

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI

KIERUNEK: Automatyka i Robotyka (AIR)
SPECJALNOŚĆ: Robotyka (ARR)

**PRACA DYPLOMOWA
MAGISTERSKA**

Laboratoryjny robot kroczący

Laboratory walking robot

AUTOR:
Rafał Gierczak

PROWADZĄCY PRACĘ:
dr inż. Robert Muszyński

OCENA PRACY:

Pracę dedykuję wszystkim, którzy nie stracili wiary w jej ukończenie. Szczególnie chciałbym podziękować Panu dr. Robertowi Muszyńskiemu za wyrozumiałość oraz poświęcony czas, a także rodzinie i bliskim za okazaną cierpliwość.

Spis treści

1	Wstęp	1
1.1	Cel i zakres pracy	2
2	Podstawy teoretyczne chodu robotów	5
2.1	Kinematyka robota	5
2.1.1	Proste zadanie kinematyki nogi robota	6
2.1.2	Odwrotne zadanie kinematyki nogi robota	8
2.2	Algorytmy chodu robota sześcionożnego	9
2.2.1	Chód pięciopodporowy	9
2.2.2	Chód mieszany	10
2.2.3	Chód czteropodporowy	10
2.2.4	Chód trójpodporowy	10
2.3	Planowanie ruchu	12
3	Konstrukcja robota	15
3.1	Konstrukcja mechaniczna i napędy	15
3.2	Układy elektroniczne	16
3.2.1	Płyta główna	18
3.2.2	Serwokontroler	18
3.2.3	Zasilanie	19
3.3	Oprogramowanie	19
3.3.1	Algorytm generowania chodu	19
3.3.2	Odbiór sygnału z aparatury zdalnego sterowania	23
4	Testy funkcjonalne	25
5	Podsumowanie	29
A	Rysunki techniczne	31
B	Ideowy schemat elektroniczny	35
C	Opis złącz	39
D	Zawartość płyty CD	41
D.1	Rysunki techniczne	41
D.2	Dokumentacja	41
D.3	Oprogramowanie	41

E Instrukcja obsługi	43
E.1 Obsługa robota	43
E.1.1 Kalibracja serwomechanizmów	43
E.1.2 Kalibracja układu zdalnego sterowania	44
E.1.3 Ładowanie akumulatorów	44
E.1.4 Demontaż robota	44
E.2 Programowanie	44
E.2.1 Przywrócenie oryginalnego oprogramowania	44
E.3 Inne	45
E.3.1 Komunikacja za pośrednictwem interfejsu RS-232	45
F Plik parametrów chodu	47
Bibilografia	49

Spis rysunków

1.1	Straszak diabelski [16]	3
2.1	Robot w układzie współrzędnych	6
2.2	Kinematyka pojedynczej nogi robota	7
2.3	Rzut nogi na płaszczyznę OXZ (dla $q_1 = 0$)	8
2.4	Diagram chodu pięciopodporowego	10
2.5	Diagram chodu mieszanego	11
2.6	Diagram chodu czteropodporowego	11
2.7	Diagram chodu trójpodporowego	12
2.8	Robot podczas chodu po łuku	14
3.1	Projekt robota	16
3.2	Podstawowe wymiary robota – widok z góry	17
3.3	Podstawowe wymiary nogi robota	17
3.4	Płyta główna przed montażem elementów	18
3.5	Diagram pętli głównej programu robota	20
3.6	Metoda wyznaczenia położenia nogi na łuku	22
3.7	Metoda pomiaru wypełnienia sygnału PWM	23
3.8	Przebieg sygnału PWM przed filtracją oraz po filtracji	24
4.1	Wykres nastaw nóg 1P oraz 4P podczas chodu pięciopodporowego na wprost	26
4.2	Wykres nastaw nóg 1P oraz 4P podczas chodu pięciopodporowego po łuku o promieniu $300mm$	26
4.3	Wykres nastaw nóg 1P oraz 4P podczas chodu pięciopodporowego po łuku o promieniu $5mm$	27
A.1	Rysunki techniczne elementów nogi robota	32
A.2	Rysunek techniczny oraz wymiary płyty spodniej	33
B.1	Schemat ideowy płyty głównej	36
B.2	Schemat ideowy układu zasilania płyty głównej	37
B.3	Schemat ideowy serwokontrolera SSC-32 [10]	38
E.1	Sposób demontażu robota	45
E.2	Możliwości komunikacji za pomocą serwokontrolera SSC-32	45

Spis tabel

C.1	Opis podstawowych złącz na płycie głównej robota	39
C.2	Opis dodatkowych złącz na płycie głównej robota	40
F.1	Lista parametrów chodu	47

Rozdział 1

Wstęp

Już od początku ludzkich dziejów, człowiek w swoich działaniach stara się naśladować genialne rozwiązania natury. Trudno wskazać wynalazek, którego wczesny prototyp nie powstał z obserwacji otaczającego świata. Geneza ta dotyczy także konstruowanych przez nas środków transportu i chociaż ze względów ekonomicznych najpopularniejszym środkiem lokomocji pozostaje transport kołowy, to warto pamiętać, że w naturalnym i nie przygotowanym przez człowieka terenie, koła ustępują sprawnym nogom i dobremu poczuciu równowagi. Z tej przyczyny temat konstrukcji maszyny kroczącej regularnie powraca do umysłów wynalazców, których dzieła z mniejszym lub większym sukcesem naśladują sposoby poruszania się zwierząt, począwszy od samych ludzi, poprzez zwierzęta czworonogie jak konie, psy czy owady, aż po niespotykane w naturze, wielonożne hybrydy będące owocem fantazji ich twórców. Niniejsza praca porusza zagadnienia dotyczące robotów sześcionożnych ze względu na ich dużą uniwersalność oraz możliwość łatwego wygenerowania wielu różnych algorytmów chodu. Nie bez powodu lokomocja na sześciu nogach jest bardzo popularna w naturze – poruszają się w ten sposób m.in. karaluchy, mrówki, muchy oraz patyczaki.

Roboty sześcionożne zwane także heksapodami cechują się dużą stabilnością [4], nawet przenosząc trzy nogi jednocześnie, robot nadal opiera się na pozostałych trzech, co pozwala na zachowywanie równowagi. W pewnym uproszczeniu, sześcionożny robot kroczący to połączenie sześciu niemal identycznych manipulatorów pełniących funkcję odnóży. Najprostsze konstrukcje posiadają nogi rozmieszczone w dwóch równoległych rzędach po obu stronach robota, jednakże lepszym rozwiązaniem jest umocowanie nóg na planie elipsy, ze środkowymi nogami wysuniętymi nieznacznie na zewnątrz, co pozwala na zwiększenie zakresu bezkolizyjnego ruchu nóg. Innym znanym rozwiązaniem jest umocowanie nóg na wierzchołkach sześciokąta foremnego – układ taki sprzyja częstym zmianom kierunku marszu. Opisywana konstrukcja posiada nogi rozmieszczone na planie elipsy, jako że celem jest umożliwienie marszu na wprost oraz po łuku.

Istnieje kilka popularnych sposobów poruszania się na sześciu nogach [13]. Najpopularniejszy – chód owadów – polega na kolejnym przenoszeniu wszystkich odnóży (pojedynczo lub grupami) w celu nadania tułowiowi prędkości w kierunku anatomicznego przodu. Nieco inną strategię chodu przyjęły wędrujące „bokiem” kraby – taki sposób przemieszczania pozwala im osiągać większą prędkość niż podczas chodu na wprost, gdyż ich nogi są usytuowane bardzo blisko siebie i przy dłuższym kroku zwyczajnie zaczęłyby się zderzać. Jeszcze inną strategię przyjęły modliszki, które najczęściej poruszają się tylko na 4 tylnych odnóżach, podczas gdy dwie przednie znacznie lepiej rozwinięte nogi służą do chwytania ofiary. Chód można podzielić na dwie kategorie, rozróżniamy chód statyczny oraz chód dynamiczny. Chód statyczny pozwala na zatrzymanie się w dowolnym momencie bez ry-

zyka utraty równowagi oraz upadku, natomiast podczas chodu dynamicznego występują chwile podczas których środek ciężkości w kontrolowany sposób przesuwany jest poza obszar stabilności. Ze względu na stosunkowo niewielką prędkość poruszania się oraz liczbę nóg pozwalającą na zachowanie co najmniej trzech punktów podparcia, owady poruszają się chodem statycznym.

W ostatnich latach, wraz z szybkim postępem technologicznym, znacznie wzrosło zainteresowanie tematem robotów kroczących. Zaowocowało to licznymi opracowaniami naukowymi, takimi jak książka autorstwa Teresy Zielińskiej [17] lub dogłębnie analizująca temat "Climbing and Walking Robots" [12]. Prowadzonym pracom naukowym towarzyszy budowa wielu udanych konstrukcji, spośród których warto wyróżnić bardzo zaawansowanego robota LAURON [15], zbudowanego na uniwersytecie w Karlsruhe z myślą o udziale w akcjach ratunkowych. Konstrukcja ta została wyposażona w panoramiczną kamerę oraz bardzo precyzyjne czujniki na końcach nóg, które w połączeniu z pomiarem prądu poszczególnych napędów pozwalają na bezproblemowe poruszanie się w nierównym terenie. Istotnym z punktu widzenia niniejszej pracy jest charakteryzujący się bardzo płynnym chodem robot Phoenix firmy Lynxmotion [9] oraz zbudowany przez tą samą firmę Apod.

Impulsem do podjęcia badań w zakresie robotów kroczących oraz inspiracją w ich trakcie była obserwacja straszyków diabelskich (łac. *Peruphasma schultei*) – sześcionożnych owadów z rodziny patyczakowatych (zobacz rysunek 1.1). Budowa ciała wspomnianych owadów (trzy pary odnóży podobnej długości) pozwala na swobodne generowanie wielu zróżnicowanych chodów w zależności od potrzeby. Straszyki są lekkie, co w połączeniu z długimi oraz dosyć silnymi nogami, pozwala im poruszać się praktycznie po każdym podłożu, włączając w to pionowe (nawet śliskie) ściany. Owad występujący w swoim naturalnym środowisku, którym w wypadku straszyków może być gałąź ligustru, porusza się chodem swobodnym (ang. *Free gait*) uzależniając każdy następny krok od aktualnie dostępnego miejsca podparcia nogi. Jednak jeśli tego samego owada wypuścimy na płaskiej powierzchni to zaobserwujemy powtarzającą się sekwencję ruchów, zależną już tylko od tego, czy chce on poruszać się szybko – wykorzystując więcej energii (np. chodem trójpodporowym), czy może powoli ale oszczędnie (np. pełzając).

1.1 Cel i zakres pracy

Celem pracy jest konstrukcja oraz zaprogramowanie sześcionożnego robota kroczącego, który posłuży jako stanowisko laboratoryjne do badania algorytmów chodu robotów wielonożnych. Opracowywany w ramach projektu robot musi cechować się dużym zakresem ruchów oraz funkcjonalnością. Ponadto robot musi być wyposażony w uniwersalny interfejs umożliwiający łatwą zmianę parametrów chodu oraz tworzenie nowych algorytmów chodu. Ze względu na przeznaczenie robota, ważnym aspektem jest wytrzymałość jego konstrukcji, otwartość architektury oraz kompletność dokumentacji, co pozwoli na łatwą jego rozbudowę i zaplanowanie kolejnych ćwiczeń laboratoryjnych oraz ułatwi serwis w wypadku ewentualnych uszkodzeń lub konieczności wymiany elementów. Na koniec gotową konstrukcję należy poddać testom funkcjonalności.

Układ pracy jest następujący. W rozdziale 2 znajduje się wprowadzenie do teorii chodu wielopodporowego. Przedstawiono w nim szczegółowo kinematykę robota sześcionożnego, proste oraz odwrotne zadanie kinematyki, a także podstawowe algorytmy chodu robota sześcionożnego. Rozdział 3 zawiera opis rozwiązań konstrukcyjnych robota, informacje na temat zastosowanych napędów, sposobów zasilania. W dalszej jego części opisano oprogramowanie napisane na potrzeby projektu. W rozdziale 4 zaprezentowano wyniki prób



Rysunek 1.1 Straszyk diabelski [16]

działania robota dla różnych algorytmów chodu. Rozdział 5 podsumowuje całość. Rysunki techniczne poszczególnych elementów robota, schemat jego układów elektronicznych, opis złącz, instrukcję obsługi oraz spis zawartości załączonej płyty CD zamieszczono w dodatkach.

Rozdział 2

Podstawy teoretyczne chodu robotów

Odpowiednio zsynchronizowane przemieszczenia poszczególnych nóg pozwalają na wprawienie robota w ruch, który może odbywać się na wiele sposobów określanych mianem algorytmów chodu. Algorytmy definiują kolejność w jakiej przestawiane są nogi robota. Jednak aby możliwe było zastosowanie dowolnego algorytmu, konieczna jest znajomość budowy robota oraz wynikającego z niej algorytmu obliczania nastaw przegubów. W niniejszym rozdziale omówimy zagadnienia teoretyczne związane z poruszaniem się na sześciu nogach po płaskiej powierzchni oraz zaprezentujemy najbardziej popularne algorytmy chodu klasyfikując je pod względem liczby nóg jednocześnie opierających się o podłoże. Na koniec przyjrzymy się metodzie planowania ruchu nóg robota.

2.1 Kinematyka robota

Do wyznaczenia położenia końców nóg w funkcji nastaw ich przegubów konieczna jest znajomość prostej kinematyki robota. W trakcie realizacji chodu przydatne jest rozwiązanie odwrotnego zadania kinematyki, polegające na określeniu kątów przegubów nóg dla zadanych położenia ich końców. Oba te zagadnienia są przedstawione poniżej.

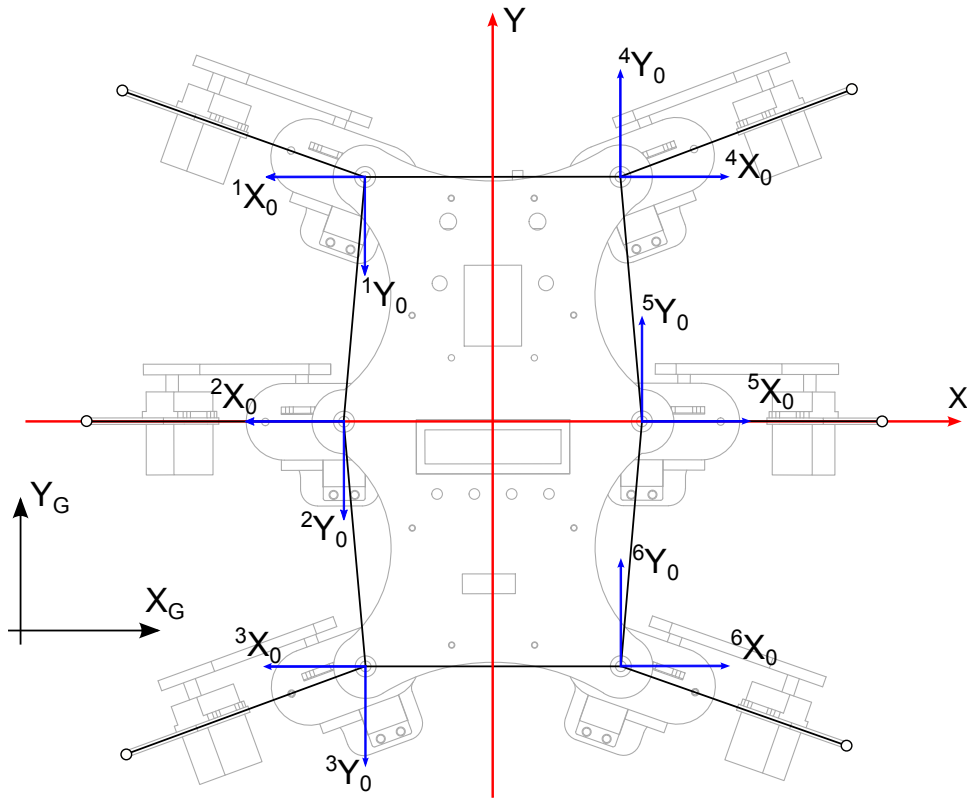
W celu opisu zachowania robota, zostanie związany z nim nadrzędny układ współrzędnych XYZ o początku umieszczonym w środku robota (zobacz rysunek 2.1). Ponadto każdej nodze robota zostaje przypisany lokalny układ współrzędnych ${}^iX_0{}^iY_0{}^iZ_0$, umieszczony w miejscu jej mocowania do korpusu, w sposób pokazany na wspomnianym rysunku. Wówczas transformacja między układem nadrzędnym, a kolejnymi układami lokalnymi ma postać

$${}^iT^0 = Trans(X, {}^ix_0)Trans(Y, {}^iy_0)Rot(Z, {}^i\alpha_0), \quad (2.1)$$

gdzie ${}^ix_0, {}^iy_0$ to położenie miejsca zamocowania i -tej nogi w układzie nadrzędnym, zaś ${}^i\alpha_0 = \pi$ dla nóg 1,2,3, ${}^i\alpha_0 = 0$ dla nóg 4,5,6, co po wyliczeniu daje macierz

$${}^iT^0 = \begin{bmatrix} c & 0 & 0 & {}^ix_0 \\ 0 & c & 0 & {}^iy_0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.2)$$

gdzie $c = -1$ dla nóg 1,2,3, $c = 1$ dla nóg 4,5,6.



