aktualny czas t , chwilę rozpoczęcia zadania $start$ oraz czas przeznaczony na wykonanie zadania T . Ten dzielony jest na trzy równe przedziały, w których wykonywane są fazy podnoszenia się robota. I tak w fazie pierwszej nogi ustawiamy w konfiguracji $q = (0, \frac{2\pi}{3}, -\frac{2\pi}{3})^T$. W drugiej analogicznie w $q = (0, \frac{7\pi}{18}, -\frac{2\pi}{3})^T$. W trzeciej następuje właściwe podnoszenie korpusu. Tworzymy punkt pozycji zadanej końca i -tej nogi w układzie robota przyjmując współrzędne x_i i y_i z tabeli 3.1. Natomiast współrzędną z_i wyliczamy w następujący sposób

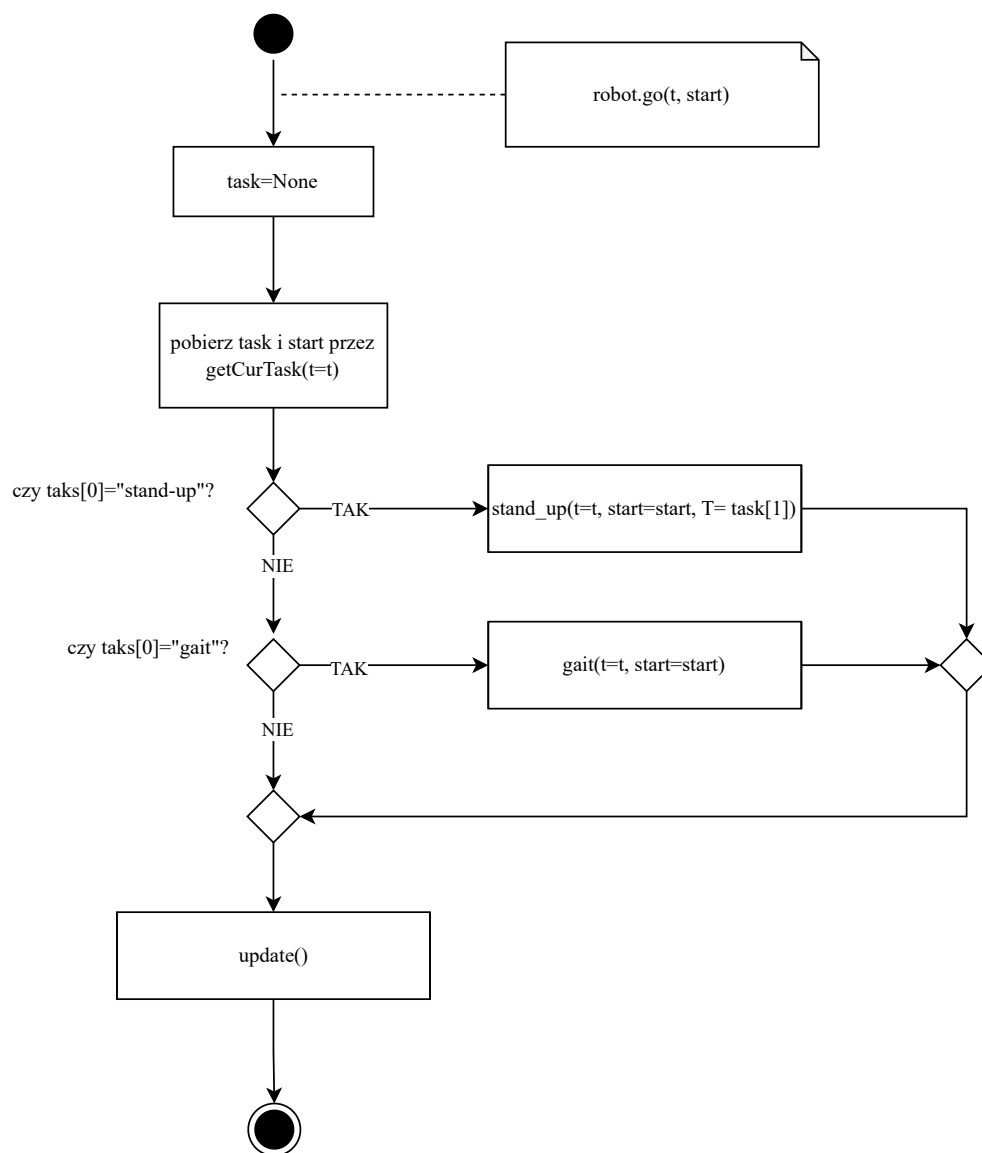
$$z_i = -h_{rob}(10\alpha^3 - 15\alpha^4 + 6\alpha^5), \quad (3.2)$$

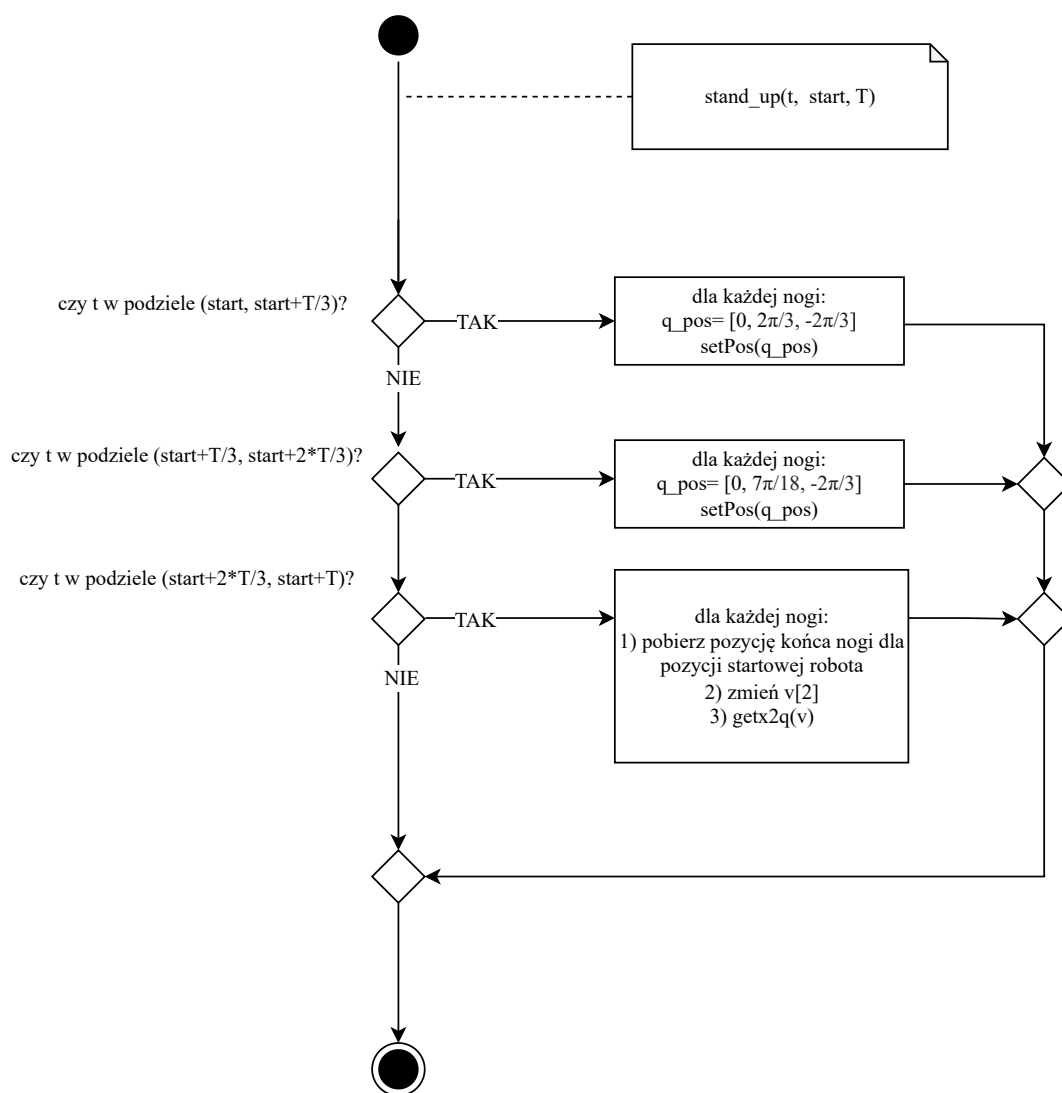
gdzie h_{rob} to zadana wysokość unoszenia korpusu nad powierzchnią X_0Y_0 , a parametr α wyliczamy jako $\alpha = \frac{3}{T} \cdot (t - (start + \frac{2T}{3}))$. Dla pozycji wywołujemy metodę $getx2q(\cdot)$, która oblicza właściwe nastawy w przegubach i ustawia według nich nogę.

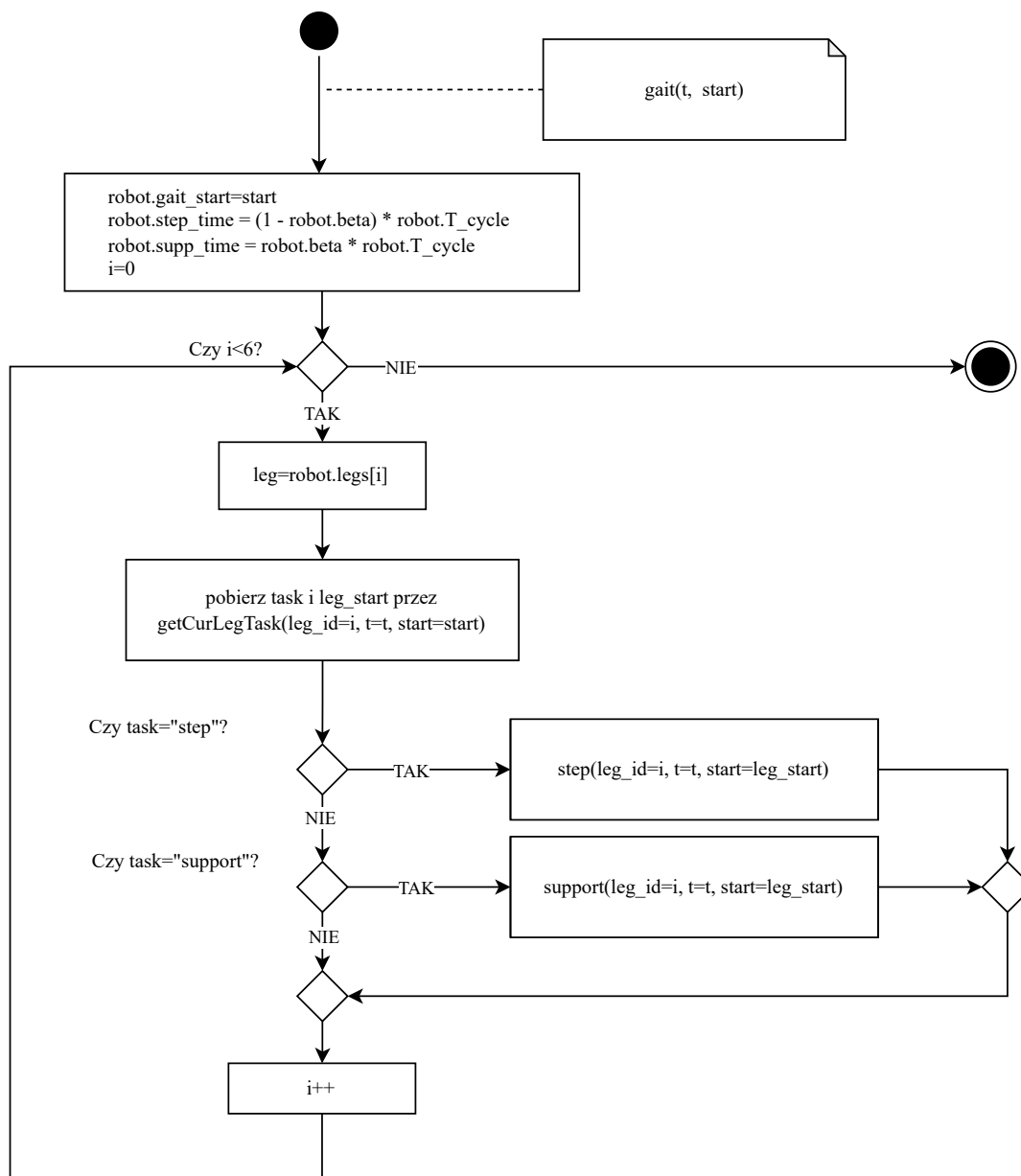
Schemat metody $gait(\cdot)$ przedstawiono na rys. 3.8. Funkcja ta jako swoje argumenty wywołania otrzymuje aktualną chwilę t oraz chwilę rozpoczęcia chodu $start$. Na początku odbywa się inicjalizacja zmiennych wewnętrznych robota tj. $gait_start$ – chwila rozpoczęcia chodu, $step_time$ – czas przeznaczony na przeniesienie nogi oraz $supp_time$ – czas przeznaczony na podpieranie korpusu. Będą one wykorzystywane w niższej warstwie. Następnie wykonywane są identyczne czynności dla każdej

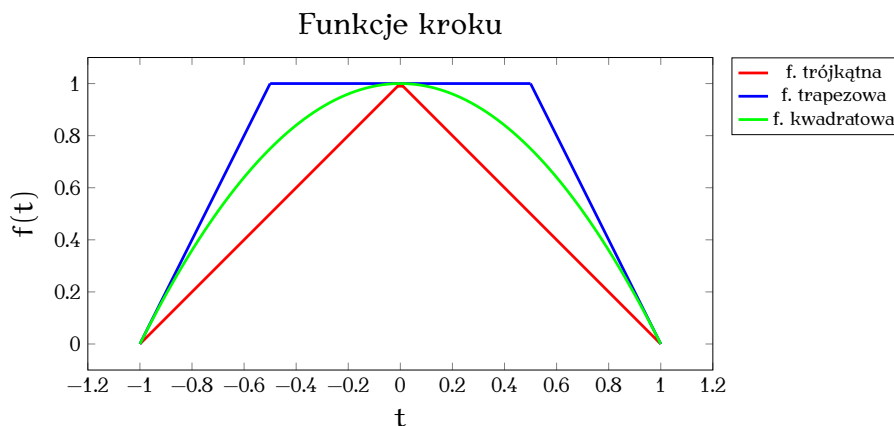


Rysunek 3.5: Główna pętla algorytmu sterowania

Rysunek 3.6: Metoda `go(.)`

Rysunek 3.7: Metoda `stand_up(·)`

Rysunek 3.8: Metoda `gait(·)`



Rysunek 3.9: Funkcje kroku

z nóg. Pobierane jest bieżące zadanie nogi poprzez `getCurTask(·)` wywołaną dla numeru aktualnej nogi, aktualnego czasu i chwili rozpoczęcia chodu. Jest to właściwa implementacja wspomnianego wcześniej diagramu chodu (rys. 3.3). W zależności od rodzaju zadania wywoływana jest funkcja `step(·)` – faza protrakcji bądź `support(·)` – faza retrakcji z niższej warstwy.

