

# MRÓWA — raport postępów prac

Dominik Urban  
Koło Naukowe Robotyków „KoNaR”

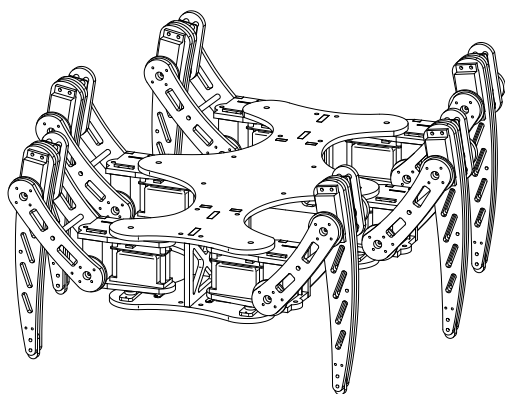
16 listopada 2009

## 1 Wstęp

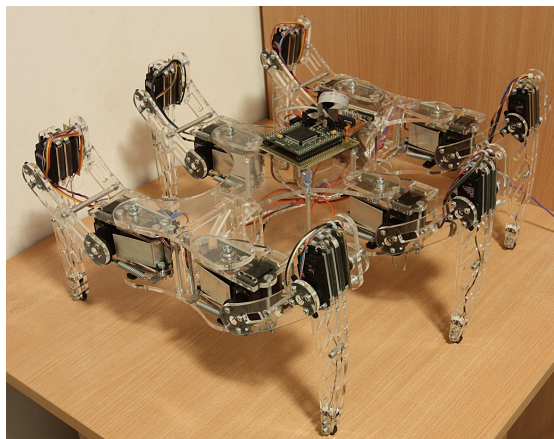
Celem projektu jest wykonanie sześcionożnego robota krocącego MRÓWA zdolnego pokonywać przeszkody i pochyłości o nieznacznej wysokości. Robot ma mieć możliwość rejestrowania nierówności terenu i następnie za pomocą oprogramowania zawartego w komputerze nadrzędnym wykonać mapę wysokości terenu. Niniejszy raport ma na celu przybliżenie postępów prac nad robotem. Dokument zawiera poglądowy opis mechaniki robota, elektroniki, kinematyki oraz trójpodporowego algorytmu chodzenia.

## 2 Mechanika

Mechanika robota zaprojektowana została w programie Solid Edge v19. Wybór padł na program Solid Edge z kilku powodów: program dostępny jest za darmo w wersji studenckiej, jest bardzo prosty w obsłudze, cały etap projektowania odbywa się w środowisku trójwymiarowym, dzięki czemu łatwo wyeliminować wiele błędów już na etapie projektowania.



(a) projekt

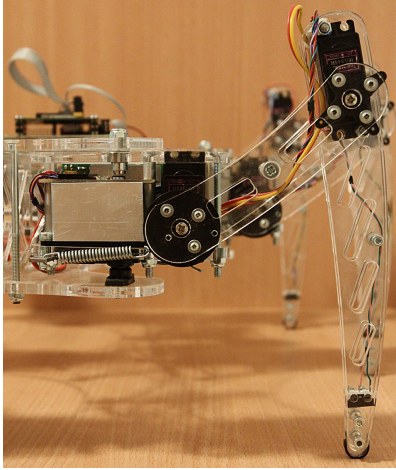


(b) realizacja

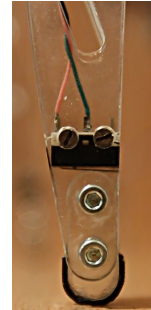
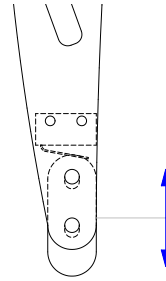
Rysunek 1: Robot kroczący MRÓWA

Konstrukcja zaprojektowana jest z myślą o łatwym serwisowaniu robota. Wszystkie łączenia elementów wykonane są bez użycia kleju tak, aby możliwa była łatwa wymiana uszkodzonych elementów. Robot w całości wykonany został z plexi o grubości  $5\text{mm}$ . Wszystkie elementy cięte były z wykorzystaniem plotera laserowego. Elementy dystansowe wykonane zostały z rurek aluminiowych. Do napędu nóg robota użyte zostały standardowe serwa modelarskie TowerPro MG995. Wybór padł na ten model ze względu na cenę i stosunkowo dużą siłę, około  $[15 \frac{\text{kg}}{\text{cm}}]$ . Serwa pozbawione zostały elektroniki, pozostawiono jedynie silnik. Cały sterownik serwa zaprojektowany został na nowo z wykorzystaniem mikroprocesora HC12. Dokładniejszy opis przedstawiono w rozdziale 5.