Rysunek 2.1 Robot w układzie współrzędnych

2.1.1 Proste zadanie kinematyki nogi robota

Założmy, że noga robota składa się z trzech członów kolejno o długościach l_1, l_2, l_3 , połączonych ze sobą za pomocą przegubów obrotowych w sposób pokazany na rysunku 2.2¹. W celu wyliczenia kinematyki takiej nogi, z każdym jej przegubem oraz końcówką zwiążemy kartezjańskie układy współrzędnych numerując je odpowiednio od 1 do 3.

Kinematyka pojedynczej nogi jest transformacją T_0^3 pozwalającą na bezpośrednie przejście od pierwszego do ostatniego układu współrzędnych. Z pomocą algorytmu Denavita-Hartenberga [14] wyznaczamy poszczególne transformacje przeprowadzające przez kolejne układy współrzędnych:

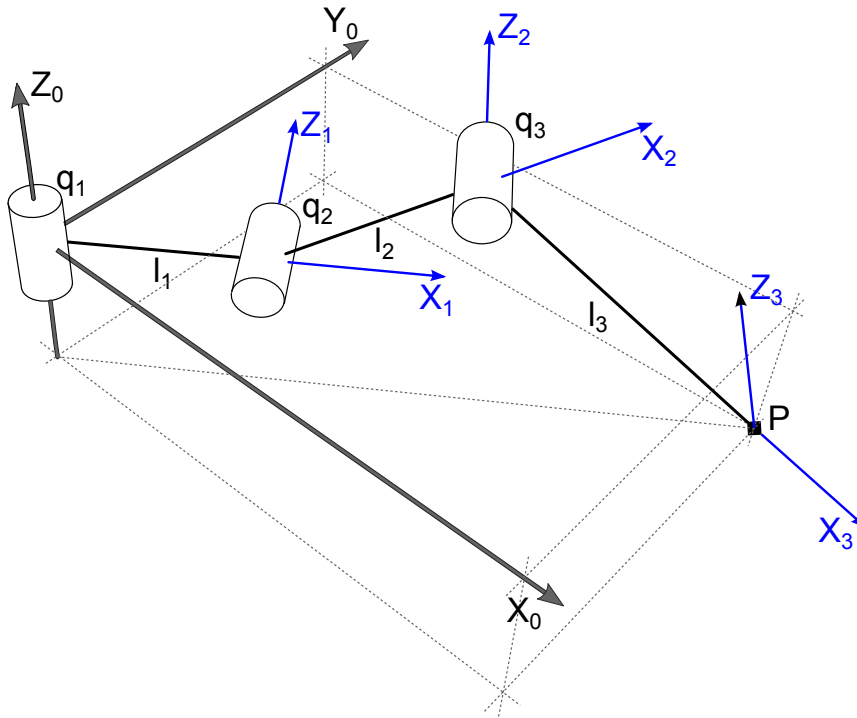
$$\begin{aligned} T_0^1 &= Rot(Z, q_1) Trans(X, l_1) Rot(X, -\frac{\pi}{2}), \\ T_1^2 &= Rot(Z, q_2) Trans(X, l_2), \\ T_2^3 &= Rot(Z, q_3) Trans(X, l_3), \end{aligned} \quad (2.3)$$

a następnie na podstawie (2.3) wyliczamy T_0^3 :

$$T_0^3 = \begin{bmatrix} c_1 c_{23} & -c_1 s_{23} & -s_1 & c_1 (c_{23} l_3 + c_2 l_2 + l_1) \\ s_1 c_{23} & -s_1 s_{23} & c_1 & s_1 (c_{23} l_3 + c_2 l_2 + l_1) \\ -s_{23} & -c_{23} & 0 & -(s_{23} l_3 + s_2 l_2) \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.4)$$

gdzie $c_i = \cos(q_i)$, $s_i = \sin(q_i)$, $c_{ij} = \cos(q_i + q_j)$, $s_{ij} = \sin(q_i + q_j)$. Stąd współrzędne położenia końca nogi – P, wyrażone w układzie współrzędnych związanym z miejscem

¹W tym i następnym podrozdziale dla uproszczenia notacji w symbolach będziemy opuszczali umieszczone z lewej strony u góry indeksy wskazujące, że dotyczą one i -tej nogi.



Rysunek 2.2 Kinematyka pojedynczej nogi robota

zamocowania nogi, mają postać

$$\begin{cases} p_x = c_1 (c_{23}l_3 + c_2l_2 + l_1), \\ p_y = s_1 (c_{23}l_3 + c_2l_2 + l_1), \\ p_z = -(s_{23}l_3 + s_2l_2), \end{cases} \quad (2.5)$$

zaś w układzie nadrzędnym są dane przez

$$\begin{cases} p_x = cc_1 (c_{23}l_3 + c_2l_2 + l_1) + {}^i x_0, \\ p_y = cs_1 (c_{23}l_3 + c_2l_2 + l_1) + {}^i y_0, \\ p_z = -(s_{23}l_3 + s_2l_2). \end{cases} \quad (2.6)$$

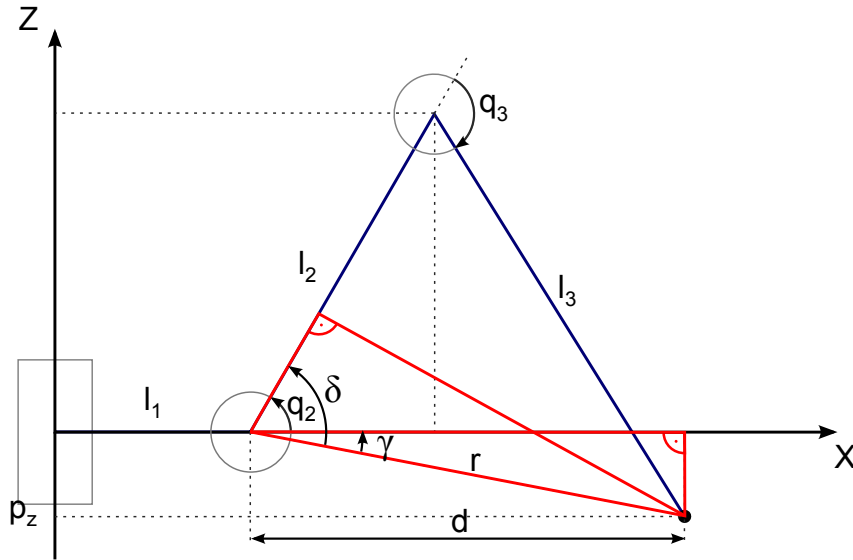
Dla tak wyliczonej kinematyki jacobian analityczny ma postać

$$J_a(q) = \begin{bmatrix} -cs_1 a & -cc_1(s_{23}l_3 + s_2l_2) & -cc_1 s_{23}l_3 \\ cc_1 a & -cs_1(s_{23}l_3 + s_2l_2) & -cs_1 s_{23}l_3 \\ 0 & -c_{23}l_3 - c_2l_2 & -c_{23}l_3 \end{bmatrix}, \quad (2.7)$$

gdzie $a = c_{23}l_3 + c_2l_2 + l_1$ natomiast $b = s_{23}l_3 + s_2l_2$. Jacobian pozwala na wyliczenie prędkości ruchu końca nogi dla danych prędkości ruchu jej przegubów

$$\dot{p} = J_a(q)\dot{q},$$

gdzie $q = (q_1, q_2, q_3)^T$ zaś $p = (p_x, p_y, p_z)^T$.

Rysunek 2.3 Rzut nogi na płaszczyznę OXZ (dla $q_1 = 0$)

2.1.2 Odwrotne zadanie kinematyki nogi robota

W celu obliczenia kinematyki odwrotnej posłużymy się metodą algebraiczną, szczegółowo opisaną w [17]. Wartość kąta pierwszego przegubu – q_1 można obliczyć jako

$$q_1 = \text{atan2}(p_y, p_x). \quad (2.8)$$

W celu uniknięcia kolizji nogi z korpusem przyjmujemy założenie, że $q_1 \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$, co pociąga za sobą ograniczenie wartości p_x do \mathbb{R}^+ .

Zauważmy, że długość całej nogi zrzutowanej na płaszczyznę OXY wynosi $\sqrt{p_x^2 + p_y^2}$. Korzystając z twierdzenia cosinusów łatwo stwierdzić, że dla sytuacji przedstawionej na rysunku 2.3 prawdziwa jest równość

$$l_2^2 + l_3^2 - 2l_2l_3\cos(\pi - q_3) = d^2 + p_z^2,$$

gdzie $d = \sqrt{p_x^2 + p_y^2} - l_1$, co po uproszczeniu daje

$$c_3 = \frac{d^2 + p_z^2 - l_2^2 - l_3^2}{2l_2l_3},$$

$$s_3 = \pm\sqrt{1 - c_3^2}.$$

Ostatecznie

$$q_3 = \text{atan2}(s_3, c_3). \quad (2.9)$$

Znak funkcji s_3 należy dobrać w zależności od postury nogi robota. W wypadku nogi owada przyjmujemy znak dodatni tak, aby kąt q_3 zawsze należał do przedziału $(0, \pi)$.

Kąt q_2 można wyznaczyć jako różnicę odpowiednich kątów dwóch trójkątów prostokątnych o przeciwprostokątnej r , zaznaczonych na rysunku 2.3 kolorem czerwonym,

$$q_2 = \gamma - \delta.$$

Wartości tych kątów mogą zostać wyznaczone w funkcji długości odpowiednich boków wspomnianych trójkątów, co ostatecznie prowadzi do zależności

$$q_2 = \text{atan2}(p_z, d) - \text{atan2}(l_3 s_3, l_2 + l_3 c_3). \quad (2.10)$$

Alternatywnie, ruch przegubów nogi może być wyznaczony za pomocą metody jakobianowej z równania

$$\dot{q} = J_a^{-1}(q)\dot{p},$$

z macierzą $J_a^{-1}(q)$ postaci

$$J_a^{-1}(q) = \begin{bmatrix} -\frac{s_1}{c(l_1+c_2l_2+c_23l_3)} & \frac{c(c_1)}{l_1+c_2l_2+c_23l_3} & 0 \\ \frac{c_1c_23}{cs_3l_2} & \frac{c_23s_1}{cs_3l_2} & \frac{s_23}{s_3l_2} \\ -\frac{c_1(c_2l_2+c_23l_3)}{cs_3l_2l_3} & -\frac{(c_2l_2+c_23l_3)s_1}{cs_3l_2l_3} & -\frac{s_2l_2+s_23l_3}{s_3l_2l_3} \end{bmatrix}. \quad (2.11)$$

2.2 Algorytmy chodu robota sześcionożnego

Już w 1968r. McGhee[11] podał wzór na maksymalną liczbę rodzajów chodu w zależności od liczby nóg postaci

$$N = (2k - 1)!, \quad (2.12)$$

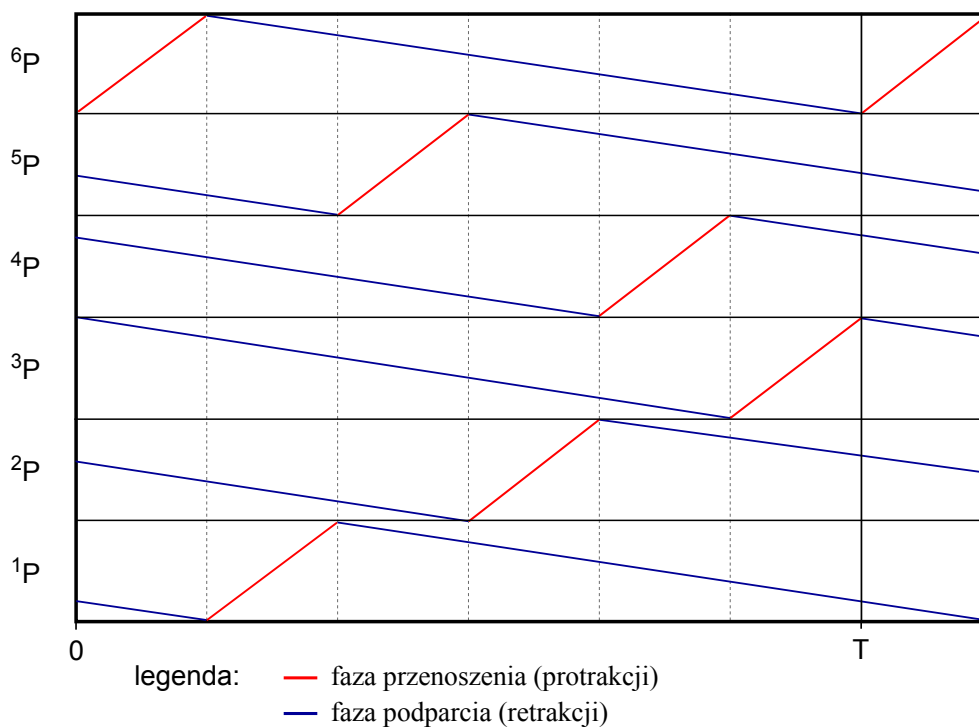
gdzie k - liczba nóg, N - liczba możliwych rodzajów chodu. Łatwo oszacować, że dla robota o sześciu nogach, liczba możliwych chodów jest ogromna ($11!$), na szczęście jednak z naukowego punktu widzenia, znaczenia ma tylko niewielka część tego zbioru. Sama natura przez setki tysięcy lat ewolucji zweryfikowała przydatność większości możliwości pozostawiając w użyciu jedynie kilka chodów, które możemy zaobserwować u zwierząt.

Cechą wspólną dla wszystkich algorytmów chodów jest ich okresowość. Bez względu na wybrany algorytm chodu, każdy krok wykonywany przez robota można podzielić na dwie fazy: fazę podparcia – gdy noga dotyka ziemi i utrzymuje ciężar robota, oraz fazę przenoszenia – gdy noga przenoszona jest w powietrzu w kierunku przemieszczania się robota. Poniżej opisano algorytmy chodu najpopularniejsze dla robotów [13].

2.2.1 Chód pięciopodporowy

Chód pięciopodporowy, zwany też pełzaniem, jest najwolniejszym ale i najstabilniejszym z możliwych algorytmów chodu. Polega on na przenoszeniu nóg kolejno po jednej, przy jednoczesnym powolnym przemieszczaniu tułowia. Obserwując niektóre owady (np. straszky lub stonogi) można zauważyć, że nogi przemieszczane są „falami” od tyłu do przodu ciała. Czas przenoszenia nogi jest relatywnie krótki ($1/6$ okresu chodu), natomiast czas podparcia długi, stąd współczynnik obciążenia, definiowany jako stosunek czasu, w którym noga styka się z podłożem do długości okresu chodu, wynosi $5/6$. Chód ten wybierany jest przez owady podczas przemieszczania się w trudnym terenie, gdy konieczny jest precyzyjny wybór miejsca podparcia każdej z nóg.

Rysunek 2.4 przedstawia położenie każdej z sześciu nóg, w zależności od momentu chodu. Linia czerwona przedstawia fazę przenoszenia nogi, natomiast linia niebieska – fazę podparcia. Kąt nachylenia linii niebieskiej odpowiada prędkości przemieszczania tułowia robota. Ze względu na liczbę nóg, prędkość ta jest skutecznie ograniczona, ponieważ musi być pięciokrotnie mniejsza od prędkości przemieszczenia pojedynczej nogi.