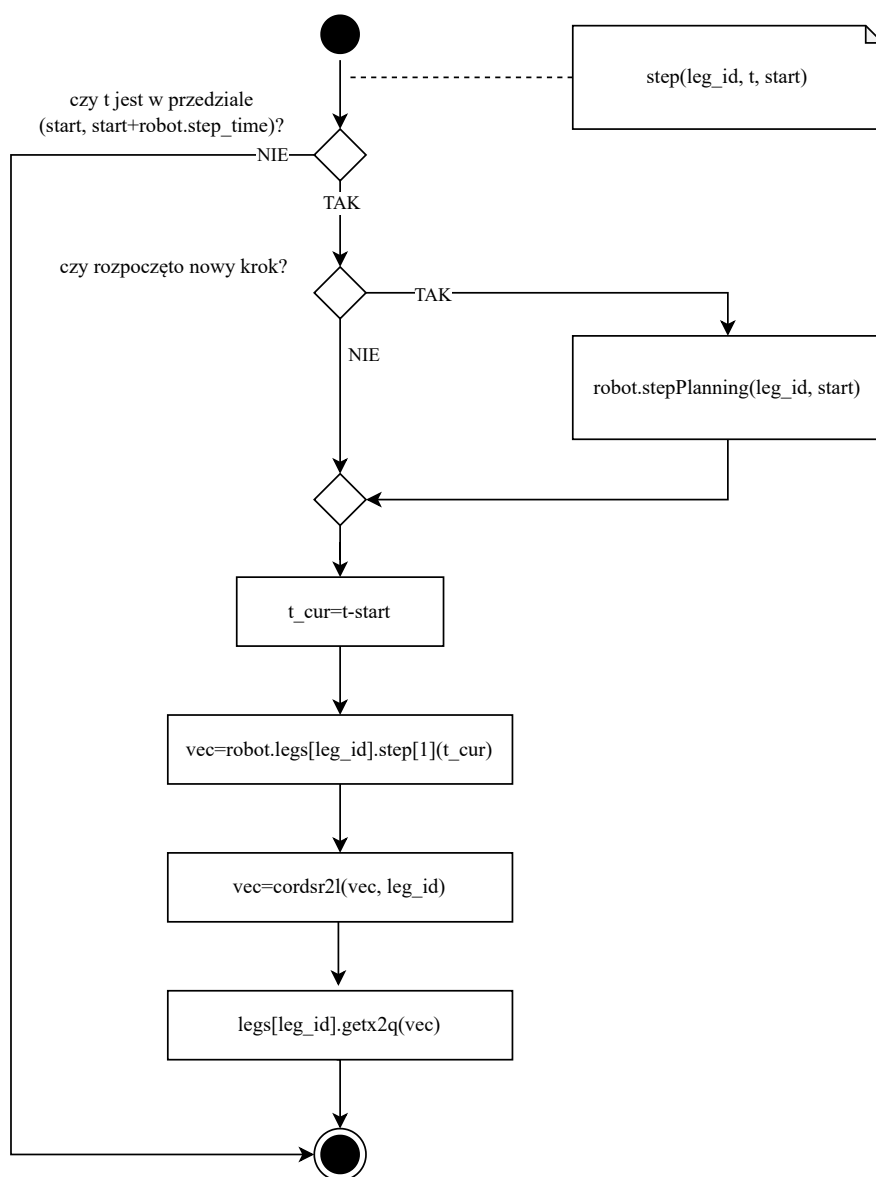
3.3.3 Warstwa generowania trajektorii ruchu końców nóg

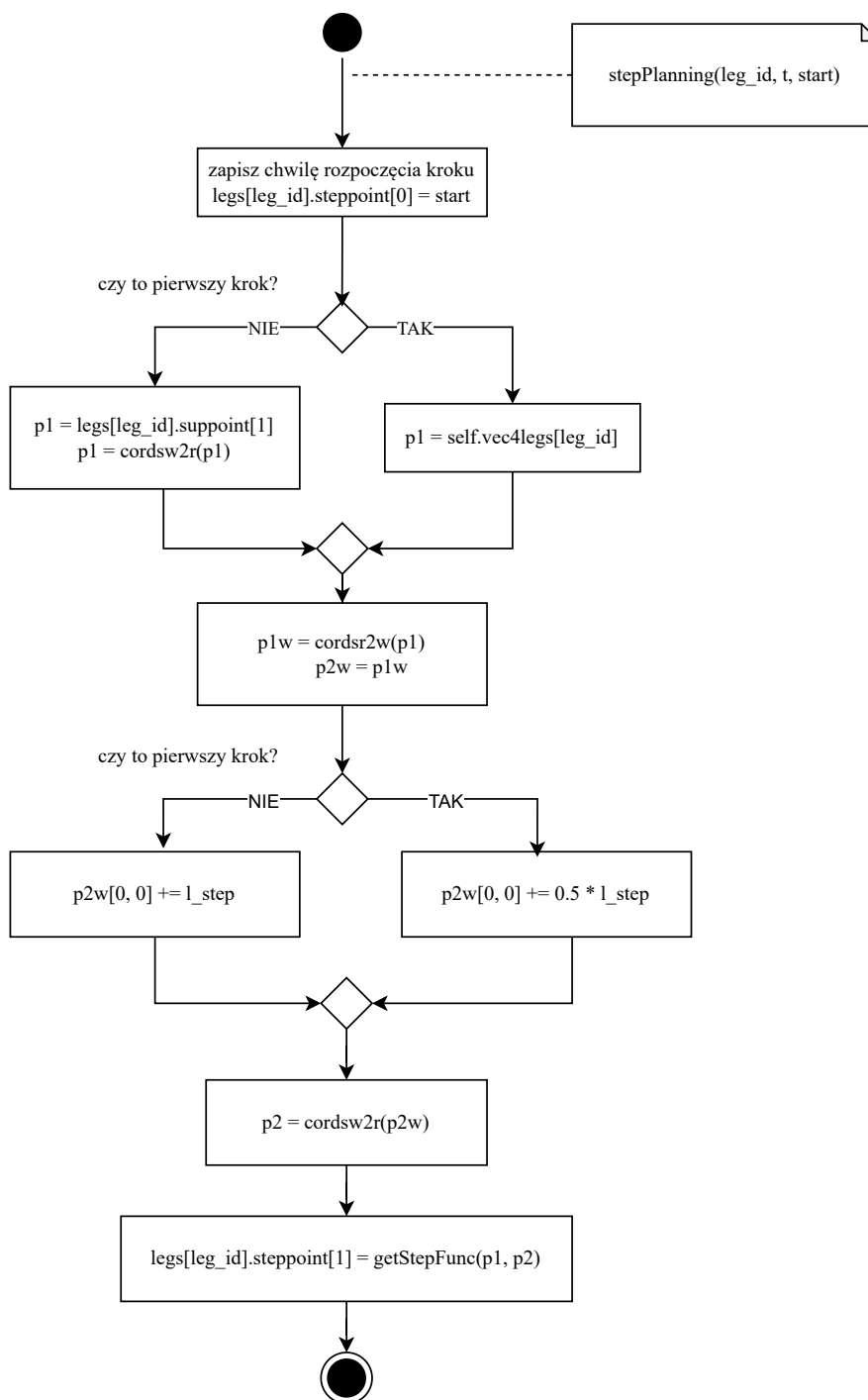
Do planowania trajektorii przenoszenia nogi nad powierzchnią chodu wybrano trzy funkcje (pokazane na rys. 3.9):

- trójkątną f_{tria} ,
- trapezową f_{trap} ,
- kwadratową f_{parab} .

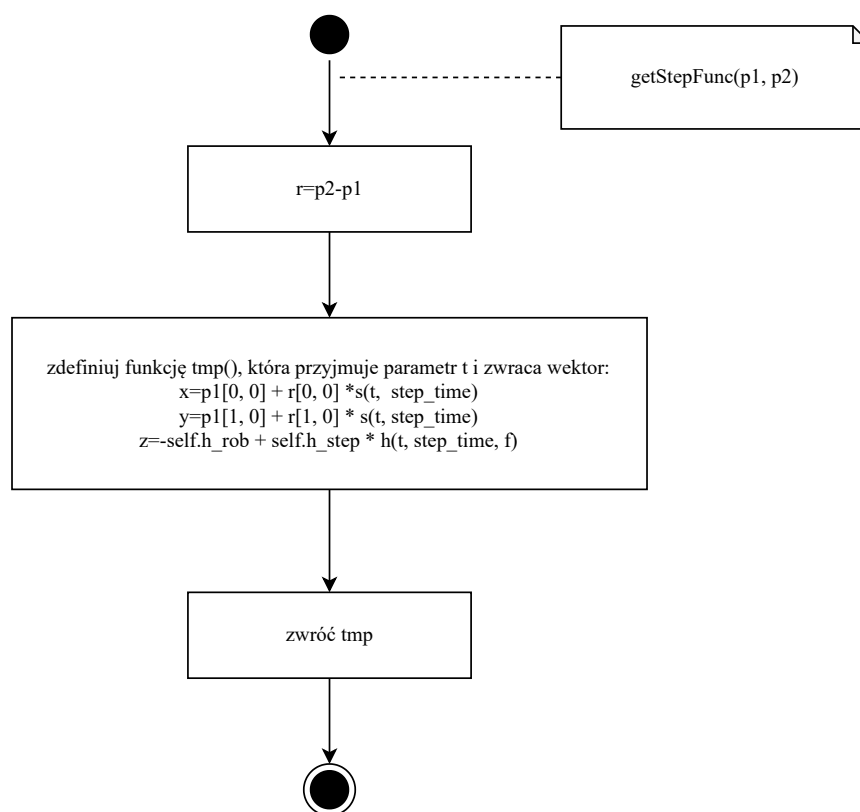
Fazę protrakcji zrealizowano poprzez zaimplementowanie funkcji `step(·)`. Jej schemat pokazano na rys. 3.10. Przyjmuje ona jako parametry wejściowe numer identyfikacyjny nogi `leg_id`, dla której wykonywana jest faza, aktualną chwilę `t` oraz czas rozpoczęcia start. Wykonuje się ona tylko, jeśli nie upłynął zaplanowany na nią czas. Jeśli rozpoczęto nowy krok, wywoływana jest funkcja planowania – `stepPlanning(·)`, która zapisuje zaplanowaną trajektorię jako funkcję w odpowiedniej zmiennej wewnętrznej. Dalej obliczany jest czas względny `t_cur` jako różnica chwili aktualnej i rozpoczęcia zadania. Następnie pobierany jest punkt z wcześniej zaplanowanej trajektorii końca nogi dla argumentu `t_cur` i zlecane jest zadanie dla warstwy niżej – wyliczenie i wysterowanie nastaw przegubów dla zadanego punktu (`getx2q(·)`).

Funkcję planowania – `stepPlanning(·)` przedstawiono na rys. 3.11. Przyjmuje ona argumenty w postaci numeru nogi `leg_id` oraz chwili rozpoczęcia fazy protrakcji start. Na początku zapamiętywana jest ta chwila w zmiennej wewnętrznej. Następnie pobierany jest punkt `p1` jako pozycja startowa nogi, jeśli to pierwszy krok. W przeciwnym razie jest to punkt zakończenia fazy retrakcji tej nogi. Dalej jego współrzędne transformowane są do układu świata. Punkt `p2` jest wyznaczany przez

Rysunek 3.10: Metoda `step(·)`



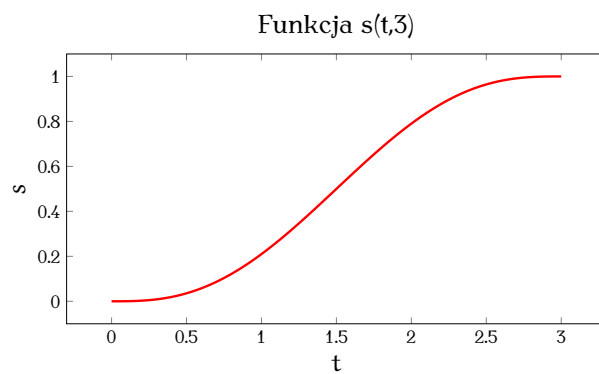
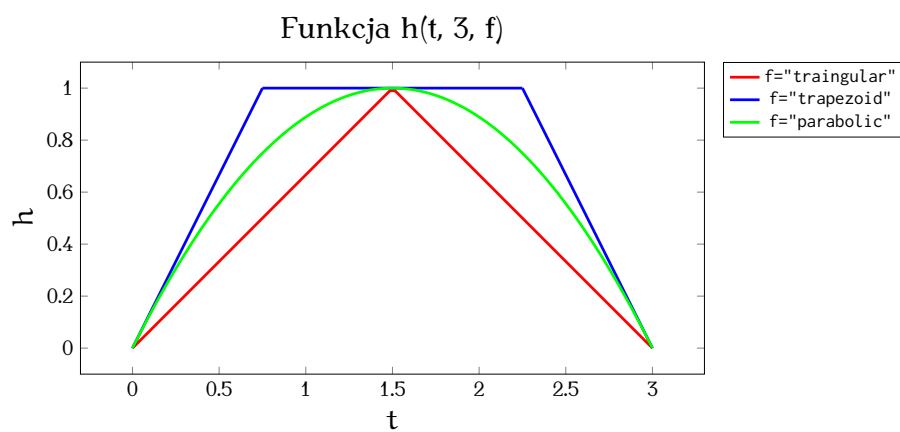
Rysunek 3.11: Metoda stepPlanning(·)

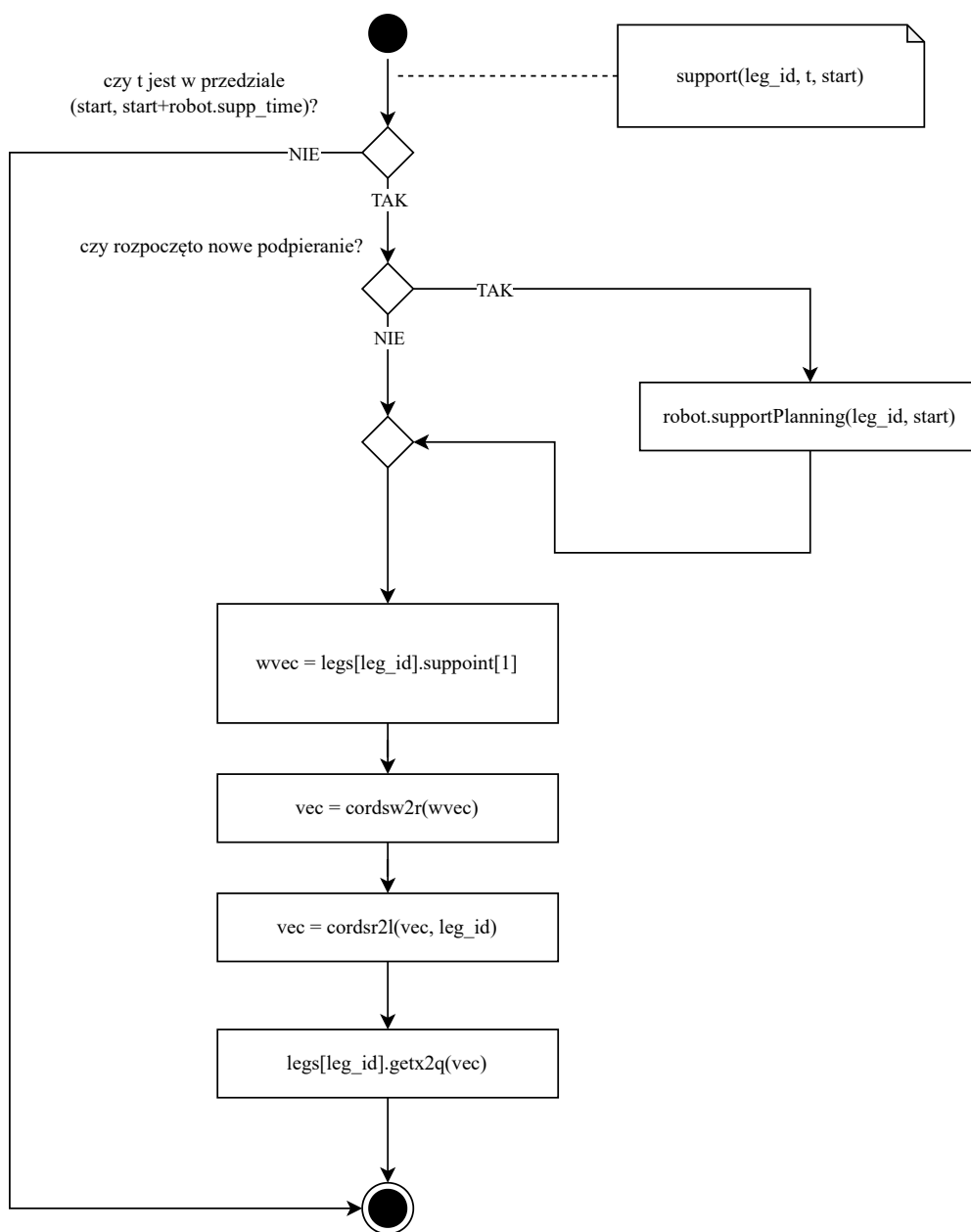
Rysunek 3.12: Metoda `getStepFunc(·)`

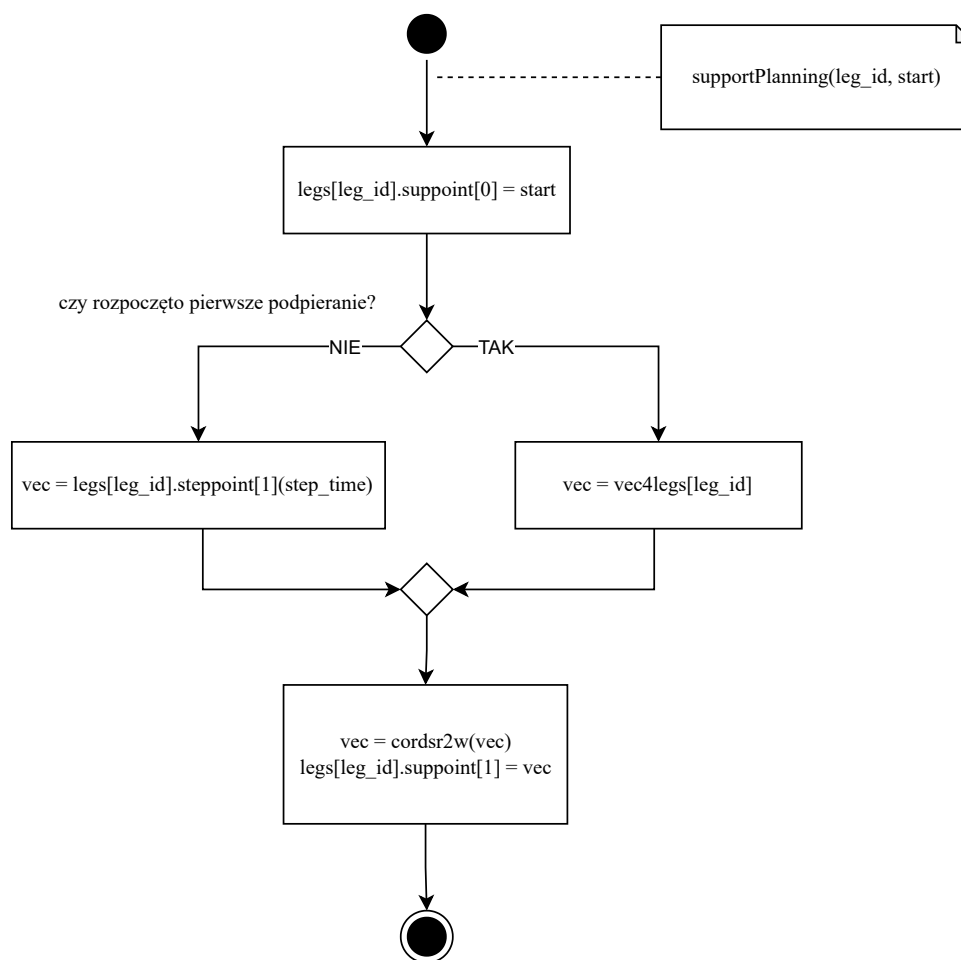
przesunięcie o długość kroku, lub pół tej długości dla kroku pierwszego, na osi OX_0 punktu $p1$ i następnie transformacji do układu robota $X_R Y_R Z_R$. Punkty te służą do wygenerowania funkcji trajektorii nogi. Zapisujemy ją w odpowiedniej zmiennej wewnętrznej.