Ważnym ograniczeniem projektu był ciężar konstrukcji. Konieczność pokonywania przeszkód wymaga uzyskania możliwie jak najwyższego prześwitu robota. Zastosowanie długich kończyn w połączeniu z silnikami o niewielkiej mocy wymagało ograniczenia masy robota, co przysparzało wiele problemów. Początkowo projekt zakładał segmenty środkowe nogi o długości około 16cm, jednak ostatecznie, po wykonaniu testów, elementy skrócone zostały do długości 9.5cm. Ponadto zastosowane zostały sprężyny (rysunek [2]), które pozwalają uzupełnić niedostatek mocy silników i w niektórych konfiguracjach całkowicie dźwigają ciężar robota, ograniczając tym samym pobór prądu. Aby monitorować oparcie nogi robota o podłoże, zaprojektowano stopę połączoną z prostym przełącznikiem (rysunek [3]).



Rysunek 2: Sprężyna podtrzymująca ciężar robota



Rysunek 3: Stopa robota

### 3 Kinematyka robota

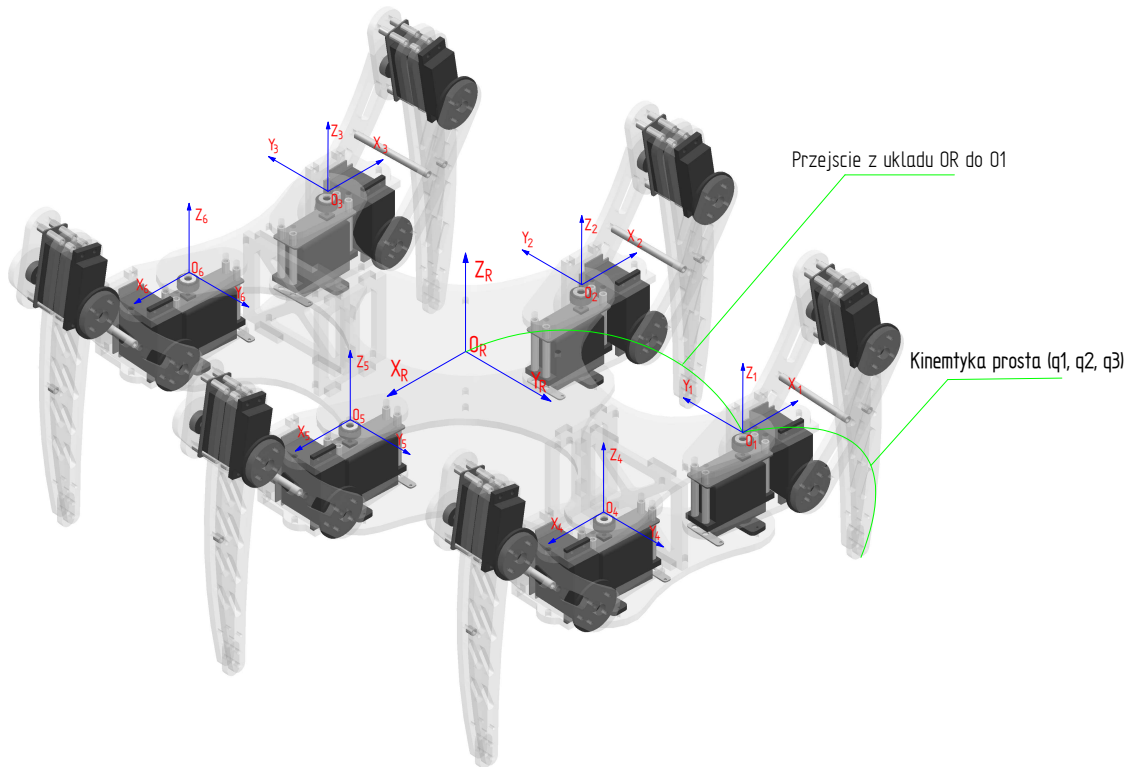
Aby przemieścić korpus, robot kroczący wymaga synchronicznej współpracy swoich kończyn. Generator chodu oblicza kolejne pozycje stopy robota w układzie 3R, natomiast jedyną wielkością, jaką można sterować, jest wychylenie katowe poszczególnych segmentów nogi. Aby móc połączyć poziom generowania chodu robota z oprogramowaniem kontrolującym wychylenie katowe segmentów, konieczna jest implementacja odwrotnego zadania kinematyki nogi robota. Wzorując się na pracy [1], kinematykę robota kroczącego można przedstawić jako połączenie sześciu manipulatorów. Pojedynczą nogę robota można przedstawić jako manipulator o trzech stopniach swobody, połączony z pozostałymi za pomocą korpusu. Podejście takie pozwala wyznaczyć rozwiązanie odwrotnego zadania kinematyki osobno dla każdej z nóg robota.