Rysunek 2.4 Diagram chodu pięciopodporowego

2.2.2 Chód mieszany

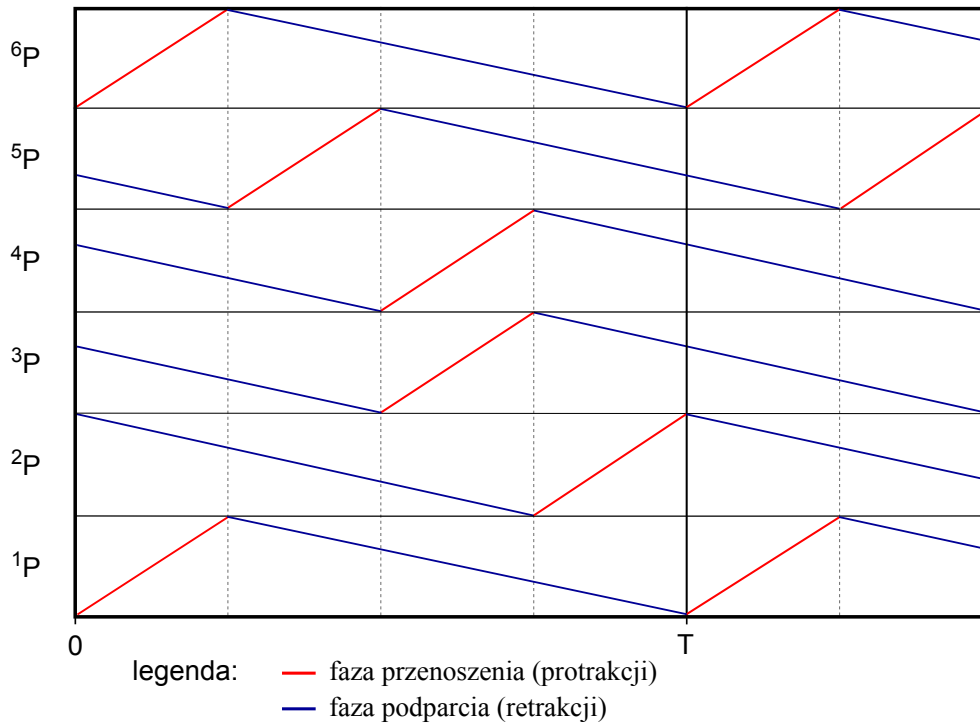
Gdyby zagęścić chód pięciopodporowy, tak aby początki i końce fal przemieszczeń lewej i prawej strony robota nachodziły na siebie, to w wyniku otrzyma się chód mieszany w którym na przemian przenoszone są dwie nogi lub jedna (rysunek 2.5). W ten sposób okres chodu zostaje skrócony względem klasycznego chodu pięciopodporowego do czterokrotności czasu przemieszczenia nogi, co czyni chód mieszany znacznie szybszym, przy nieznacznym zmniejszeniu stabilności. Algorytm chodu mieszanego często można zaobserwować u straszaków przemieszczających się po płaskiej powierzchni.

2.2.3 Chód czteropodporowy

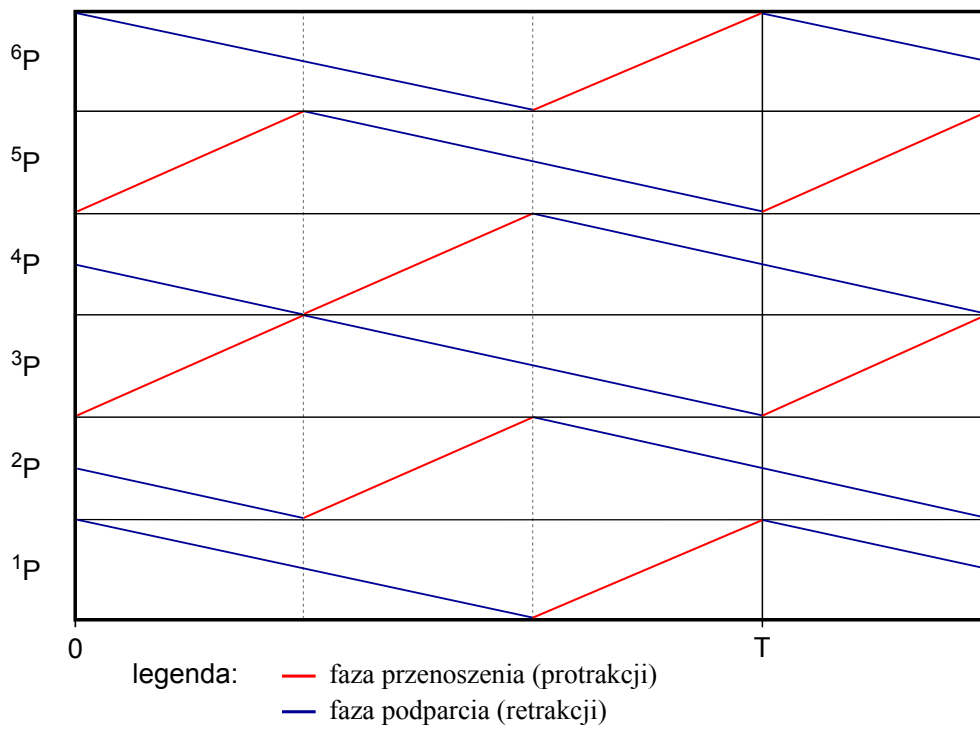
Chód czteropodporowy (rysunek 2.6), inaczej nazywany diagonalnym, jest najczęściej wykorzystywany przez owady podczas przemieszczania się po płaskim terenie. Ze względu na dobrą stabilność jest stosowany także w robotach sześcionożnych. Nazwa diagonalny pochodzi od sposobu przenoszenia nóg – parami dobranymi w taki sposób, żeby przenoszone nogi znajdowały się po dwóch stronach robota ale nie na przeciwko sobie. Tak jak w wypadku chodu pięciopodporowego ruch zaczyna się od tyłu, a przednie nogi przenoszone są na końcu. Chód czteropodporowy jest kompromisem pomiędzy stabilnością a prędkością. Okres chodu wynosi tu trzykrotność czasu przeniesienia pojedynczej nogi, natomiast współczynnik obciążenia $2/3$.

2.2.4 Chód trójpodporowy

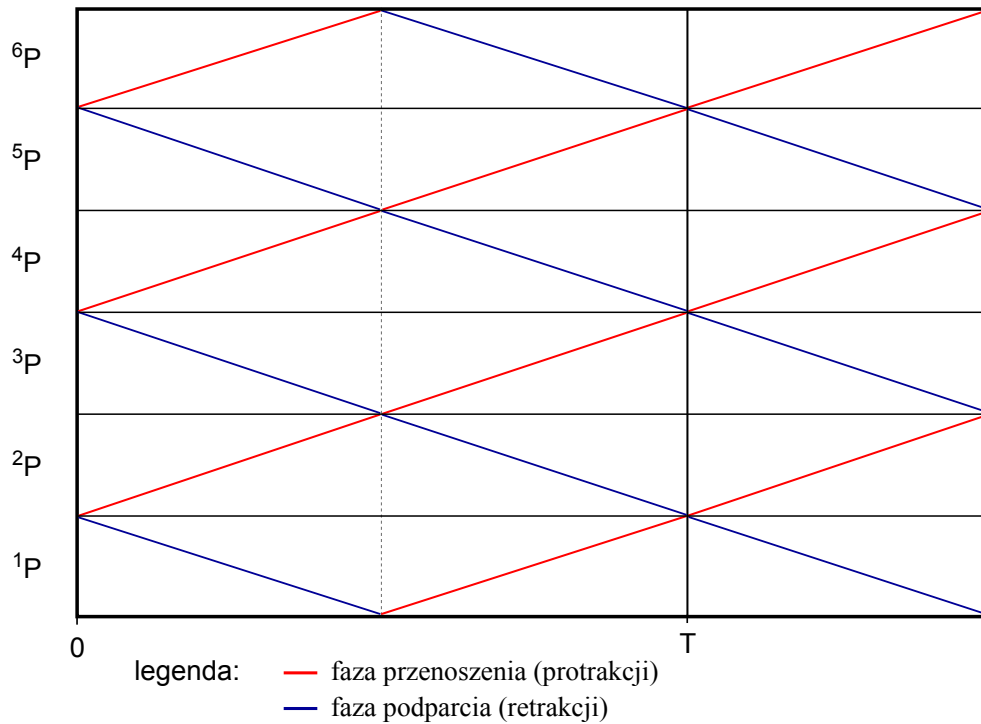
Najszybszym chodem statycznym, jakim może poruszać się sześcionożny robot jest chód podczas którego jednocześnie przestawiane są trzy nogi (rysunek 2.7). Przemiesz-



Rysunek 2.5 Diagram chodu mieszane



Rysunek 2.6 Diagram chodu czteropodporowego



Rysunek 2.7 Diagram chodu trójpodporowego

czenie wszystkich sześciu nóg zajmuje tylko dwie fazy chodu, a nogi przemieszczają się z tą samą prędkością zarówno podczas fazy przenoszenia jak i podparcia. Jako że każda z nóg opiera się o podłoże dokładnie przez połowę okresu współczynnik obciążenia chodu trójpodporowego wynosi $1/2$. Minusem takiego sposobu przemieszczania się jest mniejsza stabilność oraz ryzyko przewrócenia się robota, gdy środek ciężkości wysunięty jest do skrajnego położenia. Podczas chodu trójpodporowego ważne jest, aby punktami podparcia były zawsze nogi zamocowane możliwie najdalej od siebie – nigdy sąsiadujące. Dalsze zmniejszenie współczynnika obciążenia doprowadziłoby do sytuacji, gdy mniej niż trzy nogi opierałyby się jednocześnie o podłoże, a to oznaczałoby już konieczność utrzymywania równowagi. Chód ten jest granicznym przykładem chodu stabilnego.

2.3 Planowanie ruchu

Sposób w jaki przenoszone są nogi robota zależy od wybranego algorytmu chodu oraz od kształtu trajektorii po jakiej porusza się środek robota. Jak wspomniano wcześniej, każdy krok składa się z dwóch faz: podparcia i przenoszenia. Przy założeniu braku poślizgu nóg w punktach kontaktu z podłożem, przebieg trajektorii po których powinny przemieszczać się nogi podczas fazy podparcia jest określony trajektorią ruchu korpusu robota. W fazie przenoszenia noga nie dotyka podłoża, może więc zostać przeniesiona po dowolnej trajektorii zapewniającej jej położenie ponad podłożem, istotne jest jedynie osiągnięcie w zadanym czasie docelowego punktu końcowego, w którym noga ponownie uzyskuje kontakt z podłożem.

Ze względu na to, że celem projektu jest naśladowanie chodu owadów, przyjmujemy stałą orientację korpusu względem podłoża, zmieniać będą się tylko współrzędne x, y, z określające położenia środka robota. W praktyce oznacza to, że korpus położony jest

równoległe do płaszczyzny $X_G O Y_G$ (rysunek 2.1), natomiast ruch odbywa się w kierunku osi Y układu związanego z robotem.

Założmy, że z jest parametrem chodu określającym wysokość niesienia korpusu w trakcie ruchu, zaś funkcje $x(t), y(t)$ kreślą trajektorię ruchu korpusu charakteryzującą się lokalnie krzywizną $K(t)$. Wówczas promień ruchu

$$R(t) = \frac{1}{K(t)},$$

zaś prędkość ruchu

$$v(t) = \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2}.$$

Zwiążmy z chwilowym środkiem obrotu układ współrzędnych $X_S Y_S Z_S$ o stałej orientacji względem układu globalnego (zobacz rysunek 2.8). Wówczas prędkość kątowna robota

$$\omega(t) = \frac{v(t)}{R(t)}.$$

W takiej sytuacji, koniec nieruchomej względem korpusu nogi poruszałby się po krzywej o promieniu

$${}^i R(t) = \sqrt{(R - |x_s|)^2 + y_s^2},$$

z prędkością

$${}^i v(t) = {}^i R \omega(t).$$

By w trakcie ruchu robota koniec nogi pozostał nieruchomy względem podłoża w fazie podparcia, należy poruszać nogą tak, by osiągnął on względem korpusu prędkość

$${}^i v_w(t) = v(t)^2 + {}^i v(t)^2 + 2|v(t)||{}^i v(t)|\cos(\psi),$$

gdzie $\cos(\psi) = \frac{R - |x_s|}{R}$ (rysunek 2.8). Tak wyliczona prędkość końca nogi pozwala na wyznaczenie prędkości ruchu przegubów zapewniających nieprzemieszczanie się punktów kontaktu z podłożem w trakcie ruchu. Prędkość można obliczyć zarówno z wykorzystaniem podejścia algebraicznego, jaki i za pomocą metody jacobianowej.

Do zaplanowania pozostaje ruch nogi w fazie przenoszenia. Średnia prędkość ${}^i \bar{v}$ ruchu i -tej nogi w pojedynczym kroku jest równa

$${}^i \bar{v}(t) = \frac{{}^i D(t)}{T},$$

gdzie T – okres chodu, ${}^i D$ – długość kroku i -tej nogi robota. Równocześnie żądamy, by w pojedynczym kroku

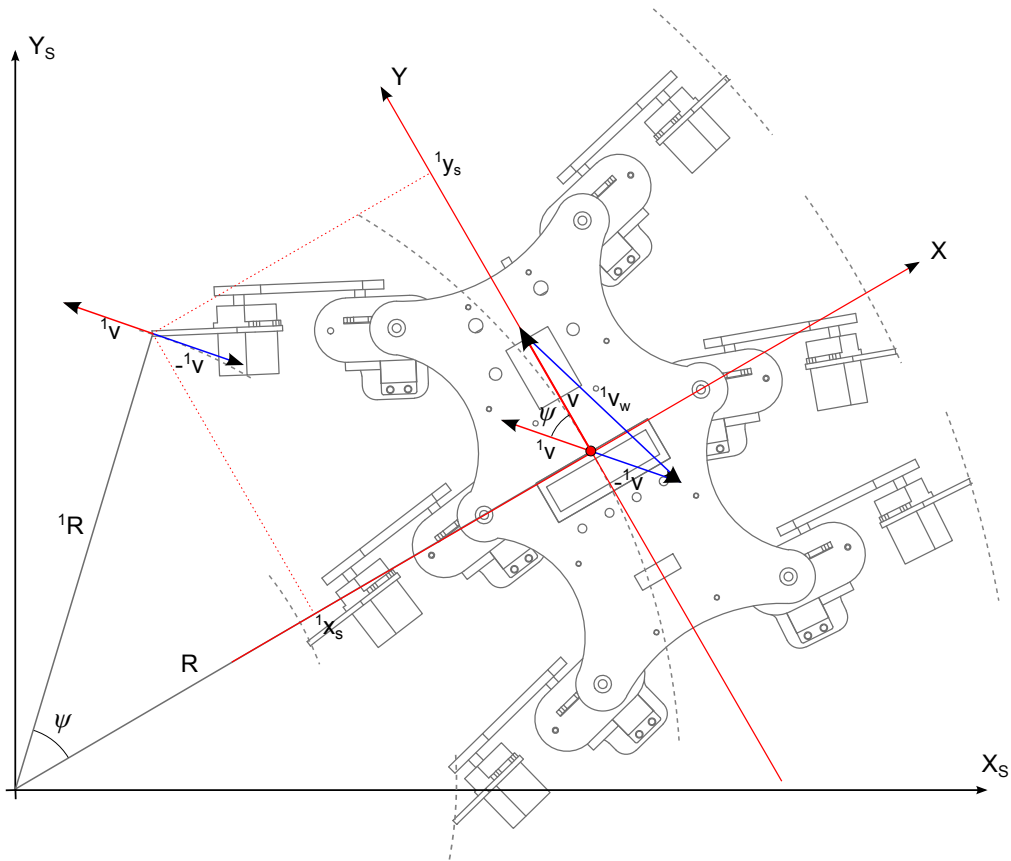
$${}^i \bar{v}(t) = \frac{1}{T} \int_t^{t+T} {}^i v(t) dt,$$

stąd

$${}^i D(t) = \int_t^{t+T} v(t) dt.$$

Ponieważ, w fazie retrakcji (gdy noga dotyka ziemi) ${}^i v(t) = 0$, stąd w fazie protrakcji średnia prędkość musi wynosić

$${}^i \bar{v}_p(t) = \frac{{}^i D(t)}{t_p}, \quad (2.13)$$



Rysunek 2.8 Robot podczas chodu po łuku

gdzie t_p to czas trwania fazy protrakcji. Równość (2.13) określa jedynie kształt trajektorii składowych $x(t), y(t)$, zaś składowa $z(t)$ może przyjmować dowolne wartości pod warunkiem, że na początku oraz na końcu fazy protrakcji (oraz przez cały czas trwania fazy retrakcji) $z(t) = z$, gdzie z jest wprowadzonym wcześniej parametrem chodu. Trajektorja $z(t)$ w trakcie fazy protrakcji może być zadawana na wiele sposobów, które najwygodniej jest określać nazwami figur geometrycznych, powstających na wykresie $z(t)$. Najczęściej wykres taki przypomina trapez, trójkąt lub półkoło. W opracowywanym rozwiązaniu wybrano trajektorję w kształcie półkoła (zadawaną przez funkcję sinus) ze względu na brak ostrych krawędzi, które w praktyce mogłyby powodować szarpnięcia, a co za tym idzie negatywnie wpływać na płynność chodu.