Poprzez funkcję `getStepFunc(·)`, której schemat przedstawiono na rys. 3.12, realizowane jest wyznaczanie trajektorii przenoszenia nogi. Wyliczany jest wektor r między $p2$ a $p1$. Następnie definiowana jest funkcja wektorowa zwracająca kolejne pozycje nogi w układzie robota dla czasu jaki upłynął od rozpoczęcia bieżącej fazy protrakcji. Wykorzystywane są tu pomocnicze funkcje znormalizowane $[0, T] \rightarrow [0, 1]$. Graficzne odwzorowanie $s(t, T)$ oraz $h(t, T, f)$ przedstawiono odpowiednio na rys. 3.13 oraz 3.14.

Funkcja z fazy retrakcji – `support(·)` – realizowana jest podobnie do `step(·)`. Jej schemat przedstawiono na rys. 3.15. Przyjmuje ona jako dane wejściowe numer nogi `leg_id`, dla której wykonywana jest faza, aktualną chwilę t oraz chwilę rozpoczęcia start. Funkcja `support(·)` wykonuje się tylko, jeśli nie upłynął zaplanowany na nią czas. Jeśli rozpoczęto podpieranie dla nowego punktu, wywoływana jest funkcja planowania – `supportPlanning(·)`, która zapisuje punkt podpierania w odpowiedniej zmiennej wewnętrznej. Punkt ten jest zapisywany w zmiennej lokalnej. Następnie wykonywana jest transformacja jego współrzędnych z układu świata do układu nogi. Na koniec wywoływana jest funkcja `getx2q(·)` niższej warstwy w celuysterowania

Rysunek 3.13: Funkcja $s(t, T)$ dla $T=3$ Rysunek 3.14: Funkcja $h(t, T, f)$ dla $T=3$

Rysunek 3.15: Metoda `support(·)`



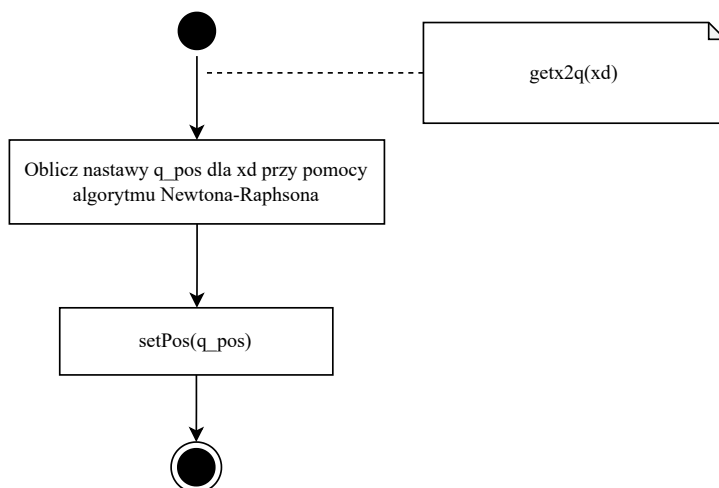
Rysunek 3.16: Metoda supportPlanning(·)

nogi do zadanego punktu.

Wspomniana funkcja supportPlanning(·) przyjmuje jako argumenty numer nogi leg_id oraz chwilę rozpoczęcia fazy start. Jej schemat przedstawiono na rys. 3.16. Na początku zapamiętywana jest chwila rozpoczęcia fazy. Następnie pobierany jest punkt końca nogi w pozycji startowej lub w chwili zakończenia fazy protrakcji, w zależności od tego czy rozpoczęto pierwszą fazę retrakcji dla tej nogi. Na koniec następuje obliczenie współrzędnych punktu w układzie świata i zapisanie pozycji w odpowiedniej zmiennej wewnętrznej robota.

3.3.4 Warstwa generowania wartości nastaw w przegubach

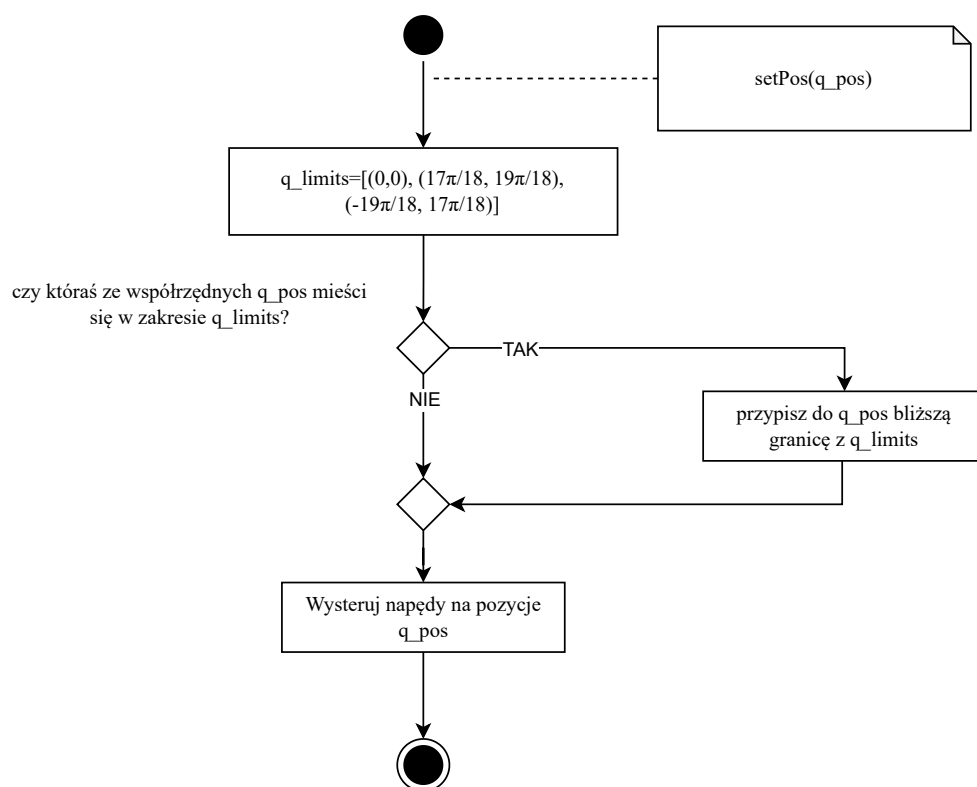
Funkcją realizującą tę warstwę jest getx2q(·), która oblicza nastawy w przegubach dla zadanego punktu końca nogi podanego w układzie nogi i ustawia je w przegubach nogi. Schemat jej implementacji pokazano na rys. 3.17. Polega ona na obliczeniu nastaw poprzez zastosowanie algorytmu do obliczania kinematyki odwrotnej Newtona-Raphsona [Ypm95]. Przyjęto tolerancję błędu $\epsilon = 0,01$ i maksy-

Rysunek 3.17: Metoda `getx2q(·)`

malną liczbę iteracji $N = 30$. Algorytm za wektor inicjujący przyjmuje aktualne nastawy przegubów w nodze.

3.3.5 Warstwa sterowania na poziomie wykonawczym

Ta warstwa została zrealizowana poprzez zaimplementowanie funkcji `setPos(·)`. Jej schemat przedstawiono na rys. 3.18. Jako argument przyjmuje ona wektor zadanych pozycji w przegubach. Na początku ustawia zmienną lokalną z przedziałami zabronionymi `q_limits`. Jeśli którakolwiek ze współrzędnych `q_pos` należy do odpowiadającego jej przedziału, wartość współrzędnej `q_pos` zastępowana jest wartością bliżej granicy przedziału określonego w `q_limits`. Na koniec pozycje `q_pos` są wysyłane jako sygnał sterujący do napędów w przegubach.

Rysunek 3.18: Metoda `setPos(·)`

Rozdział 4

Estetyka ruchu

Już w starożytnej Grecji refleksja nad pięknem miała charakter nie tylko artystyczny, ale i ścisły. Pitagorejczycy, badając zależności w muzyce, odkryli, że konsonanse (interwały brzmiące zgodnie) powstają, gdy długości wydających dźwięki strun pozostają w prostych stosunkach liczbowych, takich jak oktawa (1 : 2) czy kwinta (2 : 3). Dysonanse natomiast, jak półton (15 : 16), charakteryzowały się bardziej złożonymi proporcjami. Na tej podstawie szkoła pitagorejska sformułowała tezę, iż piękno stanowi harmonię matematyczną, będącą fundamentalną zasadą wszechświata [Arya].

Platon rozwijał tę myśl, umieszczając piękno w świecie idei jako obiektywną, wieczną i niezmienną formę [Pla]. Piękno materialne było dla niego jedynie niedoskonałym odbiciem tej doskonałej idei. Arystoteles, w przeciwieństwie do Platona, lokował źródło piękna w samych przedmiotach [Aryc]. Uważał, że wynika ono z trzech cech: uporządkowania (*gr. taxis*) elementów względem siebie, właściwej proporcji (*gr. symmetria*) pomiędzy częściami a całością oraz ograniczonej wielkości (*gr. to hōrismenon*), która pozwala objąć przedmiot jednym spojrzeniem.

Ideę matematycznego opisu piękna wskrzeszono i intensywnie rozwijano w epoce renesansu, gdy artyści i uczeni dążyli do sformułowania ścisłych, opartych na matematyce reguł rządzących architekturą, malarstwem i rzeźbą. W tym okresie włoski matematyk Luca Pacioli opublikował traktat „O boskiej proporcji” (łac. *De divina proportione*) [Pac09], w którym szczegółowo opisał właściwości złotej liczby, definiowanej jako

$$\Phi = \frac{1 + \sqrt{5}}{2} \approx 1,618. \quad (4.1)$$

Należy podkreślić, że Pacioli nie był odkrywcą tej proporcji, znanej już w starożytności, lecz jej najważniejszym renesansowym propagatorem. Traktat, ilustrowany rysunkami Leonarda da Vinci, przyczynił się do ugruntowania poglądu o wyjątkowym estetycznym statusie złotego podziału. Jednym z najsłynniejszych, choć niekiedy przecenianych, przykładów jego domniemanego zastosowania jest „Człowiek witruwiański” Leonarda da Vinci [Vin90], gdzie poszukiwano idealnych proporcji ciała ludzkiego.

Na przełomie baroku i klasycyzmu Alexander Gottlieb Baumgarten, powszechnie uznawany za twórcę estetyki jako samodzielnej dyscypliny filozoficznej, w dziele „Aesthetica” z 1750 roku [Bau50] zaproponował jej nazwę oraz dziedzinę. Według Baumgartena zadaniem nowej nauki było badanie poznania zmysłowego, którego

najwyższą i najdoskonalszą formą jest właśnie piękno. Utożsamiał je z doskonałością (*łac. perfectio*) samego zjawiska, a cel estetyki ujął w słynnej formule *perfectio cognitionis sensitivae* – doskonałość poznania zmysłowego, przeciwstawiając je poznaniu intelektualnemu.

Prawie czterdzieści lat później Immanuel Kant w fundamentalnej pracy z 1790 roku „Krytyka władzy sądzenia” (*niem. Kritik der Urteilkraft*) [Kan90] przedstawił przełomową koncepcję piękna, ujmując je poprzez sąd smaku. Sąd ten, w odróżnieniu od sądów logicznych, ma charakter subiektywny, choć rości sobie prawo do powszechnej ważności. Kant scharakteryzował go w czterech momentach:

1. **Bezinteresowność.** Zadowolenie towarzyszące postrzeganiu piękna jest wolne od jakichkolwiek interesów praktycznych; nie wynika z pożądania, użyteczności czy celowości poznawczej.
2. **Powszechność.** Uznając coś za piękne, nie tylko wyrażamy osobistą preferencję, ale oczekujemy, że wszyscy powinni podzielić nasz osąd, mimo że nie da się go udowodnić na drodze logicznej.
3. **Celowość bez celu.** Forma pięknego przedmiotu sprawia wrażenie celowej i harmonijnej, jak gdyby została zaprojektowana z myślą o określonym celu, jednak nie jesteśmy w stanie przypisać jej żadnego konkretnego, użytkowego znaczenia.
4. **Konieczność.** Piękno wywołuje w nas przekonanie o konieczności powszechnej zgody co do niego, opartej na wspólnym dla wszystkich „zmyśle wspólnym” (*łac. sensus communis*).

W pierwszej połowie XX wieku amerykański matematyk George D. Birkhoff w książce „Aesthetic Measure” [Bir33] podjął się próby ilościowego ujęcia piękna. Zaproponował w niej, by miarę estetyczną M dowolnego obiektu wyrazić jako stosunek porządku O do złożoności C :

$$M = \frac{O}{C}. \quad (4.2)$$

Złożoność C jest miarą wysiłku percepcyjnego, który odbiorca musi włożyć w ogarnięcie dzieła; im więcej elementów, relacji i różnic, tym złożoność wyższa. Porządek O to z kolei miara pozytywnych cech estetycznych, które ułatwiają percepcję i wywołują poczucie harmonii. Birkhoff wskazał, że składa się na niego suma czynników takich jak symetria S , równowaga wertykalna V , rytm lub powtarzalność R i inne:

$$O = S + V + R + \dots \quad (4.3)$$

Chociaż Birkhoff podał konkretne równania dla wybranych klas obiektów (wielokąty, wazy, ornamenty, muzyka), to jego model, pomimo matematycznej formy, w dużej mierze opiera się na subiektywnym doborze i ocenie składowych porządku oraz złożoności, co stanowiło główny przedmiot krytyki jego teorii.

Emocje i piękno można wyrazić również poprzez ruch. Aktor na deskach teatru nie zachwyca nas tylko i wyłącznie swoją mową, strojem czy dekoracją wokół, ale

również gestykulacją. Sekwencja ruchów często nie jest tylko bezcelowym wymachiwaniem kończynami a środkiem przekazu. Dla przykładu, w 1942 roku w książce „O technice aktora” (*ang. To the Actor: On the Technique of Acting*) [Che53] Michaił Czechow opisał technikę, którą dziś znamy pod jego nazwiskiem. Celem jej jest przeniesienie stanu wewnętrznego postaci na ruchy zewnętrzne. Aktor wyobraża sobie, jak ciało zmieniliby się pod wpływem emocji czy cech charakteru. I tak postać tchórzliwa może garbić się i wyobrażać, że jej ciało jest z gumy.