#### 3.1 Proste zadanie kinematyki robota

Rysunek 4 przedstawia rozmieszczenie lokalnych układów współrzędnych dla każdej z nóg robota. Oznaczone są one przez  $O_i$ , gdzie  $i$  – numer nogi. Należy pamiętać, że układy te umieszczone są na wysokości przegubu  $q_2$ . Dla poprawy czytelności rysunku umieszczone one zostały na wysokości górnej płyty korpusu. Wymiary robota przedstawione są na rysunku 5.

Macierz transformacji pomiędzy układem  $O_R$ , a układami  $O_i$ , ma postać:

$$T_{O_i}^{O_R} = \begin{cases} Trans(X, l_1) Trans(Y, l_2) Rot(Z, \pi) & \text{dla } i = 1, 2, 3 \\ Trans(X, l_1) Trans(Y, l_2) & \text{dla } i = 4, 5, 6 \end{cases} \quad (1)$$



Rysunek 4: Rozmieszenie lokalnych układow̄ wspó̄rzędnych, dla kādej z nóg

po obliczeniu uzyskuje się następujące macierze transformacji:

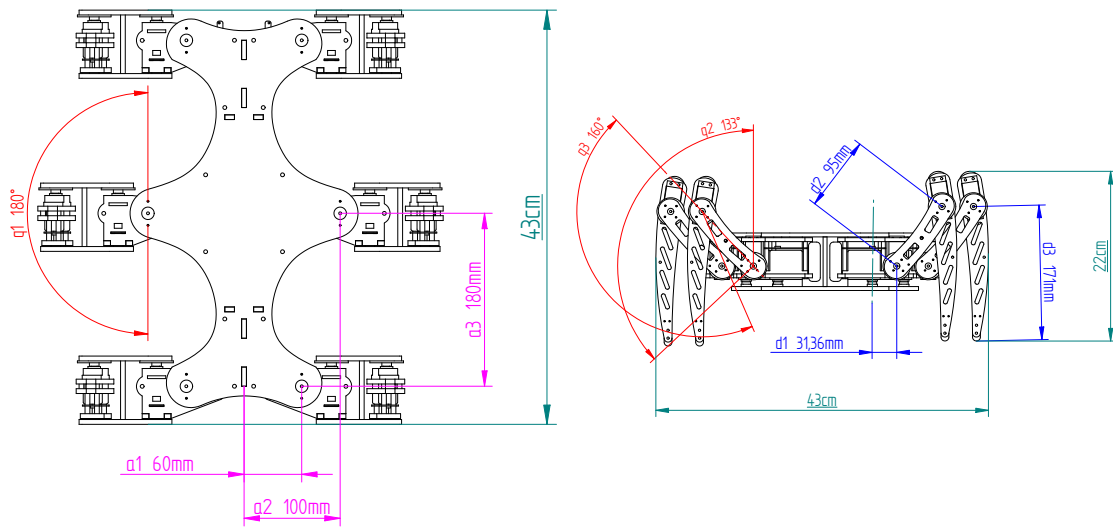
$$T_{O_i}^{O_R} = \begin{cases} \begin{bmatrix} 1 & 0 & 0 & l_1 \\ 0 & -1 & 0 & l_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \text{dla } i = 1, 2, 3 \\ \begin{bmatrix} 1 & 0 & 0 & l_1 \\ 0 & 1 & 0 & l_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \text{dla } i = 4, 5, 6, \end{cases}$$