Rozdział 3

Konstrukcja robota

Celem pracy było skonstruowanie robota umożliwiającego demonstrację dowolnego, statycznego chodu sześcionożnego. Dodatkowo położono nacisk na to, aby chód był możliwie płynny oraz stabilny. Aby osiągnąć założony cel, potrzebna była konstrukcja lekka oraz sztywna, ważna była także jej odpowiednia wytrzymałość umożliwiająca wykorzystanie robota jako stanowisko laboratoryjne. Opracowana konstrukcja pozwala na ruch w dowolnym kierunku, jednak aktualna wersja oprogramowania pozwala na przemieszczanie się po trajektoriach o prędkości stycznej do wzdłużnej osi symetrii robota. Parametrami ruchu są jego chwilowa prędkość, chwilowy promień ruchu oraz wysokość niesienia korpusu. Najważniejsze parametry robota:

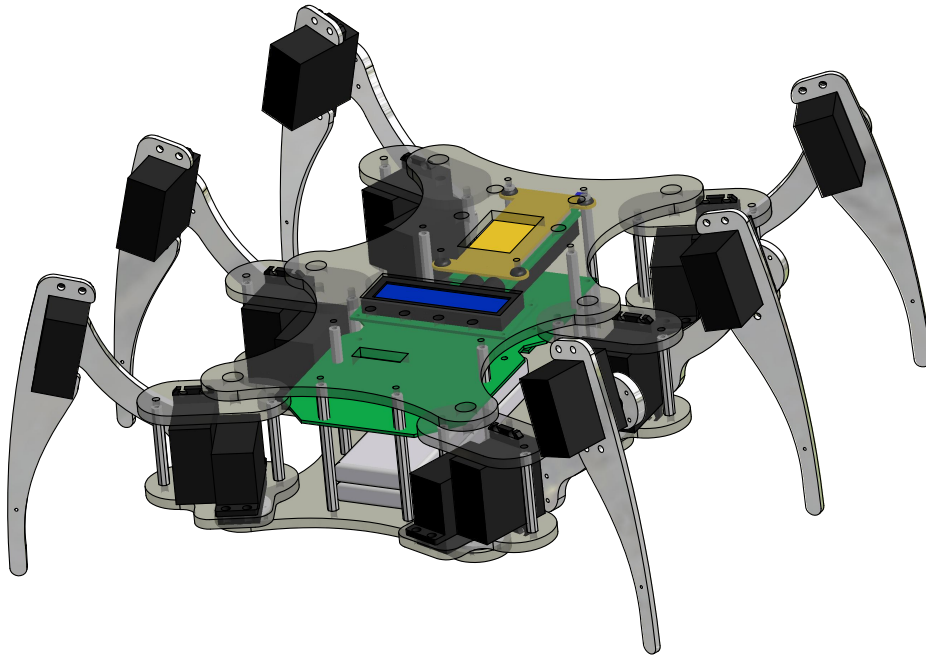
- Jednostka sterująca - mikroprocesor AVR ATmega128
- 18 niezależnych serwomechanizmów
- Dwa ogniwa Li-Po 3.7V 4400mAh
- BEC¹ o wydajności do 20A zasilający serwomechanizmy
- Masa robota to 2.6kg
- Wymiary z nogami w pozycji spoczynkowej 360 × 360mm

3.1 Konstrukcja mechaniczna i napędy

Projekt robota (rysunek 3.1) został opracowany w środowisku Autodesk Inventor [1]. Konstrukcja bazuje na dwóch równoległych płytach (spodniej oraz wierzchniej) połączonych ze sobą za pomocą ośmiu metalowych dystansów o długości 65mm. Nogi rozmieszczone na planie elipsy, po trzy na każdą stronę, mocując każdą do płyty spodniej (orczyki serwomechanizmów przykręcone śrubami) oraz do płyty wierzchniej (łożyska o średnicy zewnętrznej 11mm z kołnierzami o średnicy 13mm). Szczegóły rozmieszczenia nóg przedstawiono na rysunku 3.2. Robot jest podwójnie symetryczny – względem swojej osi podłużnej oraz poprzecznej. Uzyskany maksymalny zasięg wyprostowanej nogi to około 220mm od miejsca jej mocowania, natomiast maksymalna wysokość na jaką może zostać podniesiony tułów robota to 145mm. Budowa pojedynczej nogi została pokazana na rysunku 3.3.

Pierwotnie cała konstrukcja robota została zbudowana z przezroczystego pleksi o grubości 5mm. Największą zaletą tego materiału jest bardzo niewielka masa (gęstość 1.18g/cm³)

¹BEC (ang. Battery Eliminator Circuit) jest to układ składający się z równolegle ze sobą połączonych stabilizatorów. Służy zapewnieniu odpowiedniego napięcia dla serwomechanizmów.



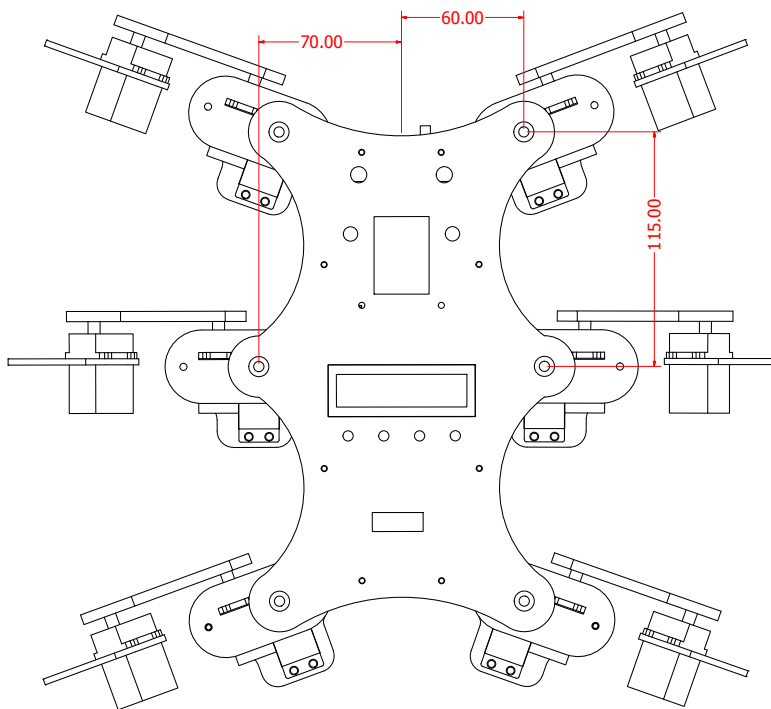
Rysunek 3.1 Projekt robota

przy stosunkowo dużej wytrzymałości. Tworzywo to sprawdziło się doskonale w wypadku płyty spodniej oraz płyty wierzchniej, tworzących konstrukcję nośną robota, jednak zupełnie nie zdało egzaminu w wypadku jego odnóży, które w trakcie chodu ulegały dużym odkształceniom. W ostatecznej wersji robota wszystkie uda oraz golenie zostały wykonane z aluminium o grubości 3mm , zabieg ten pozwolił znacznie zwiększyć stabilność chodu.

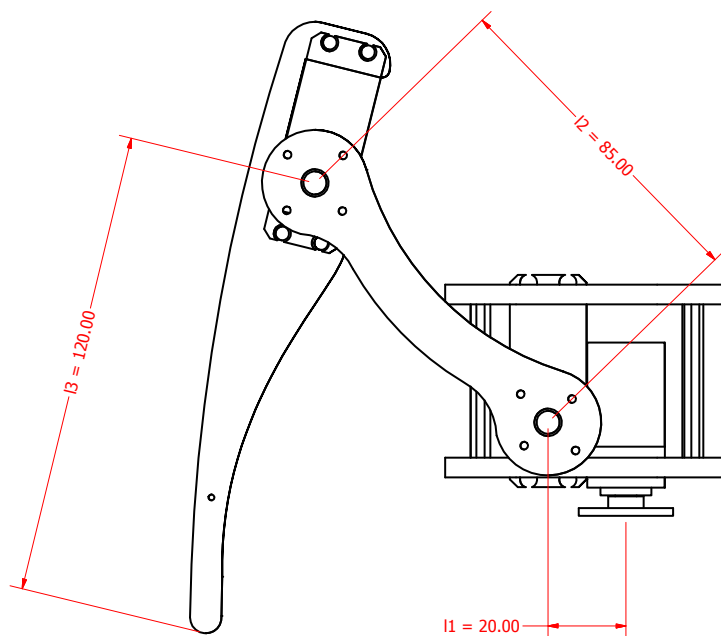
Robot wyposażony jest w 18 niezależnych napędów – po 3 na każdą nogę. Ze względu na łatwą dostępność oraz niską wagę w roli napędów zastosowano serwomechanizmy modelarskie. Przeguby ${}^i q_1$ wyposażono w serwa analogowe HS-645MG firmy Hitec [6], przeguby ${}^i q_2$ wyposażono w serwa cyfrowe HS-5625MG firmy Hitec [5], natomiast przeguby ${}^i q_3$ napędzane są serwami cyfrowymi ADS-940HMG firmy Alturn [2]. Zastosowanie napędów cyfrowych dla osi poziomych nóg robota motywowane było ich znacznie większą siłą trzymania w porównaniu do podobnych napędów analogowych. Jak widać na rysunku 3.3, na osiach pionowych spoczywa masa całego robota, podczas gdy osie poziome realizują jedynie zmianę orientacji nóg.

3.2 Układy elektroniczne

Elektronika robota została podzielona na moduły, z których każdy posiada osobną funkcję. Sercem całego układu jest mikroprocesor umieszczony na płycie głównej. Jego zadaniem jest odbiór sygnału z aparatury zdalnego sterowania oraz wysyłanie za pomocą protokołu UART komend do serwokontrolera, który przetwarza te komendy na sygnał PWM wysyłany bezpośrednio do wszystkich osiemnastu serwomechanizmów. Całość układu zasilana jest przez akumulatory litowo-polimerowe za pośrednictwem dedykowanego układu zasilania składającego się ze stabilizatorów oraz układu BEC.



Rysunek 3.2 Podstawowe wymiary robota – widok z góry

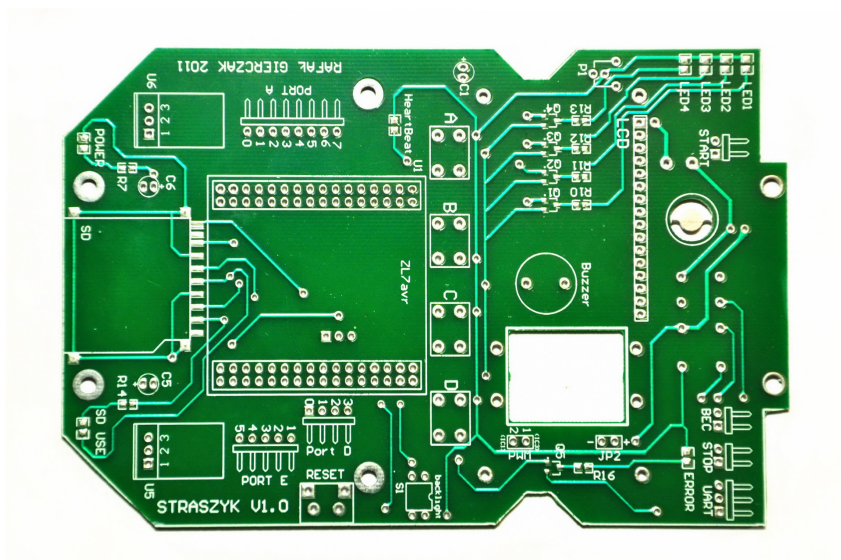


Rysunek 3.3 Podstawowe wymiary nogi robota

3.2.1 Płyta główna

Płyta główna robota (rysunek 3.4) została wykonana w technologii dwuwarstwowej. W jej centralnej części znajduje się port pozwalający na wpięcie modułu z mikrokontrolerem, co pozwala na łatwy serwis robota, a także na zastąpienie go modułem z innym procesorem, jeśli zaistnieje taka potrzeba. Płyta została zaprojektowana z myślą o module ZL7AVR [8], wyposażonym w procesor AVR ATmega128. Poza jednostką sterującą, na płycie znajdują się następujące elementy:

- ekran LCD 2 × 16 znaków,
- 3 porty uniwersalne,
- 4-przyciskowa klawiatura do obsługi menu,
- przycisk RESET,
- 7 diod sygnalizacyjnych,
- sygnalizator akustyczny,
- gniazdo kart SD,
- port UART,
- port kontroli zasilania serwomechanizmów,
- port wejściowy dla sygnałów PWM.



Rysunek 3.4 Płyta główna przed montażem elementów

3.2.2 Serwokontroler

Dla ułatwienia sterowania osiemnastoma serwomechanizmami, robot został wyposażony w serwokontroler SSC-32 firmy Lynxmotion [10]. Komunikacja z serwokontrolerem przebiega za pomocą protokołu UART z prędkością 115 200 *bps*. Urządzenie zbudowane jest w oparciu o procesor AVR ATmega168 i pozwala na jednoczesne obsłużenie do 32 serwomechanizmów. Dzięki szerokiej wachlarzowi instrukcji możliwe jest zadanie nie tylko żądanej pozycji lecz także czasu lub prędkości z jaką ma ona zostać osiągnięta.

3.2.3 Zasilanie

Ponieważ większość napędów, w które wyposażony jest robot, to serwomechanizmy cyfrowe, charakteryzujące się względnie wysokim poborem prądu (nawet do $1200mA$ przy pełnym obciążeniu [2]), robot zasilany jest przez dwa szeregowo połączone ogniwa litowo-polimerowe firmy 3E Model, o pojemności $4400mAh$. Te same akumulatory zasilają także wszystkie układy elektroniczne robota wraz z odbiornikiem zdalnego sterowania.

Ze względu na różnicę napięć znamionowych akumulatorów ($7.2V$) oraz zasilania serwomechanizmów ($6V$), w robocie konieczne było zastosowanie obwodu stabilizacji napięcia, którego rolę pełni układ MAX BEC 2 firmy Jeti [7]. Jego podstawowe cechy to:

- wydajność do $20A$ prądu ciągłego,
- sygnalizacja diodowa poziomu naładowania akumulatorów,
- automatyczne odcięcie zasilania, gdy napięcie akumulatorów spadnie poniżej $6.2V$,
- cztery poziomy napięcia wyjściowego ($6.0V$, $5.7V$, $5.4V$, $5.0V$),
- niska masa – $85g$,
- możliwość elektronicznego włączania i wyłączenia zasilania (z poziomu płyty głównej).

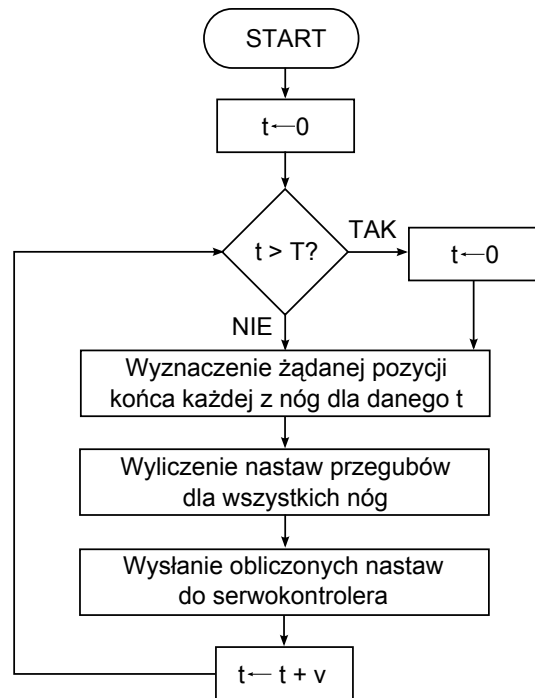
Ze względu na planowane użycie robota jako stanowiska laboratoryjnego, konieczne było wyposażenie go w łatwo dostępny wyłącznik umożliwiający natychmiastowe odłączenie zasilania – w tym celu zastosowano układ przekaźnika z podtrzymaniem. Na wierzchniej obudowie robota znajdują się dwa przyciski – zielony „START”, którego naciśnięcie powoduje zasilenie cewki przekaźnika, którego styki od tego momentu same zasilają cewkę. Naciśnięcie czerwonego przycisku „STOP” powoduje natychmiastowe odłączenie cewki, a co za tym idzie także zasilania elektroniki robota. Zastosowano przekaźnik typu DPDT (o podwójnym komplecie styków), dzięki czemu możliwe jest równoczesne wyłączenie wspomnianego powyżej układu stabilizacji napięcia dla serwomechanizmów. Szczegółowy schemat zastosowanego rozwiązania znajduje się w dodatku B.2.