Formą sztuki, która wybrała ruch jako główny środek przekazu, jest taniec. Rozwinął się on szczególnie we Włoszech w epoce renesansu. Za pierwszego mistrza a zarazem teoretyka tej dziedziny uznaje się Domenico da Piacenza. Jako jeden z pierwszych próbował opisać taniec od strony teoretycznej i tak w swoim traktacie „O sztuce tańczenia i prowadzenia tańców” (*łac. De arte saltandi et choreas ducendi*) [Pia50] podał trzy fundamentalne zasady tańca:

1. **Misura (miara).** Właściwe trzymanie się rytmu muzyki.
2. **Memoria (pamięć).** Zdolność zapamiętania i odtworzenia właściwej sekwencji kroków.
3. **Aiere (powietrze, gracia).** Wdzięk, lekkość, pozorny brak wysiłku wkładanego w ruch.

Zaskakujące jest, jak dobrze te założenia wpisują się w sterowanie robotem, który w ściśle określonych chwilach wykonuje zapisane w wewnętrznej pamięci ruchy. Czyżby wystarczyło zaopatrzyć go w wystarczająco duży napęd, aby stał się renesansowym tancerzem uwodzącym swą gracją królewskie dwory? W rozdziale opiszemy sposób matematycznego ujęcia piękna ruchu.

4.1 Ocena estetyki chodu

Aby móc ocenić estetykę chodu powinniśmy wyszczególnić elementy składowe, na które mamy wpływ podczas projektowania chodu, które same w sobie mogą podlegać już ocenie, oraz dane wyjściowe, które są od nich zależne. Wśród tych pierwszych możemy wyszczególnić:

współczynnik wypełnienia chodu β – stosunek czasu trwania podparcia do czasu pełnego cyklu chodu,

różnicę faz odnóży φ – stosunek różnicy czasów rozpoczęcia kroku przez dwie nogi do czasu pełnego cyklu chodu,

długość kroku l_{step} – odległość między punktami podniesienia i opuszczenia nogi,

prędkość chodu v_{rob} – prędkość poruszania się środka ciężkości korpusu,

wysokość kroku h_{step} – długość pomiędzy najwyższym punktem trajektorii nogi a jej rzutem na płaszczyznę podłoża,

rodzinę funkcji planowania trajektorii $f_{\text{step}}(\cdot)$ – rodzina funkcji wykorzystywana do wyznaczenia trajektorii między punktami podniesienia i opuszczenia nogi.

Wśród zależnych od nich danych wyjściowych można wymienić:

pozycję robota w przestrzeni $x(t) : \mathbb{R} \rightarrow \mathbb{R}^3$ – pozycja robota w układzie świata,

sterowanie silników $u(t) : \mathbb{R} \rightarrow \mathbb{R}^{18}$ – wartości sterowania dla silników w czasie,

straty energii ΔE – różnica między energią dostarczoną do robota a energią potrzebną do przeniesienia środka ciężkości korpusu po trajektorii zadanej,

margines stabilności $SM(t) : \mathbb{R} \rightarrow \mathbb{R}$ – najmniejsza odległość rzutu środka ciężkości od krawędzi wielokąta podparcia w danej chwili.

Wszystkie wymienione powyżej wielkości powinny mieć wpływ na ocenę końcową estetyki chodu. Jeśli którakolwiek z nich okaże się nie mieć odzwierciedlenia w estetyce będzie to oznaczało, że zostały nieprawidłowo wybrane numeryczne wskaźniki oceny, które pomijają część informacji.

4.2 Wskaźniki estetyki

Wskaźnik estetyki powinien jak najwierniej ocenić ściśle, który ruch jest najpiękniejszy. Powinien być to funkcjonal, który przyjmie za argument trajektorię pojedynczej nogi, całego korpusu robota, sterowania lub innego parametru zmiennego w czasie i zwróci liczbę rzeczywistą.

Poniżej przedstawimy funkcjonały szpetoty Γ , które należy minimalizować w celu osiągnięcia ruchu charakteryzującego się jak największą gracją. Odwrotność wartości Γ będziemy nazywać gracjanami $G = \Gamma^{-1}$, które będą stanowić właściwe wskaźniki oceny estetyki.

4.2.1 Zgrabność

Ruch estetyczny powinien być możliwie gładki. W [FH85] stwierdzono, że aby osiągnąć możliwie gładką trajektorię w przestrzeni zadaniowej (w ich przypadku ręki) należy minimalizować zryw (*ang. jerk*). Autorzy podają funkcję celu jako kwadrat wartości zrywu efektora, scałkowany w czasie trwania całego ruchu ($0, t_f$). Odnosząc to do będącego przedmiotem pracy przypadku heksapoda i nogi, funkcja ta wyglądałaby następująco

$$\Gamma_Z = \sum_{i=1}^6 \int_0^{t_f} \left(\left(\frac{d^3 x_i}{dt^3} \right)^2 + \left(\frac{d^3 y_i}{dt^3} \right)^2 + \left(\frac{d^3 z_i}{dt^3} \right)^2 \right) dt, \quad (4.4)$$

gdzie i to numer nogi, zaś x_i, y_i, z_i to współrzędne jej końca w chwili t . Za pomocą Γ_Z definiujemy *gracjan zgrabny*

$$G_Z = \Gamma_Z^{-1}. \quad (4.5)$$

4.2.2 Odporność na błędy

Ewolucja kształtowała ruch w taki sposób, aby był możliwie efektywny, a więc też odporny na błędy. W [HW98] zaprezentowano teorię, w jaki sposób ludzki organizm wybiera optymalną trajektorię osiągnięcia celu. Wychodząc z założenia, że „sygnały w mózgu są zaburzone przez szum, którego wariancja rośnie wraz ze wzrostem sygnału sterującego”, proponowane jest wyznaczanie trajektorii takiej, która minimalizuje wariancję pozycji końcowej

$$\Gamma_B = \sum_{i=1}^6 \text{Var}(x_i), \quad (4.6)$$

gdzie x_i to pozycja i -tej nogi w chwili rozpoczęcia fazy retrakcji. Γ_B posłuży nam do zdefiniowania *gracjanu błędnego*

$$G_B = \Gamma_B^{-1}. \quad (4.7)$$

4.2.3 Zużycie energii

Ruch wielokrotnie powtarzany jest przez ciało tak modyfikowany, aby osiągnąć jak najmniejszy wydatek energetyczny. W [FKMP21] pokazano, że minimalizacja energii wykorzystanej do ruchu prowadzi do pojawienia się naturalnych ruchów zwierząt. W analizowanym tam przypadku – czworonoga – ruchy były zbliżone do ruchów owiec i koni. Zatem powinniśmy minimalizować funkcję

$$\Gamma_E = E_C - E_W, \quad (4.8)$$

gdzie E_C to energia całkowita, dostarczona do układu napędowego, a E_W – energia wykorzystana, wynikająca z przemieszczenia środka ciężkości robota. W ten sposób definiujemy *gracjan energetyczny*

$$G_E = \Gamma_E^{-1}. \quad (4.9)$$

4.2.4 Finezja

Do wykonania ruchu potrzebna jest siła, która umożliwi przeniesienie ciała w zadany sposób. Jednak mimo iż sztangista potrafi podnieść dużo więcej masy niż danseur noble (*franc. szlachetny tancerz*) to jednak ruchy tego drugiego nas urzekają, a pierwsze uznajemy za nieposiadające finezji. Dlatego naszym zdaniem, w celu niejako zmuszenia do takich ruchów, powinniśmy minimalizować funkcję

$$\Gamma_F = \int_0^{t_f} \tau_R - |\tau_u(t)| dt, \quad (4.10)$$

gdzie τ_R to wektor maksymalnych momentów sił w przegubach robota, $\tau_u(t)$ – funkcja wektorowa momentów sił sterowania, a $|\cdot|$ – operator wartości bezwzględnej. I tak definiujemy *gracjan finezyjny*

$$G_F = \Gamma_F^{-1}. \quad (4.11)$$

4.2.5 Subtelność

Ruch to również składowe, których nie widać, które są na tyle naturalne, że o nich nie myślimy. Jedną z kluczowych rzeczy jest nie przewrócić się, nie stracić stabilności. Potykające się o własne nogi szczenie, może być dla nas urocze i zabawne, jednak sam sposób poruszania się – nieestetyczny. Stabilność w czasie chodu możemy opisywać poprzez margines stabilności SM, który jest najmniejszą odległością rzutu środka ciężkości S_G od dowolnej krawędzi wielokąta podparcia. Wartość tej odległości będzie maksymalna, gdy wspomniany rzut będzie środkiem S_O maksymalnego okręgu wpisanego (okręgu o największym promieniu mieszczącym się wewnątrz wielokąta). Z tego względu powinniśmy minimalizować funkcję

$$\Gamma_S = \int_0^{t_f} |S_O(t) - S_G(t)| dt, \quad (4.12)$$

gdzie $S_O(t)$ i $S_G(t)$ są funkcjami zależności odpowiednio środka okręgu maksymalnego i rzutu środka ciężkości od czasu. W konsekwencji definiujemy *gracjan subtelny*

$$G_S = \Gamma_S^{-1}. \quad (4.13)$$

4.2.6 Piękno ogólne

Aby oceniać estetykę chodu całościowo powinniśmy posługiwać się w tym celu wskaźnikiem ogólnym, który będzie zawierał w sobie informacje ze wszystkich poprzednich. Dlatego definiujemy *gracjan powabny*

$$G_P = \frac{\langle k, G \rangle}{\langle k, k \rangle^{\frac{1}{2}}}, \quad (4.14)$$

gdzie G to wektor złożony z kolejnych gracjanów $(G_Z, G_B, \dots)^T$, natomiast k – wektor odpowiadających im nieujemnych wag $k = (k_Z, k_B, \dots)^T$, $\langle \cdot, \cdot \rangle$ oznacza iloczyn skalarny. Wektor k nazywać będziemy wektorem subiektywności.

Rozdział 5

Badania

Załączków prowadzenia badań możemy doszukiwać się już w epoce kamienia łupanego. Ówczesny człowiek dokonywał setek doświadczeń i obserwacji zdobywając potrzebną do przetrwania wiedzę. Ileż czasu i istnień musiało zająć zanim opanował obróbkę kamieni, tworząc pierwsze narzędzia, ujarzmił ogień, przyrządzając pierwsze gorące potrawy, poznał moc pracy w grupie, polując skoordynowanie na mamuty. W ten sposób powstały podwaliny takich dyscyplin naukowych jak inżynieria materiałowa, gastronomia molekularna czy zarządzanie zasobami ludzkimi. Gromadzoną wówczas wiedzę przekazywano ustnie i w postaci malunków naskalnych, z których jedne z najsłynniejszych możemy podziwiać w Lascaux [Las]. Po wielu tysiącach lat badań tych bardziej udolnych i tych nieco mniej, język stęków i pojękiwań zamieniliśmy na wzory matematyczne, a ścianę jaskini na cyfrowy papier. Jednak to, co pozostało niezmiennie to sposób poszukiwań. Rzekomo sam Albert Einstein miał powiedzieć [goo]

„Gdybyśmy wiedzieli, co robimy, to nie nazywałoby się to badaniem, prawda?”

W rozdziale opiszemy badania jakie przeprowadziliśmy w poszukiwaniu najbardziej estetycznego ruchu robota.

5.1 Sformułowanie zadania

Założmy niezależność wymienionych wcześniej elementów składowych chodu (rozdz. 4.1) i przypiszmy każdemu odpowiadający mu wersor,

- dla β : $e_\beta = (1, 0, 0, \dots, 0) \in \mathbb{R}^n$,
- dla φ : $e_\varphi = (0, 1, 0, \dots, 0) \in \mathbb{R}^n$ i.t.d.,

gdzie n określa liczbę branych pod uwagę składowych. Wersory te będą rozpinąć n wymiarową przestrzeń $\mathcal{G}^n$ będącą przestrzenią parametrów chodu. Każdy z parametrów ma swój zakres wartości, więc będziemy poruszać się po pewnej dozwolonej podprzestrzeni przestrzeni $\mathcal{G}^n$.

Chód robota możemy potraktować jako transformację Υ przestrzeni parametrów chodu $\mathcal{G}^n$ w przestrzeń funkcyjną chodu $\mathfrak{G}$

$$\mathcal{G}^n \xrightarrow{\Upsilon} \mathfrak{G}. \quad (5.1)$$

Zdefiniujmy teraz funkcję $\lambda : \mathfrak{G} \times \mathfrak{G} \rightarrow \mathbb{R}$ za pomocą gracjanu powabnego w postaci

$$\lambda(g_1, g_2) = |G_P(g_2) - G_P(g_1)|, \quad (5.2)$$

gdzie $g_1, g_2 \in \mathfrak{G}$. Zauważmy, że funkcja ta dla każdego $g_1, g_2 \in \mathfrak{G}$ spełnia następujące własności:

1. zwraca wartości nieujemne $\lambda(g_1, g_2) \geq 0$,
2. jest symetryczna $\lambda(g_1, g_2) = \lambda(g_2, g_1)$,
3. zachowuje nierówność trójkąta $\lambda(g_1, g_3) \leq \lambda(g_1, g_2) + \lambda(g_2, g_3)$.

Równocześnie nie spełnia ona aksjomatu identyczności nierozróżnialnych

$$\lambda(g_1, g_2) = 0 \iff g_1 = g_2. \quad (5.3)$$

Prowadzi to do konkluzji, że λ może być wykorzystana jako pseudometryka [Enc]. Parę $(\mathfrak{G}, \lambda)$ – pseudometryczną przestrzeń funkcyjną będziemy nazywać *przestrzenią gracji*.

Wykorzystując nowopoznane struktury, zadanie:

Poprawić estetykę chodu sześcionożnego robota krocącego poprzez modelowanie i optymalizację parametrów ruchu, w celu osiągnięcia bardziej płynnego i naturalnego sposobu poruszania się.

możemy sformułować w następujący sposób:

Przeszukać podprzestrzeń dozwoloną przestrzeni $\mathcal{G}^n$ w celu znalezienia punktu, którego obraz w przestrzeni gracji $\mathfrak{G}$ będzie znajdował się w największej odległości λ od zerowego gracjanu powabnego $G_P(g) = 0$.

5.2 Własności ilościowego opisu piękna

Zdefiniowana przestrzeń gracji $\mathfrak{G}$ posiada wiele własności, które utwierdzają nas w przekonaniu, że została opisana za pomocą właściwych struktur matematycznych. A mianowicie:

1. **Odległość estetyczna.** Dwa różne punkty przestrzeni mogą mieć zerową odległość między sobą. Zgadza się to z przeświadczeniem, że dwa różne obiekty (w tym przypadku sposoby poruszania się) mogą być tak samo estetyczne, a więc odległość między nimi w wymiarze piękna będzie zerowa.
2. **Wektor subiektywności.** Wektor wag k może przyjąć dowolne, nieujemne wagi. Stanowi on element subiektywności oceny. Każdy odbiorca może inaczej odbierać piękno, kierując się tymi samymi kryteriami (gracjanami), poprzez wybranie innego wektora wag.

W szczególnym przypadku wektor ten może posiadać wszystkie składowe zerowe – odpowiadałoby to sytuacji, kiedy odbiorca nie chce poznać piękna

obiekty lub składowe o wartości nieskończonej – co świadczyłoby o porzuceniu kantowskiego momentu o powszechności swojej oceny.

Przytoczony w rozdziale 4 moment kantowskiego sądu smaku mówiący o konieczności powszechnej zgody co do piękna, opartej na „zmyśle wspólnym” niejako sugeruje, że wektor subiektywności w populacji podlegać powinien rozkładowi normalnemu, a szczególne jego przypadki o bardzo dużych lub bardzo małych wagach powinny być rzadkością.

3. **Szpetota totalna.** Funkcjonały szpetoty zwracają wartości skończone, ponieważ definiowane są poprzez całkowanie funkcji ograniczonych na skończonym przedziale, różnice skończonych energii, wariancję zmiennej o skończonych wartościach.
4. **Niezbywalne piękno.** Gracjan zerowy $G_P(g) = 0$ jest nieosiągalny. Wynika to wprost z jego definicji (równanie (4.14)). Wymagałby on
 - (a) nieskończonych wartości wszystkich branych pod uwagę funkcjonałów szpetoty
 - (b) albo zerowego wektora wag k ,
 - (c) tudzież odpowiedniej kombinacji dwóch powyższych.

Zatem każdy rzeczywisty ruch ma w sobie przynajmniej minimalne ilości piękna. Analogicznie nie można uzyskać gracjanu nieskończonego $G_P(g) = \infty$. Niektórzy zakładają, że nieskończone piękno jest cechą Boga, więc wartość $G_P(g) = \infty$ nazwijmy gracjanem boskim G_Δ .