gdzie  $l_1$  i  $l_2$  oznaczają pozycję lokalnego układu współrzędnych nogi względem układu  $O_R$ . Wartości te można odczytać z rysunku 5. Na przykład dla nogi pierwszej ( $i = 1$ ),  $l_1 = -a_1$ ,  $l_2 = a_3$ , natomiast dla  $i = 4$ ,  $l_1 = a_1$ ,  $l_2 = a_3$ .

Do wyznaczenia kinematyki pojedynczej nogi robota wykorzystuje się podejście stosowane dla manipulatorów przemysłowych. Rysunek 6 przedstawia rozmieszczenie układów współrzędnych zgodnych z notacją Denavita–Hartenberga.

Przejs̄cie do kolejnych układów współrzędnych wygląda następująco:

$$\begin{aligned} {}^0T_1 &= Rot(Z, q_1), \\ {}^1T_2 &= Rot\left(X, -\frac{\pi}{2}\right) Trans(X, d_1) Rot(Z, q_2), \\ {}^2T_3 &= Trans(X, d_2) Rot(Z, q_3), \\ {}^3T_4 &= Trans(X, d_3). \end{aligned}$$



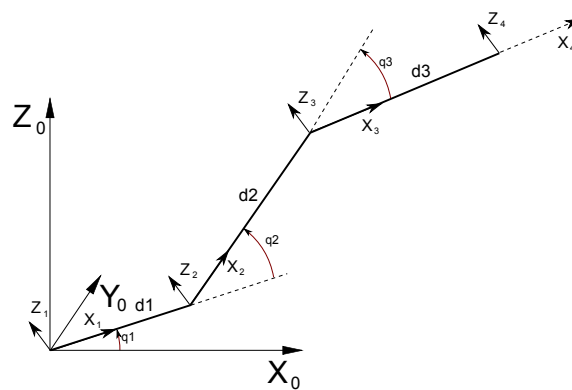
(a) widok z góry

(b) widok z przodu

Rysunek 5: Poglądowe wymiary robota wraz z zakresami ruchu poszczególnych segmentów nogi

Po obliczeniu wektor translacji  ${}^0T_4$  ma postać:

$$\begin{cases} x = c_1 (c_{23}d_3 + c_2d_2 + d_1) \\ y = s_1 (c_{23}d_3 + c_2d_2 + d_1) \\ z = s_{23}d_3 + s_2d_2 \end{cases} \quad (2)$$

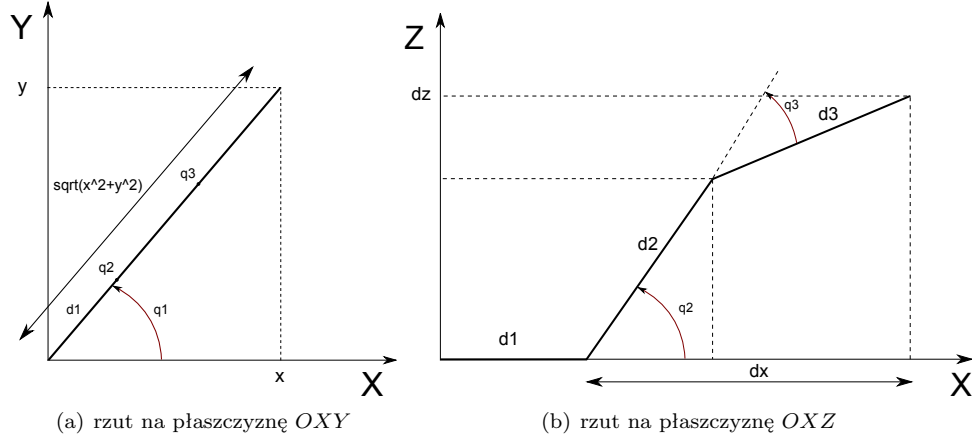


Rysunek 6: Kinematyka nogi

### 3.2 Odwrotne zadanie kinematyki nogi robota.

Do rozwiązania zadania odwrotnego kinematyki zastosowano metodę algebraiczną. Dzieląc  $y$  przez  $x$ , z zależności 2 uzyskuje się