Zasilanie elementów elektronicznych zlokalizowanych na płycie głównej zrealizowano przy pomocy dwóch stabilizatorów typu Low-dropout. Stabilizowane napięcia to odpowiednio $3,3V$ dla obsługi kart SD oraz $5V$ dla mikroprocesora oraz wszystkich pozostałych podzespołów płyty głównej.

3.3 Oprogramowanie

3.3.1 Algorytm generowania chodu

Aby umożliwić robotowi chód, konieczne jest obliczanie pozycji końcówki każdej z nóg w zależności od aktualnego etapu chodu. Obliczenia muszą być proste, gdyż w przeciwnym wypadku ich złożoność znacznie zmniejszy częstotliwość wyznaczania położeń odnóży, co pogorszy płynność i gładkość chodu. W niniejszej pracy nie zajmujemy się zagadnieniem chodu po nierównej powierzchni, dlatego zakładamy, że powierzchnia po której porusza się robot jest idealnie płaska, a korpus robota jest zawsze niesiony równolegle do niej. Dodatkowo, dla uproszczenia obliczeń, przyjmujemy że podczas jednego okresu chodu każda noga wykonuje dokładnie jeden krok. Dane wejściowe algorytmu generowania chodu są następujące:



Rysunek 3.5 Diagram pętli głównej programu robota

T – okres chodu (wyrażony w umownych jednostkach, czas przestawienia wszystkich nóg),

t – aktualna chwila (w przedziale od 0 do T),

R – promień łuku po którym porusza się robot (liczony od środka robota),

$krok_start[1..6]$ – chwila rozpoczęcia kroku dla poszczególnych nóg,

$krok_koniec[1..6]$ – chwila zakończenia kroku dla poszczególnych nóg,

L/Φ – długość kroku (dla marszu na wprost) lub całkowity kąt o jaki powinna przemieścić się noga (podczas marszu po łuku).

Na podstawie powyższych parametrów, możliwe jest wyliczenie chwilowej pozycji końcówki każdej z nóg w nadrzędnym układzie współrzędnych².

Wyznaczenie pozycji końca nogi podczas chodu po łuku

Wyznaczenie położenia końca nogi jest dość złożonym problemem obliczeniowym, szczególnie kiedy robot porusza się po łuku. Aby chód przebiegał bez poślizgów, każda z nóg robota musi poruszać się po łuku o odpowiednim promieniu oraz długości. Najprostszym (z teoretycznego punktu widzenia) rozwiązaniem jest przeliczenie spoczynkowego położenia stopy robota na współrzędne biegunowe, obliczenie promienia łuku, po którym porusza się dana noga, wyznaczenie przemieszczenia o odpowiedni kąt oraz przeliczenie współrzędnych z powrotem do układu kartezjańskiego. Położeniem spoczynkowym nazywamy miejsce, w którym znajduje się koniec nogi, podczas gdy robot nie jest w trakcie chodu, natomiast gdy robot idzie – położenie spoczynkowe wyznacza punkt

²Alternatywnie algorytm chodu można sparametryzować przydzielając dla każdego kroku zmienną określającą długość trwania fazy protrakcji oraz zmienną określającą fazę, w której znajduje się dany krok w momencie gdy $t = 0$

środkowy każdego kroku. Metody podane w rozdziale 2, pozwalają na zrozumienie problemu oraz przedstawienie go w postaci prostego algorytmu w dziedzinie współrzędnych kartezjańskich, którego szczegóły zaprezentowano poniżej.

Pierwszym krokiem jest wyznaczenie kąta, o który i -ta noga powinna oddalić się od położenia spoczynkowego, aby przyjąć pozycję odpowiednią dla danej chwili chodu. Kąt ten nazwiemy ${}^i\beta$. Sposób wyliczenia ${}^i\beta$ zależy od aktualnej fazy kroku. Dla fazy przenoszenia (protrakcji) wartość tego kąta wynosi

$${}^i\beta = \Phi \left(\frac{t - \text{krok_start}[i]}{\text{krok_koniec}[i] - \text{krok_start}[i]} - 0.5 \right),$$

natomiast dla fazy podparcia (retrakcji), wzór zależy jeszcze od tego, czy w danym okresie chodu, noga jest jeszcze przed fazą protrakcji

$${}^i\beta = \Phi \left(\frac{T + t - \text{krok_koniec}[i]}{T - (\text{krok_koniec}[i] - \text{krok_start}[i])} - 0.5 \right),$$

czy po fazie protrakcji

$${}^i\beta = \Phi \left(\frac{t - \text{krok_koniec}[i]}{T - (\text{krok_koniec}[i] - \text{krok_start}[i])} - 0.5 \right).$$

Odjęcie wartości 0.5 od ułamka określającego aktualny etap kroku zapewnia, że ${}^i\beta$ będzie miało wartość ujemną przez pierwszą połowę kroku, a więc pozwoli, aby noga osiągała swoje położenie spoczynkowe dokładnie w środku kroku.

Następnie, wychodząc od wzorów na sinus oraz cosinus sumy kątów, wyznaczamy wzory na współrzędne kartezjańskie końcówki i -tej nogi robota przemieszczonej o kąt ${}^i\beta$

$$\begin{cases} \sin(\alpha + {}^i\beta) = {}^i s \cdot \cos({}^i\beta) + {}^i c \cdot \sin({}^i\beta), \\ \cos(\alpha + {}^i\beta) = {}^i c \cdot \cos({}^i\beta) - {}^i s \cdot \sin({}^i\beta), \end{cases} \quad (3.1)$$

gdzie ${}^i c = \cos({}^i\alpha)$ i ${}^i s = \sin({}^i\alpha)$ są parametrami robota zależnymi od wybranej pozycji spoczynkowej i -tej nogi.

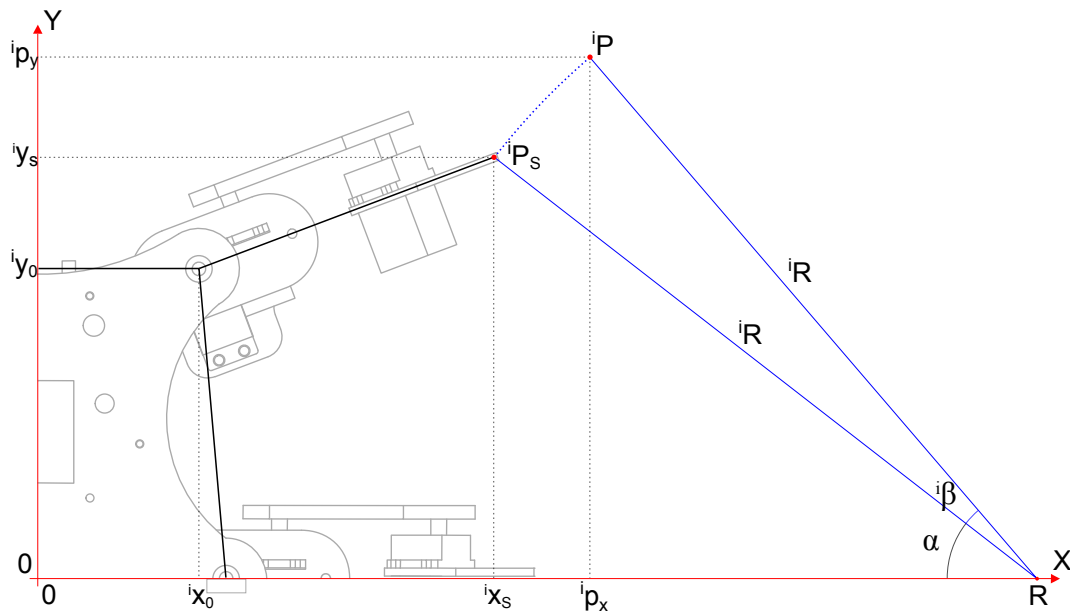
Wyrażając zależne od kątów ${}^i\alpha$ funkcje trygonometryczne jako stosunki długości odpowiednich odcinków (patrz rysunek 3.6) otrzymujemy

$$\begin{cases} \frac{{}^i p_y}{{}^i R} = \frac{{}^i y_s}{{}^i R} \cos({}^i\beta) + \frac{R - {}^i x_s}{{}^i R} \sin({}^i\beta), \\ \frac{R - {}^i p_x}{{}^i R} = \frac{R - {}^i x_s}{{}^i R} \cos({}^i\beta) - \frac{{}^i y_s}{{}^i R} \sin({}^i\beta). \end{cases}$$

Po obustronnym pomnożeniu powyższych równań przez ${}^i R$ oraz odpowiednich przekształceniach otrzymujemy zależność na położenie końca i -tej nogi w funkcji kąta ${}^i\beta$

$$\begin{cases} {}^i p_y = {}^i y_s \cos({}^i\beta) + (R - {}^i x_s) \sin({}^i\beta), \\ {}^i p_x = {}^i y_s \sin({}^i\beta) - (R - {}^i x_s) \cos({}^i\beta) + R. \end{cases} \quad (3.2)$$

Warto zwrócić uwagę na brak konieczności wyznaczania promienia krzywej ruchu poszczególnych nóg ${}^i R$ oraz tylko jednokrotne użycie funkcji *sinus* oraz *cosinus* dla każdej z nóg.



Rysunek 3.6 Metoda wyznaczenia położenia nogi na łuku

Wyznaczenie pozycji końca nogi podczas chodu po prostej

Chód po prostej jest szczególnym przypadkiem chodu po łuku, którego promień ma nieskończoną długość, a więc wartość kąta ${}^i\beta$ dąży do 0. Przyjmując, że Δ określa jak daleko koniec nogi przesunął się od położenia spoczynkowego, zauważmy, że

$$\sin\left(\frac{{}^i\beta}{2}\right) = \frac{{}^i\Delta}{2{}^iR}.$$

Dla bardzo małych kątów $\sin(x) = x$ oraz $\cos(x) = 1$, stąd

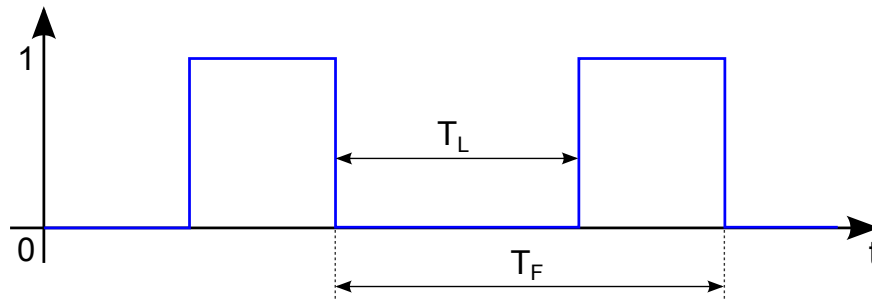
$${}^i\beta = \frac{{}^i\Delta}{{}^iR}.$$

Przy ruchu po prostej dla $K \rightarrow \infty$ z równania (3.2) otrzymujemy

$$\begin{cases} {}^i p_y = {}^i y_s + {}^i \Delta, \\ {}^i p_x = {}^i x_s. \end{cases} \quad (3.3)$$

Wyliczenie nastaw przegubów nogi

Znając żądane położenia końca nogi, przystępujemy do wyliczenia nastaw jej przegubów. Metoda obliczania kinematyki odwrotnej została opisana już wcześniej w rozdziale 2.1.2, dlatego tutaj przytoczymy jedynie szczegóły związane z implementacją. Zgodnie z dokumentacją zastosowanych serwomechanizmów [5], każde $10\mu s$ szerokości impulsu sygnału PWM odpowiada obrotowi osi napędu o 1 stopień. Informacja ta pozwala na bardzo proste przekształcenie wyliczonych nastaw przegubów na wypełnienie sygnału sterującego. Obliczenie nastaw pociąga za sobą konieczność wielokrotnego wyznaczania wartości funkcji trygonometrycznych, dlatego niezbędne okazało się poszukiwanie metod przyspieszenia działania algorytmu. Ze względu na duży zapas pamięci flash, zastosowano tablicowanie wartości funkcji trygonometrycznych z dokładnością do trzech miejsc po przecinku. Metoda ta pozwoliła w ostatecznym programie na wyznaczanie nastaw wszystkich przegubów z częstotliwością $30Hz$, co przełożyło się na satysfakcjonującą płynność ruchów robota.



Rysunek 3.7 Metoda pomiaru wypełnienia sygnału PWM

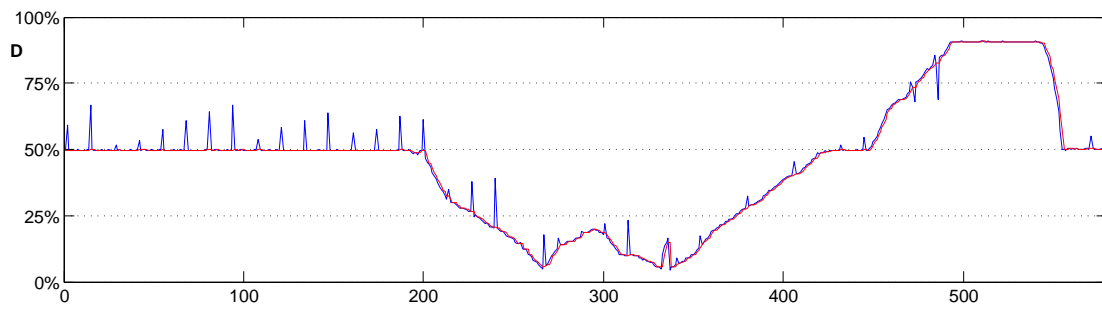
3.3.2 Odbiór sygnału z aparatury zdalnego sterowania

Podczas praktycznego wykorzystania robota, konieczna jest możliwość zadawania chwilowej prędkości oraz wartości promienia skrętu, najlepiej jeśli czynność tą będzie można wykonywać zdalnie. W tym celu zdecydowano się zastosować typową aparaturę modelarską. Każda standardowa aparatura służąca do zdalnego sterowania modeli jest przystosowana do przesyłania kilku sygnałów PWM o częstotliwości 50Hz . Wartość parametru, takiego jak na przykład promień skrętu robota, przekazywana jest w postaci wypełnienia sygnału. Aby odczytać wypełnienie zastosowano mechanizm *Input Capture*, występujący jako integralna część timera praktycznie w każdym mikroprocesorze firmy Atmel. Mechanizm ten umożliwia automatyczne wywołanie przerwania w razie wystąpienia malejącego lub rosnącego zbocza w badanym sygnale, co pozwala na pomiar czasu (np. taktów licznika) dzielącego te wydarzenia. Rysunek 3.7 przedstawia przykładowe odcinki sygnału PWM, których pomiar umożliwi określenie wypełnienia. Wartość wypełnienia można obliczyć za pomocą wzoru

$$D = \frac{T_F - T_L}{T_F} \cdot 100\%.$$

Mikroprocesor użyty w opisywanym robocie, wyposażony jest w dwa niezależne timery (TCNT1, TCNT3) z funkcją *Input Capture*, które wykorzystano do pomiaru sygnałów określających odpowiednio promień łuku R po którym porusza się robot oraz jego prędkość v . Bardziej szczegółowe informacje na temat opisaney metody znajdują się w dokumentacji udostępnionej przez producenta użytego mikroprocesora [3].

W trakcie przetwarzania sygnału zdalnego sterowania napotkano na problem znacznych zakłóceń negatywnie wpływających na jakość ruchu robota. Krótkotrwałe skoki wartości sygnału są charakterystyczne dla tańszych analogowych aparatów zdalnego sterowania i pozostają niezauważalne gdy do odbiornika bezpośrednio podłączymy serwomechanizm. Jednak w wypadku opisywanej konstrukcji skok sygnału bezpośrednio przeliczanego na promień łuku, po którym porusza się robot, powodował gwałtowne szarpnięcia wszystkimi nogami. Problem został rozwiązany przy pomocy filtra medianowego [18]. Eksperymenty wykazały, iż wystarczy poddać filtracji 3 ostatnie wartości sygnału aby całkowicie wyeliminować zakłócenia. Przykładowy efekt filtracji przedstawia wykres na rysunku 3.8, linia niebieska prezentuje przebieg sygnału przed filtracją, natomiast linia czerwona – po filtracji.



Rysunek 3.8 Przebieg sygnału PWM przed filtracją oraz po filtracji

Rozdział 4

Testy funkcjonalne

W celu zbadania funkcjonalności konstrukcji, robot został poddany serii testów. Wykonano próbę chodu do przodu oraz do tyłu, zarówno po prostej jak i po łuku, w lewo oraz w prawo, przyjmując za promień różne stałe wartości włącznie z wartością 0. Następnie wykonano test chodu po łuku o zmiennym promieniu zadawanym przez program, oraz zadawanym przez operatora za pomocą aparatury zdalnego sterowania. W każdym z przypadków robot zadziałał właściwie. Robot poruszał się stabilnie oraz gładko, a nogi w fazie podparcia nie przemieszczały się względem podłoża.