5.3 Sposób poszukiwań

Przeszukiwanie przestrzeni wysokowymiarowych jest problemem skomplikowanym, w którym należy zmierzyć się z wieloma przeszkodami zbiorczo nazwanymi przez R.E. Bellmana „przekleństwem wielowymiarowości” [Bel61]. Możemy wśród nich wskazać takie elementy jak:

1. **Zasobożerność.** Operacje na danych wysokowymiarowych są wyjątkowo kosztowne pod względem zasobów obliczeniowych i pamięciowych.
2. **Rozrzedzenie danych.** Przy stałej liczbie punktów pomiarowych, wzrost wymiarowości prowadzi do wykładniczego spadku gęstości tych punktów w przestrzeni. Oznacza to, że dane stają się ekstremalnie rozrzedzone, co utrudnia uogólnianie i modelowanie statystyczne.
3. **Cechy nieistotne.** Istnieje wysokie ryzyko, że zdefiniowany zbiór cech zawiera zmienne nieistotne lub zaszumione. Elementy te, zamiast wnosić wartość, niepotrzebnie komplikują problem i mogą obniżać jakość wyników.
4. **Kontrast odległości.** W miarę wzrostu wymiarowości, metryki odległości pomiędzy dowolnymi dwoma punktami w przestrzeni mają tendencję do stawania się liczbowo bardzo podobnymi. Zjawisko to prowadzi do utraty znaczenia

odległości jako wskaźnika podobieństwa, co negatywnie wpływa na algorytmy bazujące na bliskości.

Biorąc pod uwagę powyższe fakty, do przeszukiwania przestrzeni $\mathcal{G}$ (poprawienia parametrów chodu pod względem estetyki) wykorzystano trzy algorytmy:

- algorytm genetyczny (GA) [Ho192], bazujący na krzyżowaniu aktualnego zbioru rozwiązań,
- algorytm symulowanego wyżarzania (SA) [KGV83], przypominający proces stygnięcia metalu,
- algorytm przeszukiwania tabu (TS) [GL97], polegający na zmiennym sąsiedztwie punktu.

5.4 Poszukiwanie piękna

Badania przeprowadzono na komputerze o parametrach:

- CPU: Intel Core i5-10300H 2,50GHz,
- RAM: 16GB,
- GPU: NVIDIA GeForce GTX 1650 Ti.

Do przeszukiwania wybrano przestrzeń $\mathcal{G}^6$ rozpiętą na $n = 6$ wersorach odpowiadających wszystkim wyszczególnionym w pracy parametrom chodu oraz jej podprzestrzeń dozwoloną $\mathcal{G}_{\mathcal{D}}^6$ definiowaną przez zbiory dla odpowiednich parametrów:

- współczynnik wypełnienia chodu $\beta \in \{\frac{1}{2}, \frac{2}{3}, \frac{5}{6}\}$, które osiągane są poprzez odpowiednie rodzaje chodu: trójpodporowy (3|3), czwórpodporowy (2|4), pięciopodporowy (1|5),
- funkcja planowania kroku $f_{\text{step}}(\cdot) \in \{f_{\text{tria}}(\cdot), f_{\text{trap}}(\cdot), f_{\text{parab}}(\cdot)\}$, gdzie $f_{\text{tria}}(\cdot)$ oznacza rodzinę funkcji trójkątnych, $f_{\text{trap}}(\cdot)$ – trapezowych, $f_{\text{parab}}(\cdot)$ – kwadratowych,
- prędkość korpusu $v_{\text{rob}} \in \{0,1k \mid k \in \mathbb{Z} \wedge 0,5 \leq 0,1k \leq 3,0\}$,
- wysokość kroku $h_{\text{step}} \in \{0,1k \mid k \in \mathbb{Z} \wedge 0,5 \leq 0,1k \leq 2,0\}$,
- długość kroku $l_{\text{step}} \in \{0,1k \mid k \in \mathbb{Z} \wedge 0,5 \leq 0,1k \leq 2,0\}$,
- wysokość chodu $h_{\text{rob}} \in \{0,1k \mid k \in \mathbb{Z} \wedge 0,8 \leq 0,1k \leq 2,0\}$.

Poniżej przedstawimy i omówimy wpływ zmiany każdego z tych parametrów na wartości gracjanów oraz wyniki poszukiwań wymienionych w podrozdziale 5.3 algorytmów na przykładzie wektora parametrów chodu

$$\rho = (2/3; 1,0; 0,5; 1,8; f_{\text{parab}}(\cdot); 1,8)^T \in \mathcal{G}_{\mathcal{D}}^6. \quad (5.4)$$

oraz wektora subiektywności

$$k = (10^7, 10, 10^3, 10^4, 10^2)^T \quad (5.5)$$

Do obliczeń wskaźników wykorzystano pobrane z symulacji dyskretne ciągi danych:

- położenia robota w czasie w układzie świata,
- położenia końców nóg w czasie w układzie świata,
- momentów sił w przegubach odnóży w czasie,
- pozycji w przegubach odnóży w czasie,
- wektora siły działającej na środek ciężkości robota w czasie.

5.4.1 Wpływ parametrów na gracjany

Gracjan zgrabny

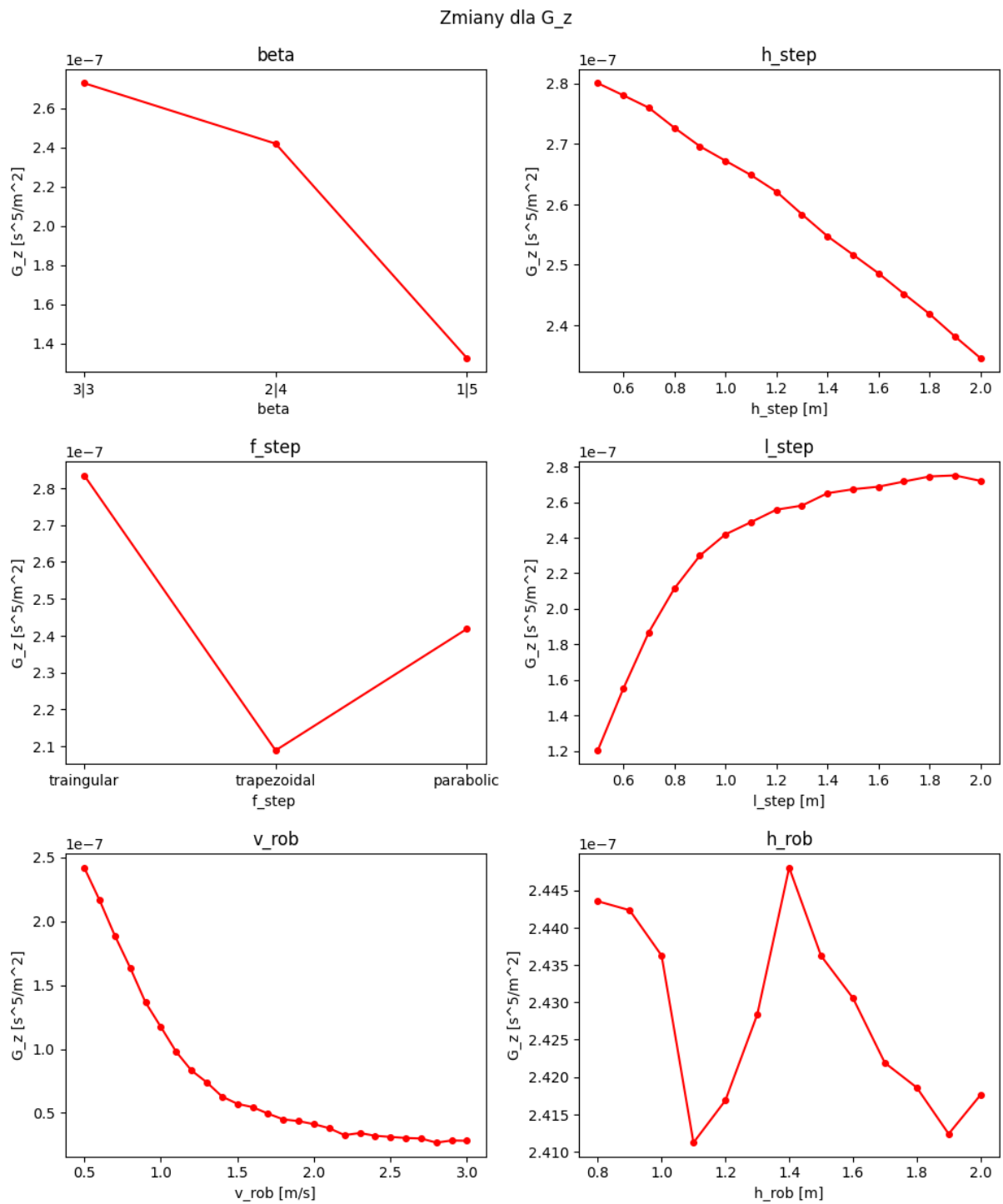
Dla ciągu próbek położenia końców nóg robota obliczono zgodnie ze wzorem (4.5) wartość gracjanu zgrabnego. Wykorzystano przy tym zaimplementowane funkcje `gradient(·)` oraz `trapezoid(·)` pakietu do obliczeń numerycznych NumPy [Num] realizujących odpowiednio różniczkowanie i całkowanie numeryczne. Wyniki zebrano na rys. 5.1. Można zauważyć, że wartość tego wskaźnika rośnie wraz ze wzrostem długości kroku, natomiast maleje dla zwiększania wypełnienia chodu, wysokości kroku czy prędkości korpusu. Zgadza się to z przypuszczeniami. Im dłuższy krok, tym trajektoria ruchu końca nogi wygładza się, w konsekwencji mniejszy jest jej zryw. Jeśli robot zmuszony jest wykonywać kroki szybciej, realizować kroki dłuższe czy po prostu przemieszczać się szybciej, to trudniej zachować mu zryw o niskiej wartości.

Co ciekawe, możemy zauważyć wyraźne maksimum dla wysokości korpusu dla $h_{\text{rob}} = 1,4$ m. Wartość ta wydaje się być optymalną pod względem minimalizacji zrywu. Najlepszą funkcją planowania kroku w tym zestawieniu okazała się funkcja trapezoidalna.

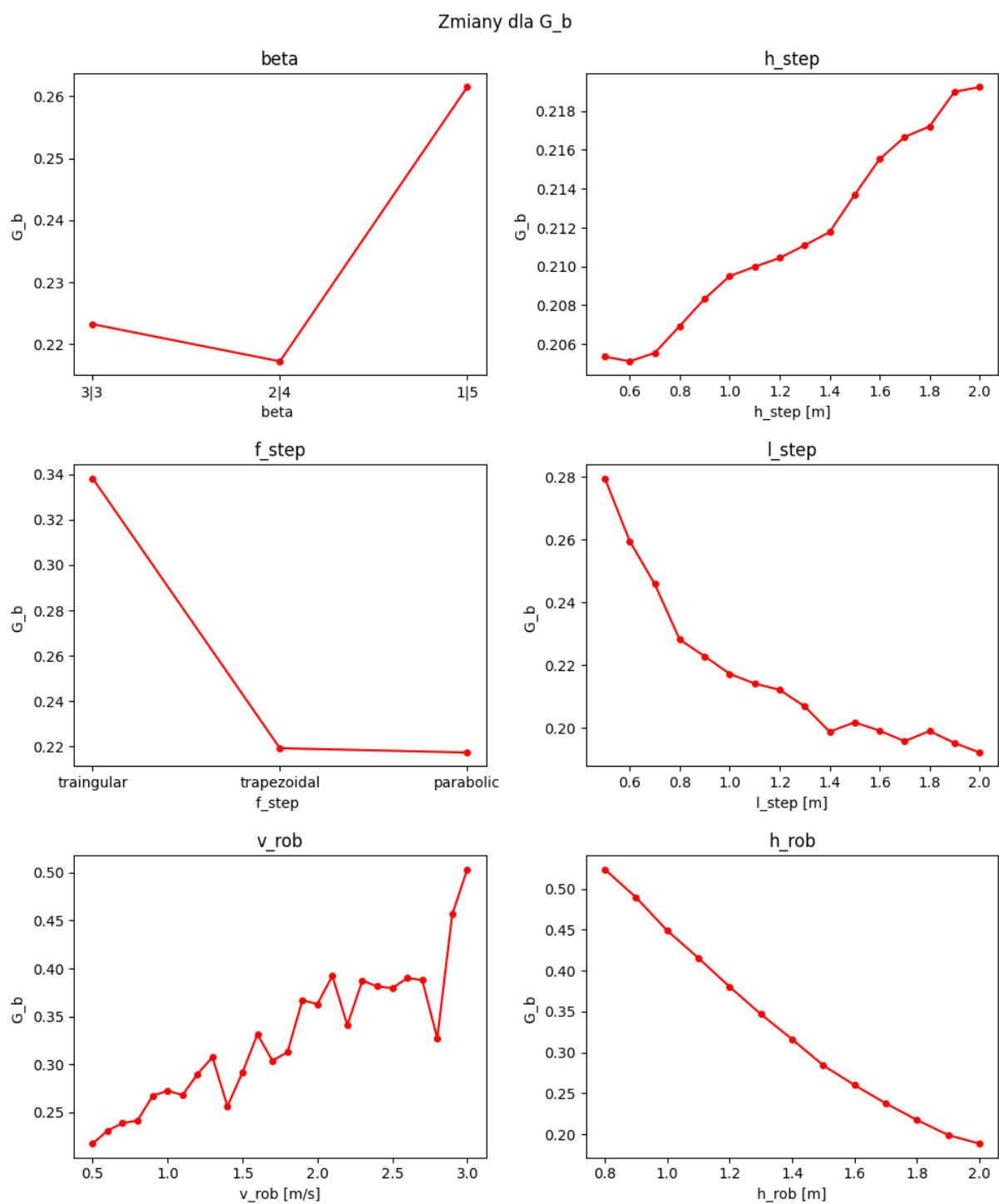
Gracjan błędny

Dla danych o położeniu końców nóg robota przyjęto, że kontakt z podłożem następuje, gdy wartość współrzędnej z spełni warunek $z < \epsilon$, gdzie przyjęto $\epsilon = 0,01$ m. Wariancję obliczano z serii pierwszych próbek spełniających wspomniany warunek wyrażonych w układzie robota $X_R Y_R Z_R$. Wartość gracjanu uzyskano zgodnie ze wzorem (4.7), co przedstawiono na rys. 5.2 w zależności od zmian parametrów chodu. Jego wartość zwiększa się przy wyższych krokach, ponieważ łatwiej jest trafić w założony punkt mając do pokonania dłuższą drogę. Natomiast zmniejsza się przy krokach dłuższych, ponieważ trajektoria ruchu nogi wypłaszcza się. Najlepszą funkcją planowania trajektorii kroku okazała się funkcja trójkątna, a najlepszym typem chodu – chód pięciopodporowy. Warto odnotować, że również w przyrodzie chód ten wykorzystywany jest przez zwierzęta do precyzyjnego przemieszczania się w nieznajomym terenie.

Co wymaga zauważenia, to to, że wartość gracjanu rośnie wraz ze wzrostem szybkości korpusu i maleje ze wzrostem wysokości korpusu nad płaszczyzną ruchu.



Rysunek 5.1: Wartości gracjanu zgrabnego w zależności od zmian parametrów chodu



Rysunek 5.2: Wartości gracjanu błędnego w zależności od zmian parametrów chodu

Gracjan energetyczny

Na podstawie próbek sił działających na środek ciężkości korpusu robota $F(t)$, pozycji robota $r(t)$, momentów sił $\tau(t)$ i pozycji w przegubach nóg $q(t)$ obliczono energię wykorzystaną jako pracę

$$E_W = W_W = \int_0^{t_f} F(t)r(t)dt \quad (5.6)$$

a całkowitą energię dostarczoną jako pracę

$$E_C = W_C = \int_0^{t_f} \tau(t)q(t)dt \quad (5.7)$$

Do obliczeń wykorzystano implementację całkowania numerycznego realizowanego poprzez funkcję `trapezoid(·)`. I tak z równania (4.9) wyznaczono wartość gracjanu energetycznego. Wartości wskaźnika w zależności od zmiany parametrów chodu zebrano na rys. 5.3. Rośnie ona wraz ze wzrostem długości kroku, prędkości chodu i wysokości korpusu nad płaszczyzną chodu. Można zauważyć wyraźne minimum dla wysokości kroku $h_{step} = 1,3$ m. Najlepszym typem chodu okazał się chód pięciopodporowy, a najlepszą funkcją do planowania trajektorii – funkcja trójkątna.

Gracjan finezyjny

W przeprowadzonych symulacjach wektor maksymalnych momentów sił w przegubach miał postać

$$\tau_R = (\tau_1, \tau_2, \dots, \tau_i)^T = \quad (5.8)$$

$$= (\tau_{1,1}, \tau_{1,2}, \tau_{1,3}, \tau_{2,1}, \dots, \tau_{i,3})^T = \quad (5.9)$$

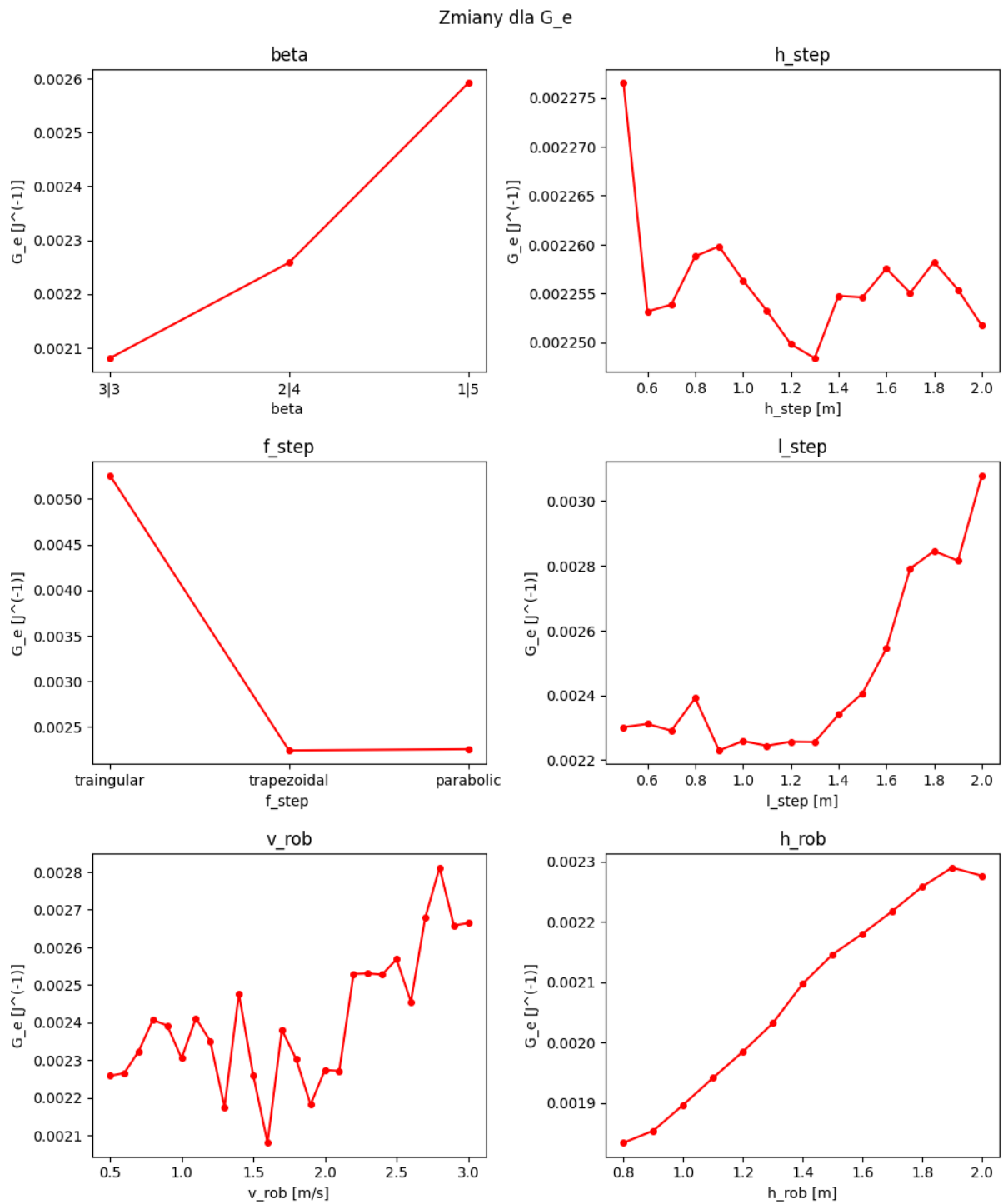
$$= (50, 40, 30, 50, 40, 30, 50, 40, 30, 50, 40, 30, 50, 40, 30, 50, 40, 30)^T, \quad (5.10)$$

gdzie τ_i jest wektorem maksymalnych momentów sił napędów w i -tej nodze i odpowiednio $\tau_{i,1}, \tau_{i,2}, \tau_{i,3}$ jego składowymi dla kolejnych przegubów $q_{i,1}, q_{i,2}, q_{i,3}$ (rys. 2.6). Podstawiając do równania (4.11) τ_R oraz próbki momentów sił działających w przegubach robota obliczono gracjan finezyjny. Wartości wskaźnika przy zmianie parametrów chodu przedstawiono na rys. 5.4.

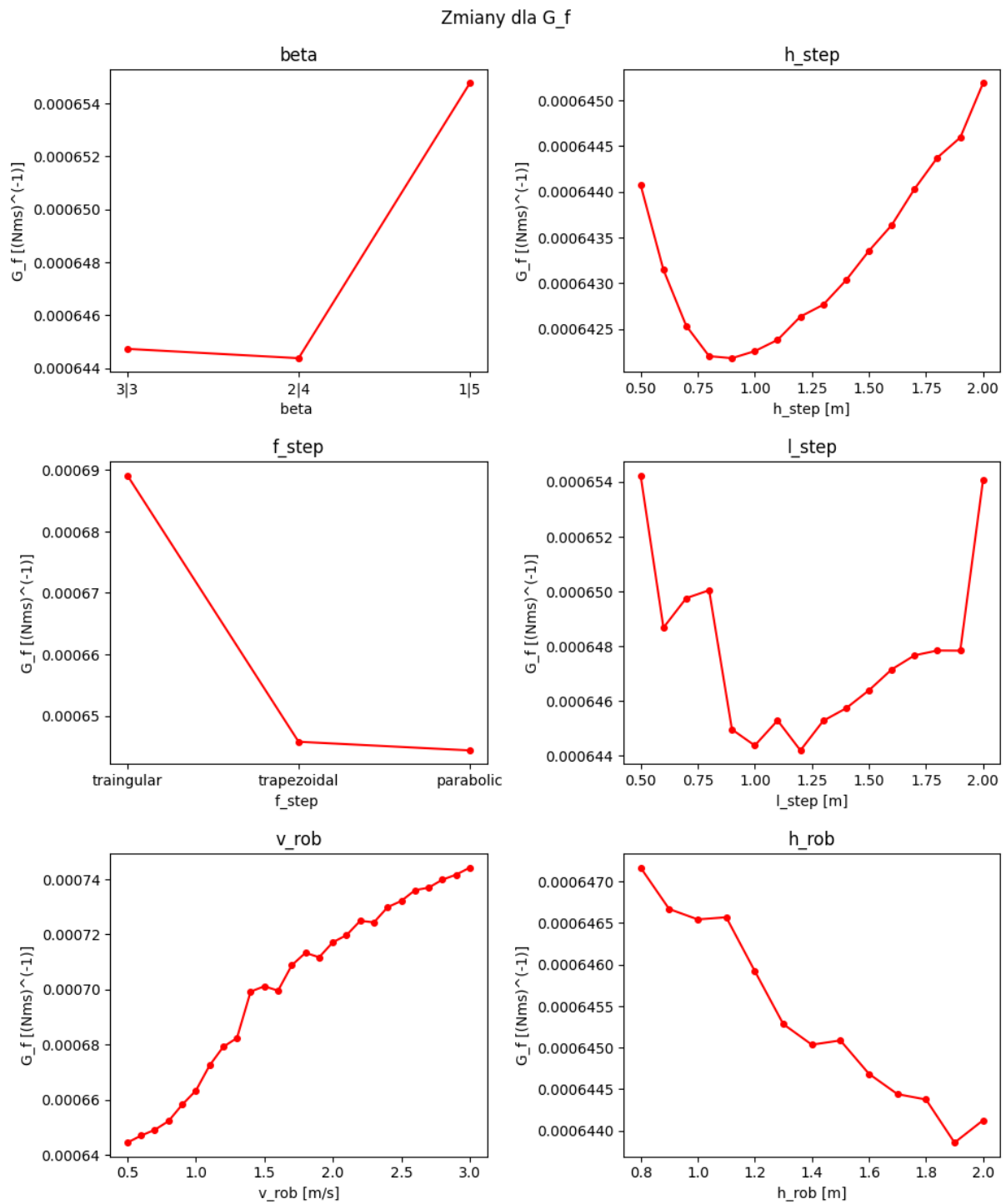
Można zauważyć wzrost wartości wyliczonego wskaźnika przy zwiększaniu prędkości robota, wysokości i długości kroku. Wynika to z potrzeby przyłożenia większej siły, aby nogę przestawić w krótszym czasie lub po dłuższej trajektorii. Obserwujemy również tendencję spadku zapotrzebowania na siłę przy coraz to niższych wysokościach utrzymywania korpusu, co również wydaje się być naturalne. Ruchem wykorzystującym najmniej dostępnej siły okazał się ruch czwórpodporowy oraz przemieszczanie nogi po trajektorii parabolicznej.

Gracjan subtelny

Dla próbek położenia końców nóg robota przyjęto punkt kontaktu z podłożem analogicznie jak przy wyznaczaniu gracjanu błędnego. Na tej podstawie wyznaczono



Rysunek 5.3: Wartości gracjanu energetycznego w zależności od zmian parametrów chodu



Rysunek 5.4: Wartości gracjanu finezyjnego w zależności od zmian parametrów chodu

wartość rzuty środka ciężkości S_G i środek okręgu maksymalnego wielokąta podparcia S_O , a następnie gracjanu subtelnego zgodnie z równaniem (4.13). Wartości wskaźnika dla zmiennych parametrów chodu zebrano na rys. 5.5.

Można zaobserwować wzrost wartości wskaźnika dla zwiększania wysokości kroku i prędkości chodu, natomiast spadek dla zwiększania długości kroku i wysokości korpusu nad płaszczyzną chodu. Najlepszym typem chodu okazał się chód trójpodporowy, a najlepszą funkcją planowania kroku – funkcja trójkątna.

Gracjan powabny

Na podstawie wyliczonych wcześniej gracjanów dla wybranego wektora k obliczono zgodnie ze wzorem (4.14) wartość gracjanu powabnego. Wartości wskaźnika dla zmiennych parametrów chodu przedstawiono na rys. 5.6

Można zauważyć, że kształtem wykresy są bardzo zbliżone do tych wykreślonych dla gracjanu subtelnego. Oznacza to, że w dużej mierze zdominował on wartość gracjanu powabnego, miał przypisaną zbyt dużą wagę.

5.4.2 Wyniki poszukiwań algorytmów

Algorytm genetyczny

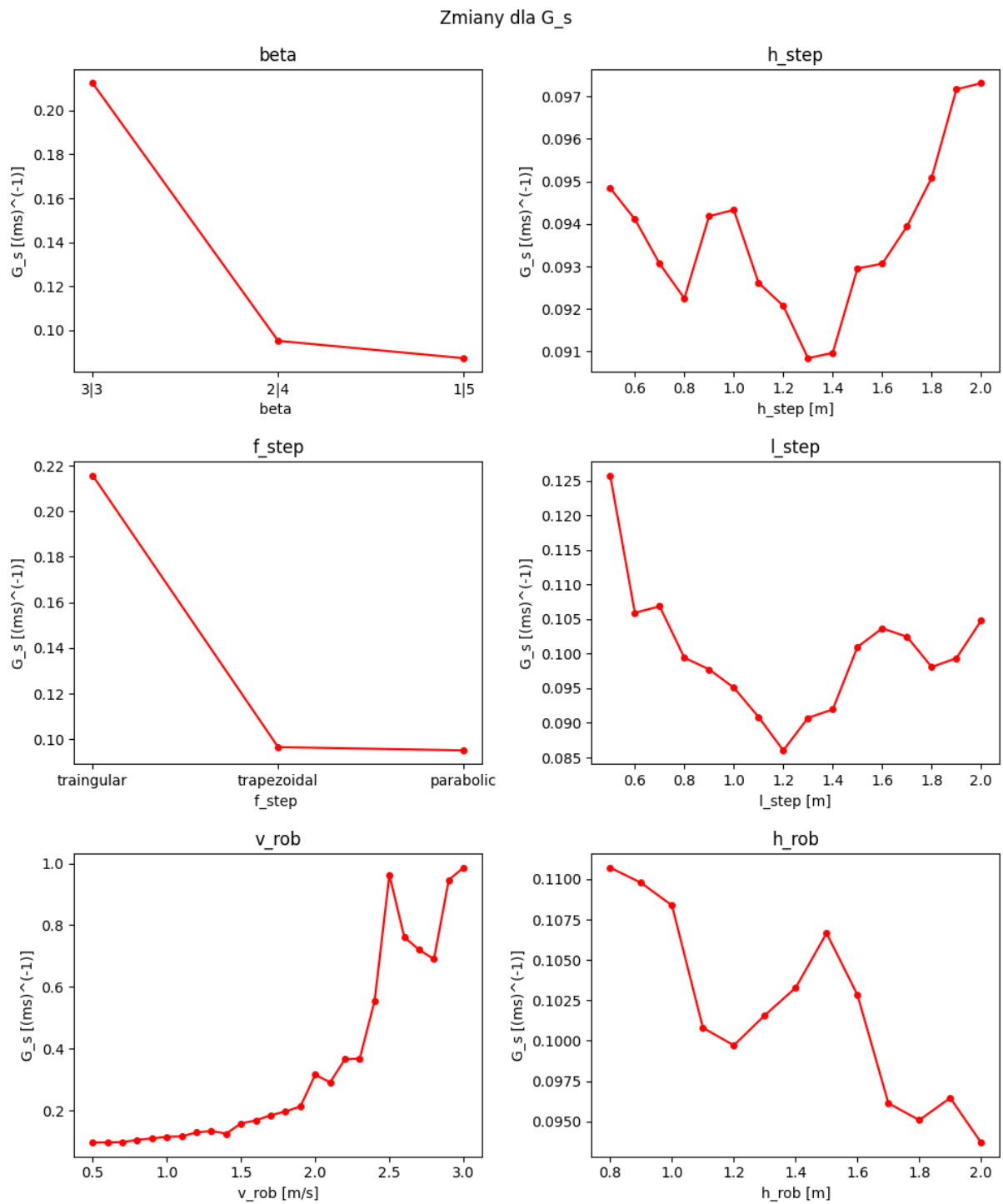
Zrealizowaną implementację algorytmu genetycznego [Hol92] można przedstawić w następujących krokach:

1. **Populacja początkowa.** Losowanych jest $N = 20$ punktów $p \in \mathcal{G}_D^6$ tworząc zbiór $\mathcal{N}$.
2. **Ocena osobników.** Dla każdego z punktów $p \in \mathcal{N}$ przeprowadzana jest symulacja i przypisywana ocena równa gracjanowi powabnemu G_p .
3. **Wybór najlepszych osobników.** Wybieranych jest 10 punktów o najwyższej ocenie tworzących zbiór $\mathcal{N}^\dagger$.
4. **Krzyżowanie.** Nowy punkt powstaje z dwóch wybranych losowo punktów $p_A, p_B \in \mathcal{N}^\dagger$ jako

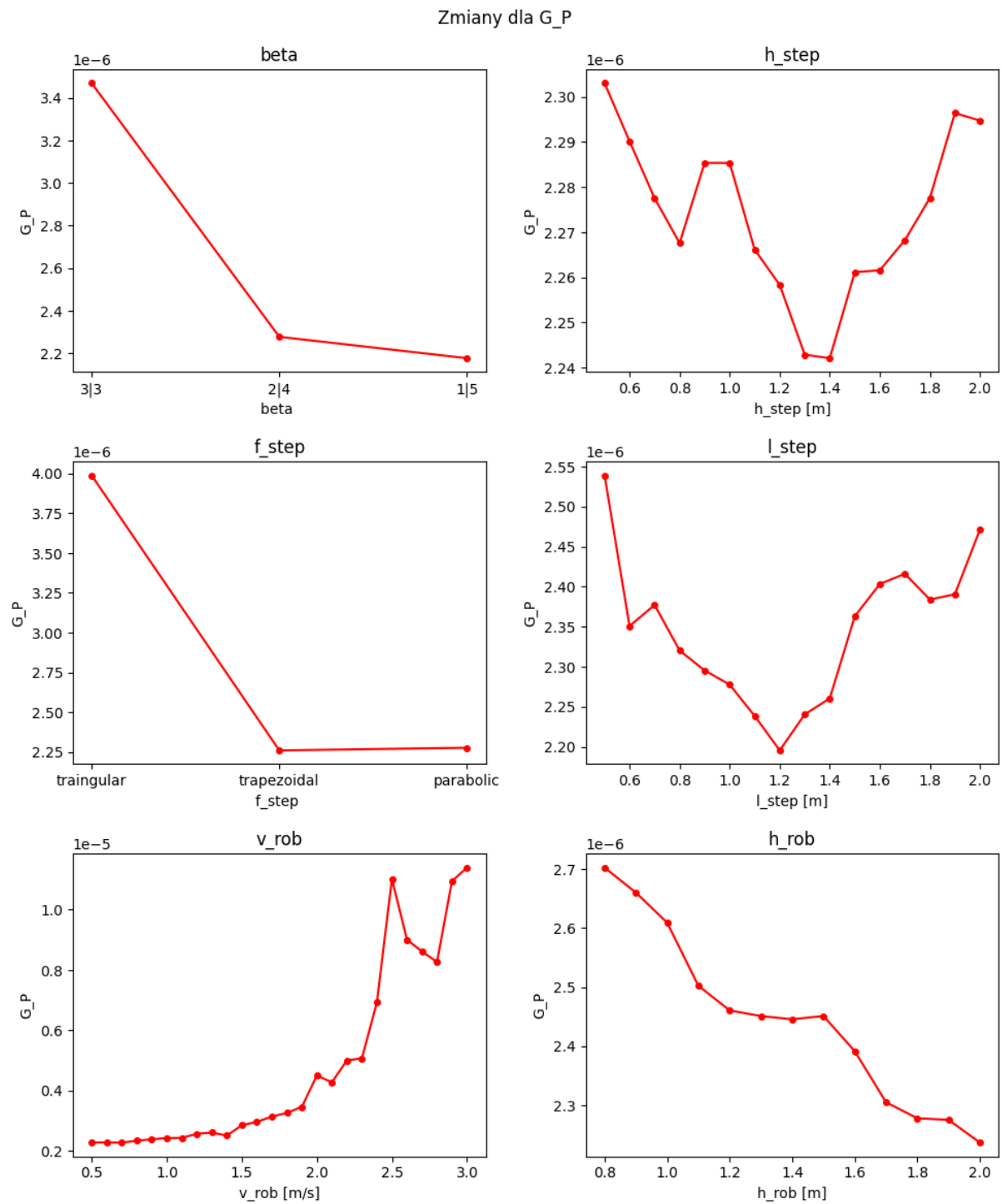
$$p_n = (p_{A,1}, p_{B,2}, p_{A,3}, \dots, p_{A,i}, p_{B,i+1}), \quad (5.11)$$

gdzie $p_{A,i}, p_{B,i}$ to kolejne współrzędne punktów p_A i p_B . W ten sposób tworzy się N punktów stanowiących zbiór $\mathcal{N}'$.