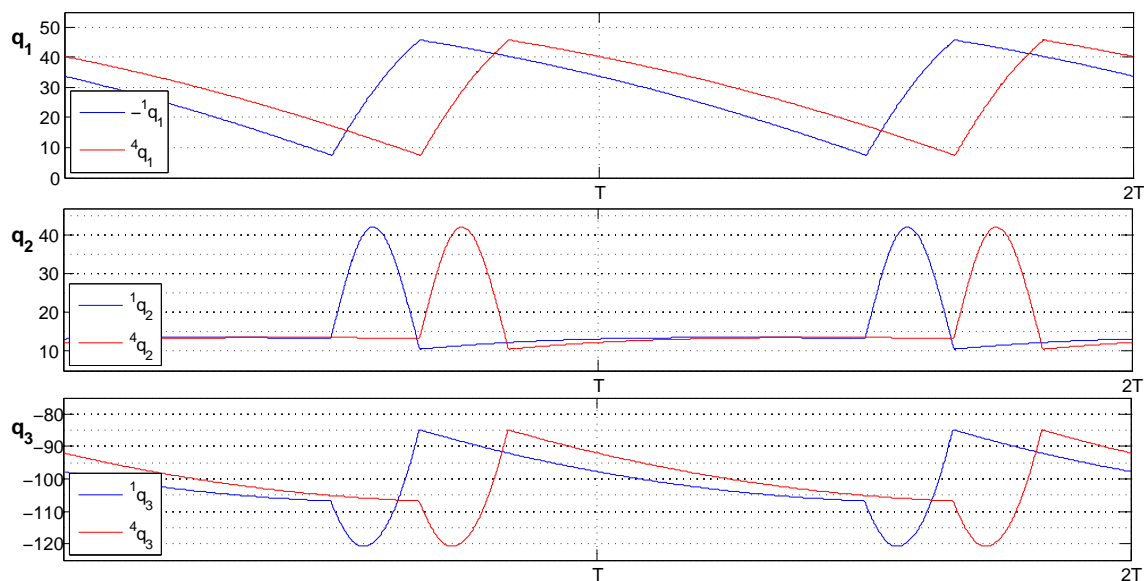
Dla ilustracji jakości ruchu robota poniżej zebrano wykresy przemieszczeń przegubów nóg dla wybranych przypadków.

Na rysunku 4.1 pokazano ruch przegubów nóg pierwszej i czwartej przy chodzie na wprost ze stałą prędkością. Wyraźnie widać, że nogi wykonują dokładnie ten sam ruch przesunięty w czasie.

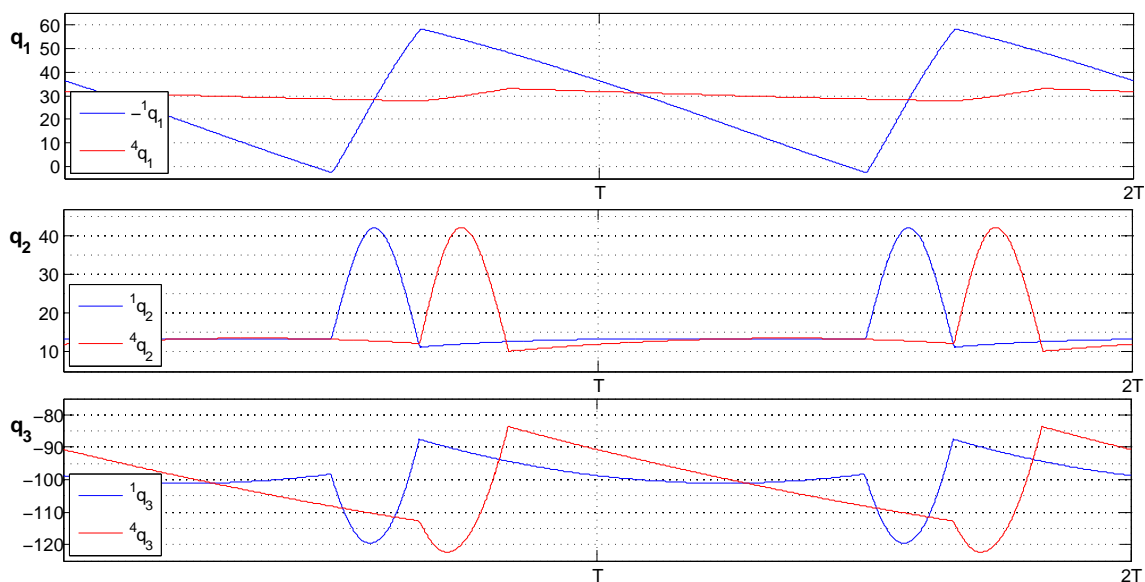
Rysunek 4.2 przedstawia ruch przegubów nóg 1P oraz 4P podczas chodu pięciopodporowego po łuku o promieniu $300mm$. Śledząc przebiegi nastaw 1q_1 oraz 4q_1 łatwo zaobserwować, że krok wykonywany przez nogę 1P , a więc znajdującą się po zewnętrznej stronie zakrętu, wyraźnie się wydłużył, natomiast długość kroku nogi 4P uległa znacznej redukcji.

Na rysunku 4.3 pokazano wykresy ruchu przegubów w trakcie chodu pięciopodporowego po łuku o promieniu $5mm$, będącym praktycznie obrotem wokół własnej osi. Jak widać, zamontowane na przeciwko siebie nogi wykonują w tym przypadku praktycznie ten sam ruch z tą tylko różnicą, że noga 4P wykonuje go „do tyłu”.

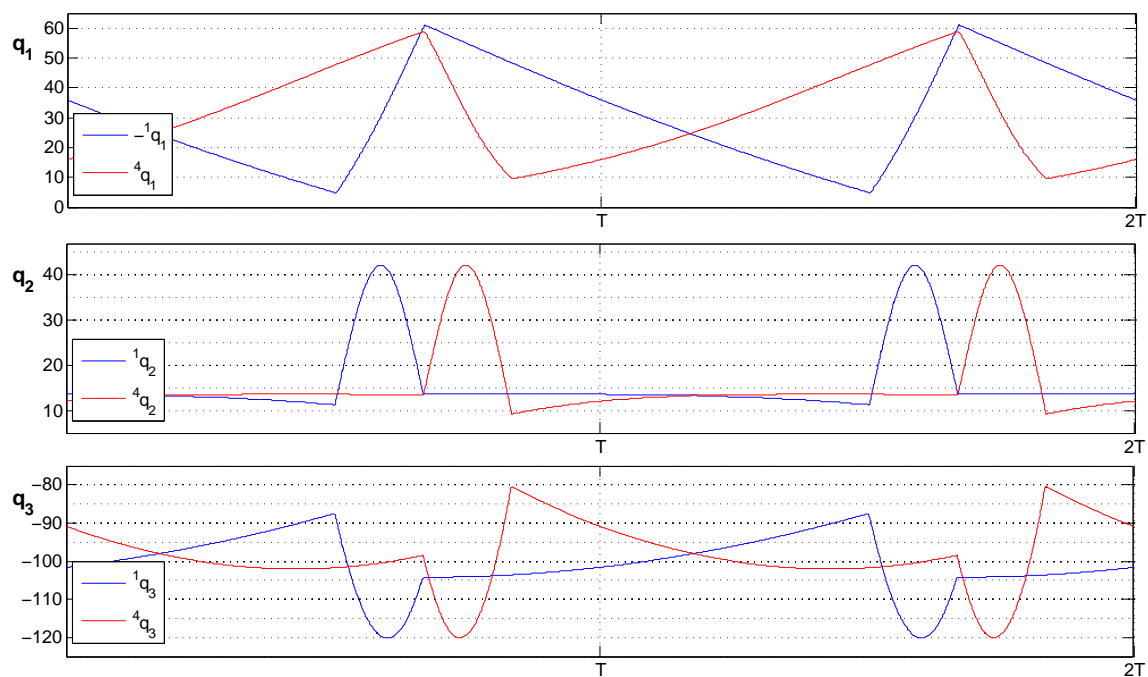
Przebiegi nastaw iq_2 oraz iq_3 są do siebie bardzo zbliżone, gdyż niezależnie od promienia łuku, po którym porusza się robot, noga zawsze podnoszona jest na tą samą wysokość (w tym wypadku $40mm$), zgodnie z tą samą trajektorią $z(t)$.



Rysunek 4.1 Wykres nastaw nóg 1P oraz 4P podczas chodu pięciopodporowego na wprost



Rysunek 4.2 Wykres nastaw nóg 1P oraz 4P podczas chodu pięciopodporowego po łuku o promieniu $300mm$



Rysunek 4.3 Wykres nastaw nóg 1P oraz 4P podczas chodu pięciopodporowego po łuku o promieniu $5mm$

Rozdział 5

Podsumowanie

Celem pracy była konstrukcja sześcionożnego robota, który pozwoliłby na testowania dowolnych algorytmów chodu podczas zajęć laboratoryjnych. Robot musiał posiadać możliwość generowania chodu na podstawie zadanych parametrów oraz jego realizacji z wykorzystaniem rozwiązania zadania odwrotnego kinematyki. Dodatkowo potrzebne było przystosowanie konstrukcji do zdalnego odbioru sygnałów sterujących, aby obecność dodatkowych kabli nie ograniczała robota, ani nie zmniejszała swobody badań. Zadanie to udało się w pełni zrealizować, a uzyskana platforma pozwala na badanie w praktyce dowolnych algorytmów chodu sześcionożnego.

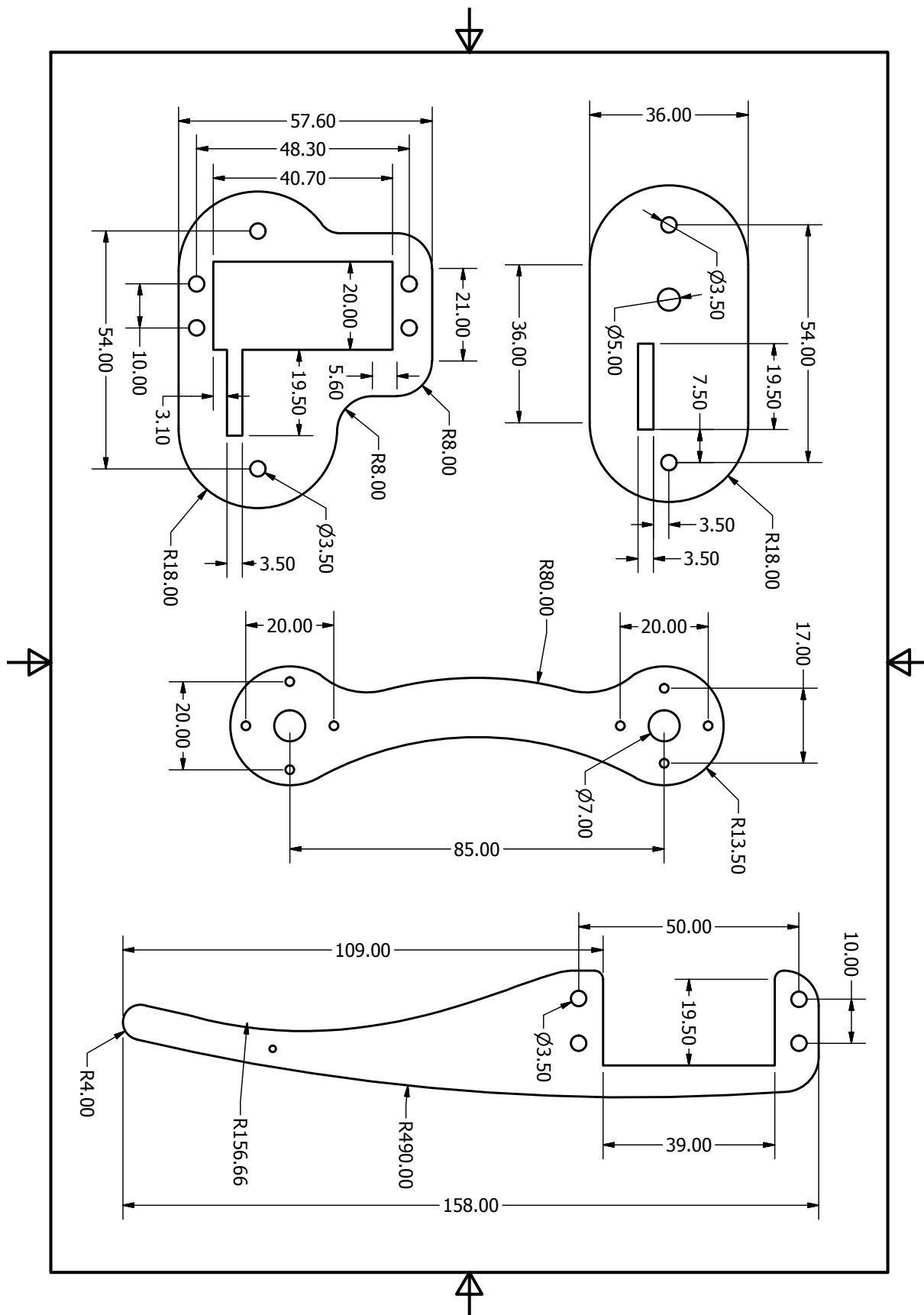
Zbudowany robot może być rozwijany w wielu kierunkach oraz wzbogacany o nową funkcjonalność taką jak na przykład możliwość chodu w dowolnym kierunku (a nie tylko wzdłuż osi robota) lub kontrola kąta nachylenia korpusu względem podłoża (kołysanie, kiwanie oraz myskowanie). Robot może zostać także wyposażony w proste, mechaniczne czujniki dotykowe na końcach nóg oraz żyroskopy, które umożliwią chód po nierównej powierzchni lub budowę mapy terenu. Innym interesującym kierunkiem rozwoju konstrukcji może być próba zapewnienia jej autonomiczności poprzez instalację kilku czujników wykrywających przeszkody (np. dalmierzy działających w paśmie podczerwieni). Tak wyposażony robot mógłby samodzielnie budować mapę podłoża lub szukać drogi pomiędzy przeszkodami. Z myślą o rozszerzeniu funkcjonalności robota, na płycie głównej wprowadzono wiele portów centralnej jednostki sterującej, które dokładnie opisane są w dodatku C.

Trudno porównywać lokomocję żywego owada do mechanicznej konstrukcji, gdyż rozwiązania jakie występują w naturze w dziedzinie materiałów, napędów oraz miniaturyzacji pozostają niedoścignione, jednak systematyczny rozwój konstrukcji mechanicznych gwałtownie zmniejsza jakościowy dystans dzielący je od naturalnych wzorców. Konstrukcja opisana w niniejszej pracy nie powstała w specjalistycznym laboratorium ani z udziałem wyszukanego sprzętu, pewnym ograniczeniem był także budżet przeznaczony na zakup napędów oraz wykonanie konstrukcji, jednak cel jakim było odtwarzanie różnych algorytmów chodu na sześciu nogach został w pełni osiągnięty, co pozwala wierzyć, że wkrótce – w profesjonalnych warunkach z wykorzystaniem lekkich i wytrzymałych materiałów – powstaną maszyny kroczące w niczym nie ustępujące swoim biologicznym odpowiednikom, a może nawet od nich sprawniejsze.

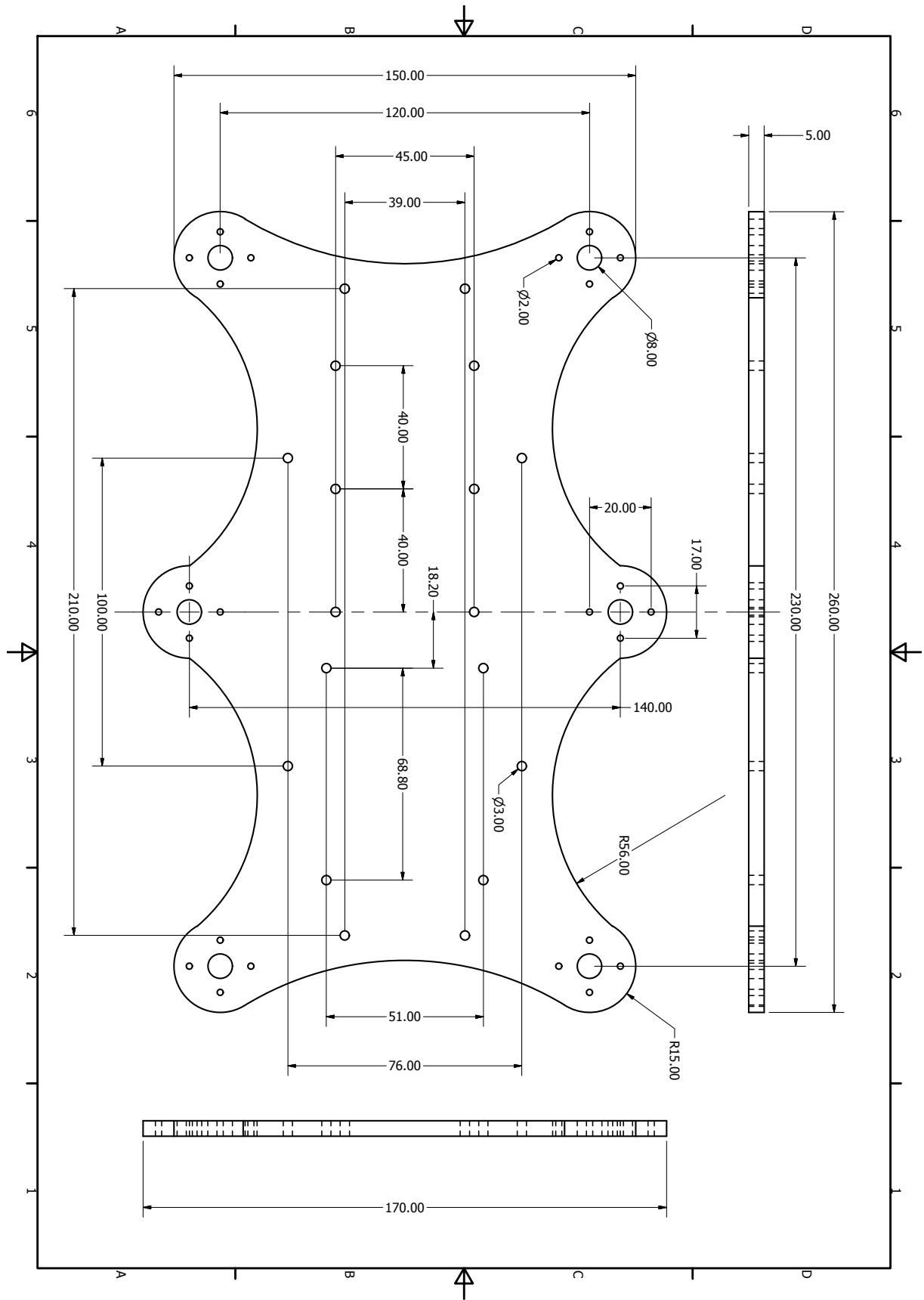
Dodatek A

Rysunki techniczne

W tym dodatku zamieszczono rysunki techniczne poszczególnych elementów konstrukcyjnych robota wraz z wymiarami. Rysunek A.1 ukazuje elementy konstrukcyjne pojedynczej nogi, natomiast rysunek A.2 przedstawia płytę spodnią.



Rysunek A.1 Rysunki techniczne elementów nogi robota

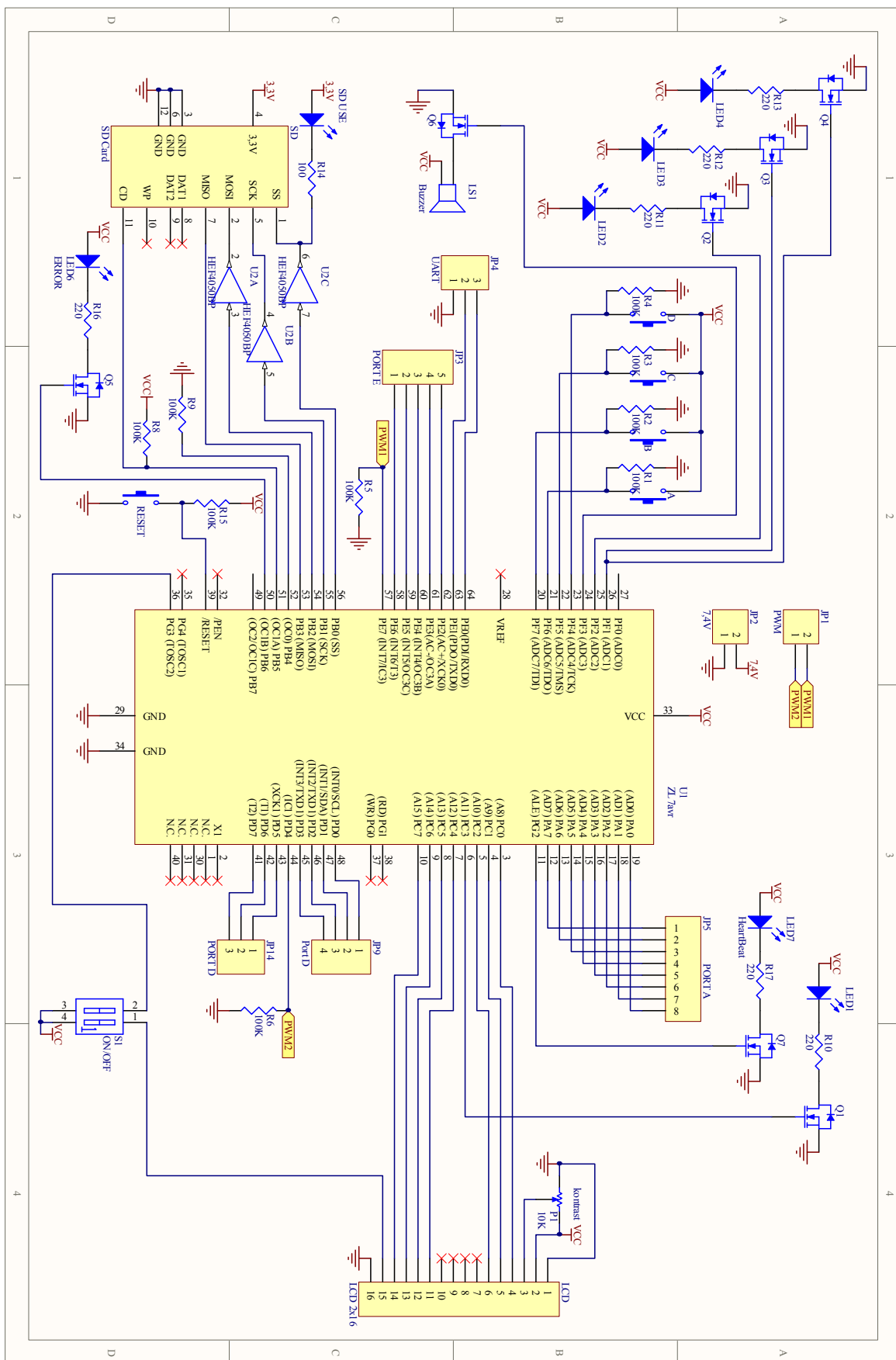


Rysunek A.2 Rysunek techniczny oraz wymiary płyty spodniej

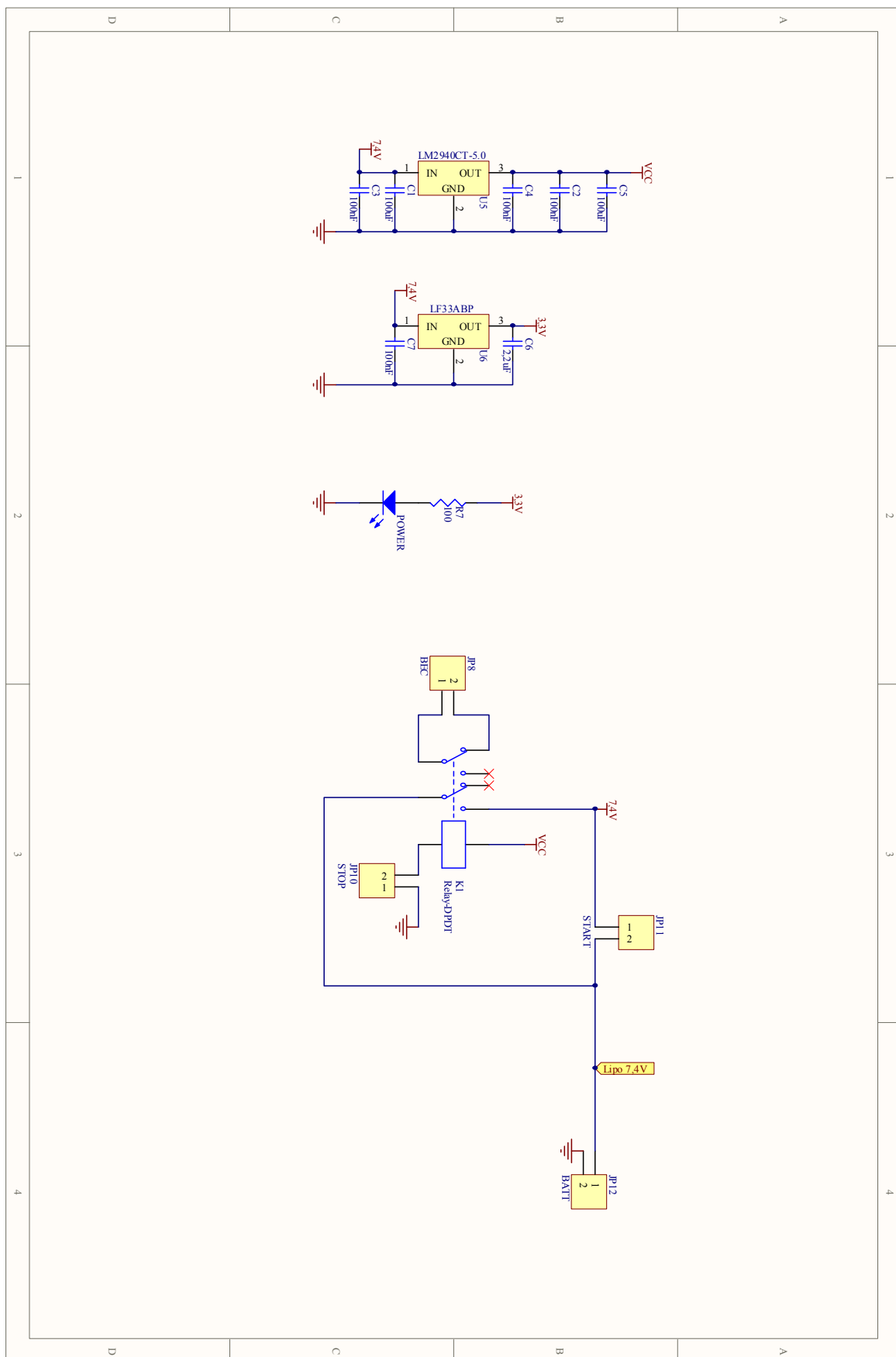
Dodatek B

Ideowy schemat elektroniczny

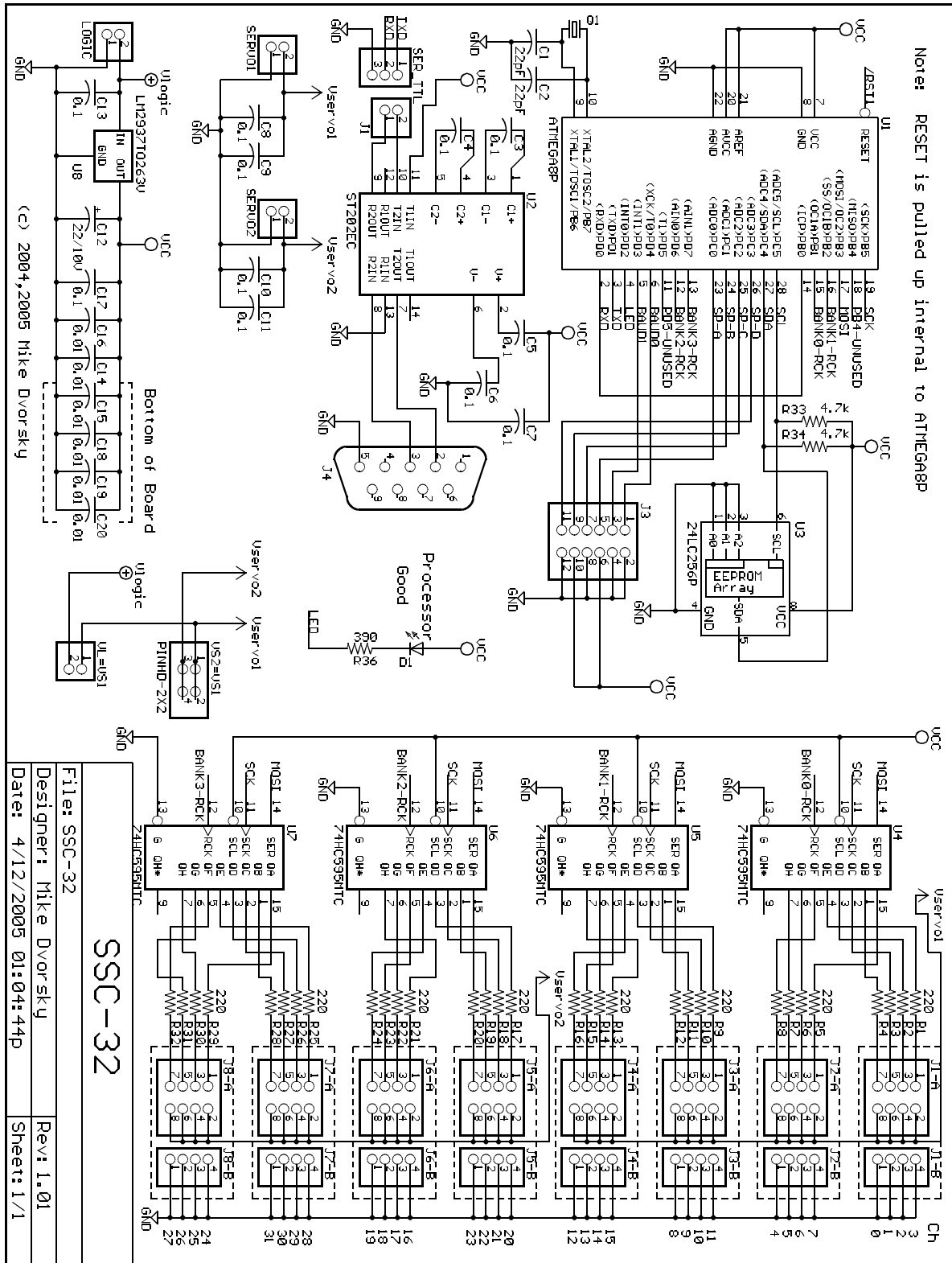
Dla przejrzystości schemat układów elektronicznych płyty głównej został podzielony na dwie osobne części. Rysunek B.1 przedstawia centralną jednostkę obliczeniową oraz wszystkie elementy towarzyszące podczas gdy na rysunku B.2 przedstawiono zlokalizowane na płycie układy zasilania wraz z układem przekaźnika z podtrzymaniem. Dodatkowo na rysunku B.3 pokazano schemat ideowy użytego w pracy serwokontrolera SSC-32.



Rysunek B.1 Schemat ideowy płyty głównej



Rysunek B.2 Schemat ideowy układu zasilania płyty głównej



Rysunek B.3 Schemat ideowy serwokontrolera SSC-32 [10]

Dodatek C

Opis złącz

W tym dodatku opisane są wszystkie złącza obecne na płycie głównej robota. Tabela C.1 zawiera listę podstawowych złącz służących do zasilania, sterowania oraz komunikacji wraz z krótkimi opisami. Tabela C.2 zawiera listę niewykorzystanych złącz uniwersalnych, które w przyszłości mogą posłużyć do rozszerzenia funkcjonalności robota.

Tabela. C.1 Opis podstawowych złącz na płycie głównej robota

Złącze	Pin	Opis	Komentarz
JP12		Złącze zasilania płyty głównej	
	1	plus zasilania (7.4V)	
	2	masa zasilania (GND)	
JP11		Gniazdo włącznika	przełącznik monostabilny NO
	1	(polaryzacja	
	2	bez znaczenia)	
JP10		Gniazdo wyłącznika	przełącznik monostabilny NC
	1	(polaryzacja	
	2	bez znaczenia)	
JP8		Gniazdo sterowania układem BEC	Bez zasilania złącze jest zwarte
	1	(polaryzacja	
	2	bez znaczenia)	
JP4		Złącze UART	Złącze domyślnie użyte do sterowania serwokontrolerem SSC-32
	1	masa (GND)	
	2	TXD0	
	3	RXD0	
JP1		Wejście sygnału PWM	Sygnał z odbiornika zdalnego sterowania kanał określający promień skrętu robota kanał określający prędkość robota
	1	PWM2	
	2	PWM1	
JP2		Zasilanie odbiornika RC	Zasilanie odbiornika zostaje włączone wraz z włączeniem całego robota
	1	masa zasilania (GND)	
	2	plus zasilania (7.4V)	

Tabela. C.2 Opis dodatkowych złącz na płycie głównej robota

Złącze	Pin	Opis	Komentarz
JP5		Dodatkowe złącze - Port A	
	1	PORTA.7 lub ADC.7	
	2	PORTA.6 lub ADC.6	
	3	PORTA.5 lub ADC.5	
	4	PORTA.4 lub ADC.4	
	5	PORTA.3 lub ADC.3	
	6	PORTA.2 lub ADC.2	
	7	PORTA.1 lub ADC.1	
8	PORTA.0 lub ADC.0		
JP9		Dodatkowe złącze - Port D	
	1	PORTD.0	
	2	PORTD.1	
	3	PORTD.2	
JP14		Dodatkowe złącze - Port D	
	1	PORTD.5	
	2	PORTD.6	
JP3		Dodatkowe złącze - Port E	
	1	PORTE.6	
	2	PORTE.5	
	3	PORTE.4	
	4	PORTE.3	
5	PORTE.2		

Dodatek D

Zawartość płyty CD

Do pracy dołączona jest płyta CD z rysunkami technicznymi elementów robota, dokumentacją użytych elementów zgromadzoną podczas pracy oraz kodem źródłowym oprogramowania. Poniżej znajduje się struktura katalogów oraz plików zawartych na płycie, wraz z krótkimi opisami.