5. **Mutacja.** Mutacja polega na zamianie losowej współrzędnej punktu $p \in \mathcal{N}'$ na losową wartość ze zbioru wartości odpowiadającego jej parametru. Operacji tej poddawane jest 10% punktów $\mathcal{N}'$.
6. **Nowe pokolenie.** Zbiór $\mathcal{N}'$ staje się zbiorem $\mathcal{N}$.
7. **Wynik.** Zwracany jest najlepszy napotkany podczas przeprowadzania algorytmu punkt.



Rysunek 5.5: Wartości gracjanu subtelnego w zależności od zmian parametrów chodu



Rysunek 5.6: Wartości gracjanu powabnego w zależności od zmian parametrów chodu

Tabela 5.1: Wartości poszczególnych graczy dla punktu p_{GA}

$G_Z [10^{-8}]$	$G_B [1]$	$G_E [10^{-3}]$	$G_F [10^{-4}]$	$G_S [10^{-1}]$	$G_P [10^{-5}]$
2,15	1,10	1,61	7,15	8,84	1,08

Kroki od 2 do 6 powtarzane są M razy, gdzie M to liczba pokoleń. Poszukiwanie zrealizowano dla wartości $M = 15$. W wyniku otrzymano punkt

$$p_{GA} = (1/2; 0, 5; 3, 0; 2, 0; f_{\text{trap}}(\cdot); 1, 4)^T \quad (5.12)$$

o wartościach poszczególnych graczy zebranych w tabeli 5.1.

Algorytm symulacyjnego wyżarzania

Zrealizowaną implementację algorytmu symulacyjnego [KGV83] wyżarzania można opisać w następujących krokach:

1. **Temperatura początkowa.** Przyjmowana jest wartość temperatury początkowej $T_p = 10^7$.
2. **Punkt startowy.** Wybierany jest losowy punkt $p \in \mathcal{G}_D$.
3. **Ocena punktu.** Dla punktu p przeprowadzana jest symulacja chodu i przypisywana ocena m równa graczowi powabnemu G_p .
4. **Temperatura aktualna.** Wyliczana jest aktualna temperatura $T = T_p(1 - \frac{k}{K})$, gdzie $k \in \{0, 1, 2, \dots, K - 1\}$ to numer iteracji, a K to liczba zleconych iteracji.
5. **Losowy punkt z sąsiedztwa.** Losowany jest punkt $p_n \in \mathcal{S}$, gdzie $\mathcal{S}$ to sąsiedztwo punktu p . Do $\mathcal{S}$ należą wszystkie punkty,
 - (a) których każda współrzędna różni się od odpowiadającej jej współrzędnej punktu p ,
 - (b) oraz przyjmują najbliższą możliwą wartość ze zbioru przypisanego danej współrzędnej parametru (wartości $f_{\text{tria}}(\cdot)$, $f_{\text{trap}}(\cdot)$, $f_{\text{parab}}(\cdot)$ traktujemy jak liczby całkowite, odpowiednio 0, 1, 2).
6. **Ocena nowego punktu.** Punktowi p_n przypisywana jest ocena m_n analogicznie jak dla punktu p
7. **Przeskok.** Jako nowy punkt p przyjmowany jest punkt p_n z prawdopodobieństwem

$$P = \begin{cases} 1 & \text{dla } m_n > m \\ \exp(\frac{m_n - m}{T}) & \text{dla } m_n \leq m \end{cases} \quad (5.13)$$

8. **Wynik.** Zwracany jest napotkany przez algorytm punkt, dla którego wyznaczono największy gracz powabny G_p .

Kroki od 4 do 7 powtarzane są K razy. Poszukiwanie zrealizowano dla wartości $K = 250$. W wyniku otrzymano punkt

$$p_{SA} = (5/6; 1, 9; 1, 0; 0, 9; f_{\text{tria}}(\cdot); 1, 9)^T \quad (5.14)$$

o wartościach poszczególnych graczy zebranych w tabeli 5.2.

Tabela 5.2: Wartości poszczególnych graczy dla punktu p_{SA}

$G_Z [10^{-7}]$	$G_B [10^{-1}]$	$G_E [10^{-2}]$	$G_F [10^{-4}]$	$G_S [10^{-1}]$	$G_P [10^{-6}]$
2,35	1,80	1,23	6,86	4,67	6,99

Algorytm przeszukiwania tabu

Zrealizowaną implementację algorytmu poszukiwania tabu [GL97] można opisać w następujących krokach:

1. **Lista tabu.** Definiujemy zbiór tabu jako zbiór pusty $\mathcal{T} = \emptyset$.
2. **Punkt startowy.** Wybierany jest losowy punkt $p \in \mathcal{G}_D^6$.
3. **Sąsiedztwo.** Do zbioru $\mathcal{S}$ przypisywane jest sąsiedztwo punktu p . Do $\mathcal{S}$ należą wszystkie punkty,
 - (a) które różnią się od punktu p wartością dokładnie jednej współrzędnej,
 - (b) oraz przyjmują najbliższą możliwą wartość ze zbioru odpowiadającego danej współrzędnej parametru (wartości $f_{\text{tria}}(\cdot)$, $f_{\text{trap}}(\cdot)$, $f_{\text{parab}}(\cdot)$ traktujemy jak liczby całkowite, odpowiednio 0, 1, 2).
4. **Ocena sąsiedztwa.** Dla każdego punktu $p_i \in \mathcal{S}$ przeprowadzana jest symulacja chodu i przypisywana ocena m_i równa graczowi powabnemu G_P .
5. **Wybór najlepszego sąsiada.** Wybierany jest punkt $p' \in \mathcal{S}$,
 - (a) który posiada przypisaną najwyższą ocenę spośród punktów należących do $\mathcal{S}$,
 - (b) oraz wektor swobodny $v = \overrightarrow{p'p}$ nie należy do $\mathcal{T}$
6. **Podstawienie nowego punktu.** Jeśli p' istnieje, to za nowy punkt p przyjmowany jest punkt p' . W przeciwnym wypadku punkt p nie ulega zmianie.
7. **Zapominanie.** Ze zbioru $\mathcal{T}$ usuwany jest wektor, który znajdował się tam najdłużej.
8. **Tabu.** Do zbioru $\mathcal{T}$ dołączany jest wektor v .
9. **Wynik.** Zwracany jest napotkany przez algorytm punkt, dla którego wyznaczono największy gracz powabny G_P .

Kroki od 3 do 8 powtarzane są N razy. Poszukiwanie zrealizowano dla wartości $N = 25$. W wyniku otrzymano punkt

$$p_{TS} = ({}^5/6; 1, 8; 2, 3; 1, 7; f_{\text{tria}}(\cdot); 0, 8)^T \quad (5.15)$$

o wartościach poszczególnych graczy zebranych w tabeli 5.3.

Tabela 5.3: Wartości poszczególnych gracjanów dla punktu p_{TS}

$G_Z [10^{-7}]$	$G_B [10^{-1}]$	$G_E [10^{-2}]$	$G_F [10^{-4}]$	$G_S [10^{-1}]$	$G_P [10^{-6}]$
1,16	5,50	1,11	7,14	2,20	4,69

Tabela 5.4: Wyniki algorytmów dla wektora subiektywności premiującego zgrabność

Algorytm	ρ^T
GA	$(5/6; 1, 8; 1, 0; 1, 7; f_{\text{tria}}(\cdot); 0, 8)$
SA	$(1/2; 1, 6; 1, 1; 1, 2; f_{\text{tria}}(\cdot); 1, 0)$
TS	$(1/2; 1, 9; 2, 0; 1, 7; f_{\text{tria}}(\cdot); 0, 8)$

Tabela 5.5: Wartości gracjanów dla wyników algorytmów dla wektora subiektywności premiującego zgrabność

Algorytm	$G_Z [10^{-7}]$	$G_B [10^{-1}]$	$G_E [10^{-2}]$	$G_F [10^{-4}]$	$G_S [10^{-1}]$	$G_P [10^{-7}]$
GA	2,39	5,48	1,34	6,99	3,38	5,01
SA	2,84	5,54	0,57	6,89	2,18	4,68
TS	2,74	5,84	0,76	6,89	1,78	4,84

5.5 Wnioski

Z przedstawionych powyżej wyników badań oraz obserwacji można wyciągnąć wnioski:

1. Badanie estetyki jest złożonym, nieliniowym problemem, ponieważ zmiana jednego parametru chodu wywiera różny wpływ na różne gracjany.
2. Różne odpowiedzi gracjanów na te same zmiany parametrów chodu pokazują, że zostały poprawnie dobrane. Oceniają chód pod różnymi, być może niezależnymi od siebie nawzajem aspektami albo zależnymi w małym stopniu.
3. Zastosowane algorytmy zwróciły trzy znacząco różne od siebie punkty, może to oznaczać, że trafiły w różne minima lokalne.
4. Duży wpływ na możliwy do zbadania obszar przestrzeni parametrów chodu ma algorytm sterowania, który determinuje dopuszczalne zakresy parametrów chodu.
5. Zdominowanie kształtem wykresu gracjanu powabnego przez gracjan subtelny oznacza, że dla wybranego wektora subiektywności najłatwiej osiągnąć piękno poprzez stabilizację chodu.

Możemy modyfikować to, jaki parametr chodu będzie faworyzowany, poprzez dobór odpowiedniego wektora subiektywności. Dla przykładu, wyniki algorytmów dla wektora $k = (10^8, 10, 10^3, 10^4, 1)^T$ (premiującego minimalizację zrywu) zebrano w tabeli 5.4 a odpowiadające im wartości gracjanów w tabeli 5.5.

Rozdział 6

Podsumowanie

Człowiek stając się istotą rozumną, zaczął postrzegać świat na wyższym poziomie niż kierującym się tylko wolą przetrwania i prokreacji. Jako jedyne ze zwierząt wystąpił ponad nie i wzniosł pysznie głowę nad warstwę brudu, krwi i potu. Począł zauważać uporządkowanie świata, jego harmonijny charakter i rozumieć prawa nim rządzące. Nauczył się myśleć przyczynowo i zadawać pytanie, które jest rodzicem wszystkich wielkich odkryć – dlaczego? W końcu musiał zadać pytanie: dlaczego jestem?, dlaczego myślę? I tak rozpoczął rozważania na temat początków świata.

Odpowiedzi udzielił w księdze Genesis [Rod03], w której opisał Boskie stworzenie wszechrzeczy. Dowiadujemy się z niej, że Bóg stworzył materię, powołał życie i na końcu uczynił człowieka na swoje podobieństwo, jako zwieńczenie dzieła stworzenia. I czytamy tam dalej:

„A Bóg widział, że wszystko, co uczynił, było bardzo dobre. I tak upłynął wieczór i poranek – dzień szósty”

Już w XIII w. jeden z doktorów Kościoła, Tomasz z Akwinu łączył ze sobą trzy wartości [Akw86]:

- bonum – dobro,
- verum – prawdę,
- pulchrum – piękno.

Więc czyż można odmówić czemukolwiek piękna, skoro wszystko zostało stworzone jako bardzo dobre? Czy cokolwiek co istnieje jest nieprawdziwe? Albo czy jakiegokolwiek istnienie powołano jako złe?

Nauka próbuje rozbierać świat na coraz to mniejsze części. Aktualnie dominuje przekonanie, że to co nas otacza zbudowane jest z 12 cząstek elementarnych – fermionów i ich odpowiednich antycząstek [NAG⁺24]. Jednak czy nie powinniśmy spojrzeć na sprawę szerzej i wejść na poziom meta? A co, jeśli świat składa się z trzech metacząstek będących fundamentem świata? Tak podstawowych, że nieposiadających postaci materialnej. Cząstek dobra, prawdy i piękna.

Uprawiając matematykę i poznając coraz to nowsze występujące w niej, skomplikowane struktury nie da się nie odnieść wrażenia, że cząstki te tam są. Wielu matematyków mówi też o odkrywaniu obiektów matematycznych, a nie o ich opracowywaniu. Badamy miejsce, które jest pozbawione materii, lecz przez to nie mniej

rzeczywiste. Zbudowane przede wszystkim z cząstek prawdy, bo język matematyki nie potrafi kłamać.

Jeśli uznamy słuszność przyjęcia dobra, prawdy i piękna za podstawowe składowe wszechświata, to badanie i próba pomiaru piękna są co najmniej tak samo zasadne, jak prace nad modelem standardowym [NAG⁺24]. A może i bardziej.

6.1 Rola gracjanów w ocenie estetyki

Próba pomiaru, ilościowego przedstawienia czegoś tak bardzo nienamacalnego i zarazem powszechnego jak pojęcie estetyki, piękna okazała się być czymś bardziej rzeczywistym niż mogło się wydawać. Zmusiła do myślenia nieszablonowego i przekrojowego łączącego ze sobą wiedzę z różnych dziedzin nauki oraz wymagała spojrzenia na poznane zagadnienia w odmienny sposób. Idealnie oddaje to sedno robotyki, która uczy nas przyjmować różne punkty widzenia, a właściwie to układy współrzędnych oraz opisywać relacje między nimi.

Odkryte i przytoczone w pracy gracjany okazały się interesującym narzędziem do porównywania piękna chodu robota. Podstawy analizy funkcjonalnej i topologii, stojące za tymi wskaźnikami, wydają się właściwymi narzędziami do tego zadania. Naszym zdaniem ich rozwinięcia należy szukać w właśnie tych działach matematyki. Być może do opisu czegoś tak bardzo wyrafinowanego jak piękno, potrzeba równie wyrafinowanych struktur. Czegoś bardziej wyszukanego niż prostych przestrzeni funkcyjnych.

Przedstawione w pracy wyniki uwypukliły złożoność problemu ilościowej oceny piękna i optymalizacji chodu według tego kryterium. Zagadnienie to wymaga dalszych, dogłębnych badań.