D.1 Rysunki techniczne

- Drawings/
 - Inventor/ – Pliki .ipt oraz .iam zawierające odpowiednio projekty poszczególnych części oraz całego robota wykonane za pomocą programu Autodesk® Inventor.
 - Exports/ – Rysunki poszczególnych elementów konstrukcyjnych wyeksportowane do formatów .pdf oraz .cdr.

D.2 Dokumentacja

- Documents/
 - Parts/ – Dokumentacja, instrukcje oraz noty katalogowe części wykorzystanych w robocie.
 - *MgrRG-Strasznyk.pdf* – Niniejsza praca dyplomowa.

D.3 Oprogramowanie

- Sources/ – Pliki z kodem źródłowym programu robota. Program został napisany w języku C w środowisku CodeVisionAVR V2.
- Firmware/
 - *strasznyk.rom* – Obraz pamięci flash dla mikroprocesora AVR ATmega128 zawierający skompilowaną ostateczną wersję programu.
 - *strasznyk.eep* – Obraz pamięci eeprom dla mikroprocesora AVR ATmega128 zawierający zmienne niezbędne do prawidłowego działania programu.

Dodatek E

Instrukcja obsługi

E.1 Obsługa robota

Od razu po włączeniu zasilania robot jest gotowy do używania, a na ekranie zostaje wyświetlone menu, które pozwala na wybór następujących opcji:

1. **Setup** – Wejście do podmenu pozwalającego na zmianę parametrów robota, takich jak długości poszczególnych członów nogi lub wysokość, na jakiej niesiony jest korpus robota.
2. **Kalibr. serw** – Funkcja umożliwiająca kalibrację położen środkowych zainstalowanych serwomechanizmów - szerzej opisana w E.1.1.
3. **Kalibr.zd.ster** – Funkcja umożliwiająca kalibrację minimalnych i maksymalnych zakresów sygnału zdalnego sterowania - szerzej opisana w E.1.2
4. **Wczyt. chodu** – Funkcja umożliwiająca wczytanie parametrów chodu z pliku tekstowego znajdującego się na karcie SD
5. **Parkuj** – Wybór tej funkcji powoduje uniesienie oraz złożenie nóg robota w celu ułatwienia transportu

Dokonanie wyboru możliwe jest przy pomocy przycisków LEWO i PRAWO, natomiast zatwierdzenie – OK. Naciśnięcie przycisku X powoduje zamknięcie menu i podniesienie się robota. Od tego momentu możliwe jest zadawanie prędkości oraz promienia chodu za pomocą aparatury zdalnego sterowania. W każdej chwili możliwy jest powrót do menu za pomocą przycisku X lub zmiana aktualnie stosowanego algorytmu chodu – za pomocą klawiszy LEWO i PRAWO.

E.1.1 Kalibracja serwomechanizmów

Za każdym razem gdy z któregoś serwomechanizmu zostanie zdjęty orczyk lub po dokonaniu wymiany serwomechanizmu konieczna jest kalibracja położenia środkowego danego przegubu. Aby rozpocząć procedurę kalibracji, należy jeszcze przed założeniem orczyka uruchomić robota, a następnie wybrać z menu pozycję „2.Kalibr. serw”. Po zatwierdzeniu przyciskiem OK wszystkie przeguby ustawią się tak aby tworzyć kąty proste (uda skierowane pionowo do góry, natomiast golenie poziomo na boki). Teraz należy zamontować orczyk na osi napędu kalibrowanego przegubu, tak aby wartość kąta była możliwie zbliżona do 90°. Następnie, za pomocą klawiszy LEWO i PRAWO należy odszukać właściwy przegub, a po zatwierdzeniu za pomocą klawisza OK, wyregulować wartość poprawki,

tak aby uzyskać dokładnie kąt prosty. Na koniec całą operację zatwierdzamy klawiszem OK. Liczba wyświetlana na ekranie obok numeru przegubu to mierzona w mikrosekundach wartość poprawki dodawana do szerokości impulsu sygnału PWM sterującego serwo-mechanizmem. Należy dbać o to, aby poprawka nie przekraczała wartości $150\mu s$. Podczas kalibracji warto posłużyć się ekierką.

E.1.2 Kalibracja układu zdalnego sterowania

W związku z dużą różnorodnością aparatów zdalnego sterowania, po ewentualnej wymianie wskazane jest przeprowadzenie kalibracji minimalnych oraz maksymalnych wartości odbieranych sygnałów, w tym celu należy po uruchomieniu robota wybrać z menu pozycję „3.Kalibr.zd.ster”, a następnie zatwierdzić wybór przyciskiem OK.

E.1.3 Ładowanie akumulatorów

Robot wyposażony jest w dwa ogniwa litowo-polimerowe dlatego ładowanie możliwe jest wyłącznie przy pomocy ładowarki dedykowanej do tego typu akumulatorów oraz wyposażonej w balancer. Ładowanie odbywa się za pośrednictwem standardowych kabli laboratoryjnych zakończonych wtyczkami typu „banan”, wpiętych w gniazda znajdujące się w przedniej części wierzchniej obudowy robota. Gniazda zaznaczone są kolorami – czerwony dla bieguna dodatniego oraz czarny dla ujemnego. Wtyczka balancera znajduje się w tylnej części robota, pod płytą główną. Prąd ładowania nie powinien przekraczać wartości 3A. Nie należy używać robota w trakcie ładowania

E.1.4 Demontaż robota

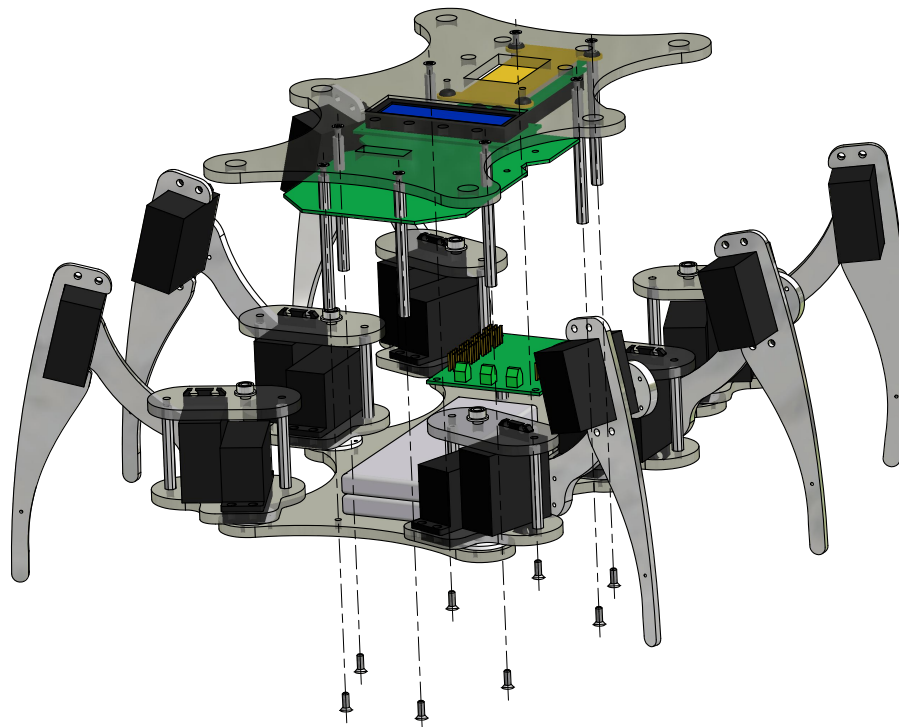
Robot został zaprojektowany w taki sposób aby umożliwiał łatwy dostęp do wnętrza. Aby zdjąć wierzchnią obudowę wraz z płytą główną oraz układem BEC należy odkręcić od spodu robota 8 śrub tak jak to pokazano na rysunku E.1, a następnie wysunąć górne łożyska z gniazd i unieść obudowę. Aby całkowicie rozdzielić obudowę wierzchnią od spodu robota konieczne jest odłączenie baterii, zasilania serwokontrolera oraz przewodu zapewniającego komunikację pomiędzy płytą główną, a serwokontrolerem. Wszystkie przewody zaopatrzone są we wtyczki i pozwalają na rozłączenie bez użycia dodatkowych narzędzi.

E.2 Programowanie

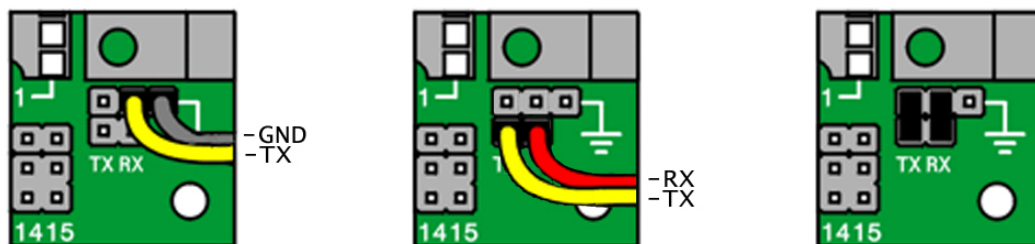
Na płycie głównej robota znajduje się standardowe gniazdo ISP (2×5 pinów) pozwalające na programowanie jednostki centralnej przy pomocy dowolnego programatora zgodnego z tym standardem.

E.2.1 Przywrócenie oryginalnego oprogramowania

W razie potrzeby możliwe jest przywrócenie oryginalnego oprogramowania robota bez potrzeby korzystania ze środowiska, w którym było ono tworzone. W tym celu na płycie CD załączonej do niniejszej pracy zamieszczono obrazy pamięci flash oraz eeprom zawierające skompilowany, oryginalny program (patrz dodatek D).



Rysunek E.1 Sposób demontażu robota



(a) ATmega128 → SSC-32

(b) ATmega128 ↔ komputer

(c) komputer → SSC-32

Rysunek E.2 Możliwości komunikacji za pomocą serwokontrolera SSC-32

E.3 Inne

E.3.1 Komunikacja za pośrednictwem interfejsu RS-232

Robot umożliwia dwukierunkową komunikację z komputerem za pośrednictwem interfejsu RS-232 jednak aby było to możliwe konieczna jest drobna modyfikacja. W przedniej części robota, na płycie serwokontrolera umieszczone jest gniazdo DE-9, obok gniazda znajduje się złącze, pozwalająca na sterowanie serwokontrolerem za pomocą protokołu UART. Rysunek E.2 przedstawia możliwe sposoby wykorzystania wspomnianego złącza. (a) – Sterowanie serwokontrolerem z poziomu płyty głównej robota - ustawienie domyślne, brak komunikacji z komputerem, (b) – Dwukierunkowa komunikacja z komputerem poprzez port DE-9, (c) – Sterowanie serwokontrolerem z poziomu komputera.

Dodatek F

Plik parametrów chodu

Robot został wyposażony w kartę SD o pojemności 8MB oraz gniazdo, pozwalające na wczytanie parametrów chodu za pośrednictwem pliku tekstowego. W tym dodatku opisano składnię pliku parametrów.

Dla każdego algorytmu chodu konieczne jest utworzenie osobnego pliku, będącego listą wszystkich jego parametrów poprzedzonych skróconymi nazwami oraz dwukropkiem. W jednym wierszu może znajdować się tylko jeden parametr. Listę wszystkich parametrów chodu, skróconych nazw oraz ich opisów podano w tabeli F.1. Kolejność podawania parametrów jest dowolna, w wypadku dwukrotnego podania tego samego parametru, zapisana zostanie tylko pierwsza wartość. Długość nazwy pliku parametrów nie powinna przekraczać 8 znaków.

Tabela. F.1 Lista parametrów chodu

Parametr	Skrócona nazwa	Opis
Nazwa chodu	<code>nazwa</code>	Przyjazna nazwa chodu (maksymalnie 16 znaków).
Wysokość chodu	<code>wysokosc</code>	Podana w <i>mm</i> wysokość na jakiej w trakcie chodu niesiony jest korpus robota.
Wysokość kroku	<code>wys_kroku</code>	Podana w <i>mm</i> wysokość na jaką unoszona jest noga.
Długość kroku	<code>dl_kroku</code>	Podana w <i>mm</i> odległość przeniesienia nogi w trakcie fazy protrakcji.
Okres chodu	<code>okres</code>	Mierzony w umownych jednostkach czas przemieszczenia wszystkich nóg robota. Optymalna wartość to 6000.
Współczynnik prędkości	<code>predkosc</code>	Podany w procentach parametr pozwalający na dobranie maksymalnej prędkości chodu ¹ . Wartość mniejsza niż 100 oznacza spowolnienie, natomiast większa – przyspieszenie chodu.
Chwila rozpoczęcia kroku przez <i>i</i> -tą nogę	<code>start_<i>i</i></code>	Wartość określająca chwilę rozpoczęcia fazy protrakcji przez <i>i</i> -tą nogę.
Chwila zakończenia kroku przez <i>i</i> -tą nogę	<code>koniec_<i>i</i></code>	Wartość określająca chwilę zakończenia fazy protrakcji przez <i>i</i> -tą nogę.

¹W wypadku chodów o krótkim okresie (np. chód trójpodporowy) przy zadaniu maksymalnej prędkości nogi mogą nie zdążyć wykonać pełnego ruchu – rozwiązaniem jest zmniejszenie współczynnika prędkości.

Poniżej przedstawiono przykładowy plik parametrów chodu czteropodporowego

```
nazwa:Czteropodporowy
wysokosc:50
wys_kroku:40
dl_kroku:85
okres:6000
predkosc:100
start_1:4000
koniec_1:6000
start_2:2000
koniec_2:4000
start_3:0
koniec_3:2000
start_4:2000
koniec_4:4000
start_5:0
koniec_5:2000
start_6:4000
koniec_6:6000
```

Bibliografia

- [1] Autodesk [®] Education Community. [http://students.autodesk.com/?nd=community_center], 2011. dostęp: 20.10.2011.
- [2] Alturn USA. *Alturn ADS-940HMG datasheet*, 2009.
- [3] Atmel Corporation. *Using Timer Capture to Measure PWM Duty Cycle*, 2005.
- [4] G. Figliolini, P. Rea. *Mechanics and Simulation of Six-Legged Walking Robots*. InTech Education and Publishing, 2007.
- [5] Hitec RCD USA. *Hitec HS-5625MG specifications*, 2010.
- [6] Hitec RCD USA. *Hitec HS-645MG specifications*, 2010.
- [7] Jeti model. MAX BEC 2 manual, 2010.
- [8] Kamami. *ZL7AVR Moduł z mikrokontrolerem ATmega128 - dokumentacja*, 2008.
- [9] Lynxmotion inc. Hexapod Phoenix users guide, 2008.
- [10] Lynxmotion inc. SSC-32 users guide, 2010.
- [11] R. McGhee. Some Finite State Aspects of Legged Locomotion. *Mathematical Bioscience* 2, strony 67–84, 1968.
- [12] B. Miripour, redaktor. *Climbing and Walking Robots*. InTech, 2010.
- [13] J. M. Porta, E. Celaya. *Gait Analysis for Six-Legged Robots*. Institut de Robotica i Informatica Industrial, 1998.
- [14] K. Tchoń, A. Mazur, I. Duleba, R. Hossa, R. Muszyński. *Manipulatory i roboty mobilne*. Akademicka Oficyna Wydawnicza PLJ, 2000.
- [15] K. Thilo, A. Jan, G. Bernd, Z. J. Marius. *Lauron IV robot datasheet*. FZI, 2005.
- [16] L. Viatour. *Peruphasma schultei*. Wikimedia Commons, 2009.
- [17] T. Zielińska. *Maszyny kroczące*. PWN Warszawa, 2003.
- [18] T. P. Zieliński. *Cyfrowe przetwarzanie sygnałów*. WKŁ, 2009.