6.2 Kierunki dalszych badań

W pracy rozwinięto tylko w niewielkim stopniu teorię gracjanów. Pracę można kontynuować w następujący sposób:

1. **Liczba i rodzaje gracjanów.** Podstawowym problemem, który wymaga rozwinięcia jest lepsze dobranie i przetestowanie większej liczby gracjanów, badających inne dane wyjściowe lub przetwarzających w odmienny sposób te przytoczone w pracy. Należałoby wybrać te, które badają piękno w jak najbardziej zróżnicowany sposób, uwzględniając jak najwięcej aspektów, a jednocześnie, aby samych wskaźników nie było zbyt wiele.
2. **Badanie wektora subiektywności.** Z gracjanami ściśle związany jest wektor subiektywności, który w pracy przyjęto a priori, a to właśnie od niego w dużej mierze zależy ocena końcowa. Badania nad nim polegałyby na przeprowadzeniu sondaży wśród ludzi, w których ocenialiby ruchy robotów w sposób ilościowy. Następnie powiązaniu ocen z konkretnymi gracjanami i estymowaniu statycznym wartości współrzędnych wektora.

3. **Sterowanie dynamiczne.** Naturalnym rozwinięciem zrealizowanych prac jest rozwinięcie struktury sterowania o model dynamiczny robota i sterowanie poprzez siły, a nie prędkości. Pozwoliłoby to na dokładniejsze i bardziej precyzyjne ruchy i tym samym lepsze badanie gracjanów, na które nie będą wpływać błędy sterowania.

Umożliwiłoby to również rozszerzenie badań na chody stabilne dynamiczne, a więc między innymi bieg robota, na akrobacje atletyczne, jak przewroty w przód lub salta, czy na taniec, w tym piruety i efektowne wymachy nóg.

Również możliwym jest opracowanie układu sterowania poprzez optymalizację piękna lub z wykorzystaniem sprzężenia od stanu piękna. Jednak do tego zagadnienia konieczne byłoby rozszerzenie gracjanów do oceny obiektów w pojedynczych chwilach.

4. **Piękno śladowe.** W pracy wyznaczano piękno na podstawie pomiarów wykonanych w trakcie chodu. Możliwym jest również odtworzenie tych pomiarów na podstawie badania odcisniętych w podłożu śladów. Będą one zmieniać swój kształt, głębokość, odległość między sobą w zależności od rodzaju chodu czy nawet biegu.
5. **Wielkość gracjanów.** Przedstawione w pracy gracjany zwracają dla badanego ruchu robota wartości różniące się nawet o sześć rzędów wielkości. Dysproporcje te możemy korygować poprzez odpowiednie współczynniki wektora subiektywności, jednak wydaje się to być tylko niezbyt udolną protezą. Wskaźniki powinny same z siebie mieć jednakowy wpływ na ocenę tzn. wektor o wszystkich współrzędnych wartości 1 odpowiada sytuacji, gdy gracjany mają jednakowy wpływ na wartość gracjanu powabnego.
6. **Kategorie piękna.** W pracy przedstawiono badanie obrazu przestrzeni parametrów chodu $\mathcal{G}^n$, gdzie n jest liczbą parametrów chodu. Jednak dla tego samego n można stworzyć zupełnie inną przestrzeń poprzez dobór tej samej liczby innych parametrów. Wszystkie te przestrzenie powinny tworzyć jeden wspólny twór matematyczny. Możliwe, że można padać relacje i transformacje między nimi oraz to jak wpływają na swój obraz – przestrzeń gracji $\mathcal{G}$. Do tego celu pomocne może okazać się pojęcie kategorii [Wika].
- Niezależnie czy użycie pojęcia kategorii okaże się właściwe czy też nie, zważywszy na to, że był ten (niesklasyfikowany lub szczególny przypadek kategorii) będzie opisywał w pewnym sensie przestrzeń przestrzeni, proponujemy nazywać go przestrzeniałem, a dokładniej przestrzeniałem parametrów chodu.
7. **Rozszerzenie zastosowania gracjanów.** Interesującym zagadnieniem pozostaje próba zastosowania gracjanów do oceny obiektów, struktur innych niż ruch, w szczególności chód. Możliwe nawet, że istnieją metagracjany, które potrafią ocenić piękno samych struktur matematycznych. Wśród fizyków panuje przekonanie, że teoria opisująca świat powinna być elegancka, piękna. Przy pomocy właśnie metagracjanów moglibyśmy próbować szukać najpiękniejszej struktury do opisu świata.

Jeszcze ciekawszym zagadnieniem pozostaje zastosowanie metagracjanów do oceny samych siebie. Czy w tej sytuacji powinny one zachowywać się narcystycznie i zwrócić nieskończoność? Czy też jakąś bardziej wysublimowaną wartość jak $\aleph_0$ czy c ?

Sugestie te pozostawiamy do rozwinięcia w ramach tworzenia nowej, tymczasowo nazwanej, Teorii Miar Piękna (TMP).

6.3 Słowo końcowe

W pracy dogłębnie przedstawiono problem oceny piękna od strony filozoficznej jak i matematycznej. Układ sterowania pozwala na przemieszczanie się robota. Algorytmy przeszukujące przestrzeń gracji & optymalizują parametry chodu pod względem estetyki. Postawiony cel pracy uznajemy za zrealizowany.

Na koniec, po wielu monologach wewnętrznych i dialogach z otoczeniem na temat sensowności idei pomiaru piękna możemy sformułować najważniejszy wniosek. A mianowicie:

„O gustach się nie dyskutuje”.

Dlatego należy je mierzyć.

C.N.D.

Literatura

- [Akw86] Tomasz z Akwinu, *Suma teologiczna*. Katolicki Ośrodek Wydawniczy "Veritas", Londyn, 1962–1986.
- [Arya] Arystoteles, *Metafizyka*. IV w. p.n.e.
- [Aryb] Arystoteles, *O niebie*. IV w. p.n.e.
- [Aryc] Arystoteles, *Poetyka*. IV w. p.n.e.
- [Bau50] A. G. Baumgarten, *Aesthetica*. Impens. I. C. Kleyb, 1750.
- [Bel61] R. E. Bellman, *Adaptive Control Processes: A Guided Tour*. Princeton University Press, Princeton, New Jersey, 1961.
- [Bir33] G. D. Birkhoff, *Aesthetic Measure*. Harvard University Press, 1933.
- [Ble] Blender. <https://www.blender.org/>.
- [Box79] G. E. P. Box, Robustness in the strategy of scientific model building. Technical Report ADA070213, Defense Technical Information Center (DTIC), 1979.
- [Che53] M. Chekhov, *To the Actor: On the Technique of Acting*. Harper & Row, 1953.
- [Ein05a] A. Einstein, Ist die trägheit eines körpers von seinem energiehinhalt abhängig? *Annalen der Physik*, 323(13):639–641, 1905.
- [Ein05b] A. Einstein, Zur elektrodynamik bewegter körper. *Annalen der Physik*, 322(10):891–921, 1905.
- [Enc] Encyclopedia of Mathematics. Pseudo-metric. <https://encyclopediaofmath.org/wiki/Pseudo-metric>.
- [FH85] T. Flash, N. Hogan, The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neuroscience*, 5(7):1688–1703, 1985.
- [FKMP21] Z. Fu, A. Kumar, J. Malik, D. Pathak, Minimizing energy consumption leads to the emergence of gaits in legged robots. W: *Conference on Robot Learning (CoRL)*, 2021.
- [GL97] F. Glover, M. Laguna, *Tabu Search*. Kluwer Academic Publishers, 1997.
- [goo] goodreads.com. Quotes by Albert Einstein. https://www.goodreads.com/author/show/9810.Albert_Einstein.
- [HD64] R. S. Hartenberg, J. Denavit, *Kinematic Synthesis of Linkages*. McGraw-Hill, New York, 1964.

- [Her59] Herodot, *Dzieje*. Czytelnik, Warszawa, 1959.
- [Hol92] J. H. Holland, Genetic algorithms. *Scientific American*, 267(1):66–73, 1992.
- [HW98] Ch. M. Harris, D. M. Wolpert, Signal-dependent noise determines motor planning. *Nature*, 394(6695):780–784, 1998.
- [Kan90] I. Kant, *Kritik der Urteilskraft*. Lagarde und Friederich, 1790.
- [KGV83] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- [Las] Lascaux cave. Unesco. World Heritage Convention. 17 000 – 15 000 lat p.n.e., <https://whc.unesco.org/en/documents/108443/>.
- [Muj] MuJoCo. <https://mujoco.org/>.
- [NAG⁺24] S. Navas, C. Amsler, T. Gutsche, C. Hanhart, J. J. Hernández-Rey, C. Lourenço, A. Masoni, M. Mikhasenko, R. E. Mitchell, C. Patrignani, C. Schwan-da, S. Spanier, G. Venanzoni, C. Z. Yuan, K. Agashe, G. Aielli, B. C. Allanach, J. Alvarez-Muñiz, M. Antonelli, E. C. Aschenauer, D. M. Asner, K. Assamagan, H. Baer, Sw. Banerjee, R. M. Barnett, L. Baudis, C. W. Bauer, J. J. Beatty, J. Beringer, A. Bettini, O. Biebel, K. M. Black, E. Blucher, R. Bonventre, R. A. Briere, A. Buckley, V. D. Burkert, M. A. Bychkov, R. N. Cahn, Z. Cao, M. Carena, G. Casarosa, A. Ceccucci, A. Cerri, R. S. Chivukula, G. Cowan, K. Cranmer, V. Crede, O. Cremonesi, G. D’Ambrosio, T. Damour, D. de Florian, A. de Gouvêa, T. DeGrand, S. Demers, Z. Demiragli, B. A. Dobrescu, M. D’Onofrio, M. Doser, H. K. Dreiner, P. Eerola, U. Egede, S. Eidelman, A. X. El-Khadra, J. Ellis, S. C. Eno, J. Erler, V. V. Ezhela, A. Fava, W. Fetscher, B. D. Fields, A. Freitas, H. Gallagher, T. Gershon, Y. Gershtein, T. Gherghetta, M. C. Gonzalez-Garcia, M. Goodman, C. Grab, A. V. Gritsan, C. Grojean, D. E. Groom, M. Grünewald, A. Gurtu, H. E. Haber, M. Hamel, S. Hashimoto, Y. Hayato, A. Hebecker, S. Heinemeyer, K. Hikasa, J. Hisano, A. Höcker, J. Holder, L. Hsu, J. Huston, T. Hyodo, Al. Ianni, M. Kado, M. Karliner, U. F. Katz, M. Kenzie, V. A. Khoze, S. R. Klein, F. Krauss, M. Kreps, P. Križan, B. Krusche, Y. Kwon, O. Lahav, L. P. Lellouch, J. Lesgourgues, A. R. Liddle, Z. Ligeti, C.-J. Lin, C. Lippmann, T. M. Liss, A. Lister, L. Littenberg, K. S. Lugovsky, S. B. Lugovsky, A. Lusiani, Y. Makida, F. Maltoni, A. V. Manohar, W. J. Marciano, J. Matthews, U.-G. Meißner, I.-A. Melzer-Pellmann, P. Mertsch, D. J. Miller, D. Milstead, K. Mönig, P. Molaro, F. Moortgat, M. Moskvic, N. Nagata, K. Nakamura, M. Narain, P. Nason, A. Nelles, M. Neubert, Y. Nir, H. B. O’Connell, C. A. J. O’Hare, K. A. Olive, J. A. Peacock, E. Pianori, A. Pich, A. Piepke, F. Pietropaolo, A. Pomarol, S. Pordes, S. Profumo, A. Quadt, K. Rabbertz, J. Rademacker, G. Raffelt, M. Ramsey-Musolf, P. Richardson, A. Ringwald, D. J. Robinson, S. Roesler, S. Rolli, A. Romaniouk, L. J. Rosenberg, J. L. Rosner, G. Rybka, M. G. Ryskin, R. A. Ryutin, B. Safdi, Y. Sakai, S. Sarkar, F. Sauli, O. Schneider, S. Schönert, K. Scholberg, A. J. Schwartz, J. Schwiening, D. Scott, F. Sefkow, U. Seljak, V. Sharma, S. R. Sharpe, V. Shiltsev, G. Signorelli, M. Silari, F. Simon, T. Sjöstrand, P. Skands, T. Skwarnicki, G. F. Smoot, A. Soffer, M. S. Sozzi, C. Spiering,

- A. Stahl, Y. Sumino, F. Takahashi, M. Tanabashi, J. Tanaka, M. Taševský, K. Terao, K. Terashi, J. Terning, U. Thoma, R. S. Thorne, L. Tiator, M. Titor, D. R. Tovey, K. Trabelsi, P. Urquijo, G. Valencia, R. Van de Water, N. Varelas, L. Verde, I. Vivarelli, P. Vogel, W. Vogelsang, V. Vorobyev, S. P. Wakely, W. Walkowiak, C. W. Walter, D. Wands, D. H. Weinberg, E. J. Weinberg, N. Wermes, M. White, L. R. Wiencke, S. Willocq, C. L. Woody, R. L. Workman, W.-M. Yao, M. Yokoyama, R. Yoshida, G. Zanderighi, G. P. Zeller, R.-Y. Zhu, S.-L. Zhu, F. Zimmermann, P. A. Zyla, J. Anderson, M. Kramer, P. Schaffner, W. Zheng, Review of particle physics. *Phys. Rev. D*, 110:030001, Aug 2024.
- [New87] I. Newton, *Philosophiæ Naturalis Principia Mathematica*. Londyn, 1687.
- [Num] NumPy. <https://numpy.org/>.
- [Pac09] L. Pacioli, De divina proportione, 1509.
- [Phy] Python. <https://www.python.org/>.
- [Pia50] D. da Piacenza. De arte saltandi et choreas ducendi, ok. 1450.
- [Pla] Platon, Państwo. IV w. p.n.e.
- [Rod03] Księga rodzaju. W: *Biblia Tysiąclecia: Pismo Świąte Starego i Nowego Testamentu*. Wydawnictwo Pallottinum, Poznań, 2003.
- [Vin90] L. da Vinci. Człowiek witruwiański. Gallerie dell'Accademia, ok. 1490.
- [Wie48] N. Wiener, *Cybernetics: Or Control and Communication in the Animal and the Machine*. John Wiley & Sons, New York, 1948.
- [Wika] Wikipedia. Kategoria (matematyka). [https://pl.wikipedia.org/wiki/Kategoria_\(matematyka\)](https://pl.wikipedia.org/wiki/Kategoria_(matematyka)).
- [Wikb] Wikipedia. Metoda elementów skończonych. https://pl.wikipedia.org/wiki/Metoda_element%C3%B3w_sko%C5%84czonych.
- [Wikc] Wikipedia. Równania Eulera-Lagrange'a. https://pl.wikipedia.org/wiki/R%C3%B3wnania_Eulera-Lagrange%E2%80%99a.
- [Wikd] Wikipedia. Zasada najmniejszego działania. https://pl.wikipedia.org/wiki/Zasada_najmniejszego_dzia%C5%82ania.
- [Ypm95] T. J. Ypma, Historical development of the Newton-Raphson method. *SIAM Review*, 37(4):531–551, 1995.
- [Zie14] T. Zielińska, *Maszyny Kroczące*. Wydawnictwo Naukowe PWN, 2014.

Spis rysunków

2.1	Model robota w symulacji MuJoCo	7
2.2	Układy nóg (a) gada – staw kolanowy zgięty pod kątem $\frac{\pi}{2}$, (b) owada – staw kolanowy ponad płaszczyzną tułowia, (c) ssaka – płaszczyzna ruchu nogi równoległa do grzbietu [Zie14]	7
2.3	Części składowe robota	8
2.4	Współrzędne biegunowe dla korpusu robota	9
2.5	Schematyczny model robota z przypisanymi układami współrzędnych	10
2.6	Schemat kinematyczny nogi robota	11
3.1	Robot w pozycji zerowej	13
3.2	Robot w pozycji startowej	13
3.3	Diagram chodu dla chodu czwórpodporowego	14
3.4	Struktura sterowania	16
3.5	Główna pętla algorytmu sterowania	18
3.6	Metoda go(.)	19
3.7	Metoda stand_up(.)	20
3.8	Metoda gait(.)	21
3.9	Funkcje kroku	22
3.10	Metoda step(.)	23
3.11	Metoda stepPlanning(.)	24
3.12	Metoda getStepFunc(.)	25
3.13	Funkcja s(t, T) dla T=3	26
3.14	Funkcja h(t, T, f) dla T=3	26
3.15	Metoda support(.)	27
3.16	Metoda supportPlanning(.)	28
3.17	Metoda getx2q(.)	29
3.18	Metoda setPos(.)	30
5.1	Wartości gracjanu zgrabnego w zależności od zmian parametrów chodu	42
5.2	Wartości gracjanu błędnego w zależności od zmian parametrów chodu	43
5.3	Wartości gracjanu energetycznego w zależności od zmian parametrów chodu	45
5.4	Wartości gracjanu finezyjnego w zależności od zmian parametrów chodu	46
5.5	Wartości gracjanu subtelnego w zależności od zmian parametrów chodu	48
5.6	Wartości gracjanu powabnego w zależności od zmian parametrów chodu	49