$$q_1 = \text{atan2}(y, x). \quad (3)$$



Rysunek 7: Kinematyka odwrotna nogi

Rysunek 7 przedstawia rzuty nogi na płaszczyznę  $OXY$  i  $OXY$ . Z twierdzenia Pitagorasa można określić długość nogi na płaszczyźnie  $OXY$ :

$$\sqrt{x^2 + y^2}.$$

Można więc wyznaczyć długości:

$$\begin{aligned} dx &= \sqrt{x^2 + y^2} - d_1, \\ dz &= z. \end{aligned}$$

Następnie, patrząc tylko na segmenty  $d_2$  i  $d_3$ , uzyskuje się

$$\begin{aligned} c_3 &= \frac{dx^2 + dz^2 - d_2^2 - d_3^2}{2d_2d_3}, \\ s_3 &= \pm\sqrt{1 - c_3^2} \end{aligned}$$

i w efekcie

$$q_3 = \text{atan2}(\pm s_3, c_3). \quad (4)$$

Wybór znaku funkcji  $s_3$  jest uzależniony od postury nogi. Znak ujemny oznacza posturę typu nogi owada (zawsze zgięta w kolanie).

Obliczenie kąta  $q_2$  jest bardziej skomplikowane. Dokładny przebieg obliczeń przedstawiony jest w pozycji [1]. Poniżej przedstawiony jest jedynie skrócony przebieg obliczeń.

$$\begin{aligned} m_1 &= d_2 + d_3c_3, \\ m_2 &= d_3s_3, \\ r &= \sqrt{m_1^2 + m_2^2} \end{aligned}$$

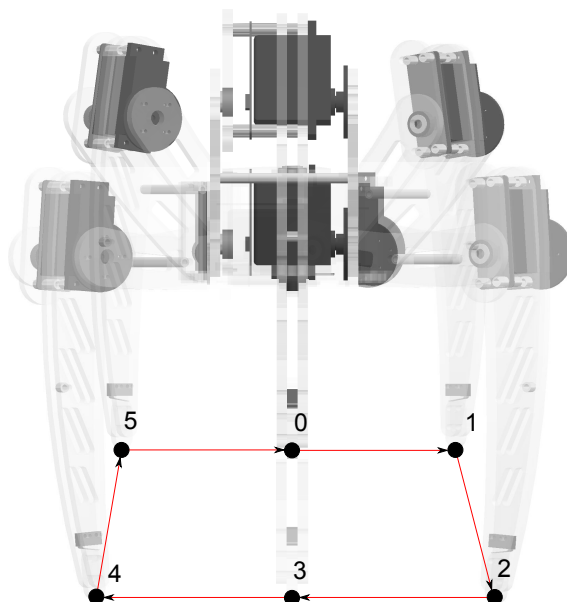
$$q_2 = \text{atan2}\left(\frac{dz}{r}, \frac{dx}{r}\right) - \text{atan2}(m_2, m_1). \quad (5)$$

Należy pamiętać o wprowadzonych ograniczeniach: postura typu noga owada (zawsze zgięta w kolanie), kąt  $q_1 \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$  — przekroczenie tego zakresu oznaczałoby kolizję nogi z korpusem.

## 4 Generator chodu

Sterowanie ruchem robota stanowi złożony problem. Aby robot mógł śledzić zadaną ścieżkę, konieczne jest odpowiednie sterowanie ruchem kończyn, pozwalające przeprowadzać korpus do kolejnych punktów ścieżki. Podczas generowania ruchu robota konieczne jest zdefiniowanie typu chodu

robota. Typy chodu opracowywane są często na podstawie wzorców biologicznych np. ssaki, owady. O ile ssaki najczęściej korzystają z chodu dynamicznego, owady korzystają z chodu stabilnego statycznie. Pozwala on utrzymać równowagę poprzez odpowiednie przemieszczenie środka ciężkości. W maszynach sześcionożnych stosuje się najczęściej chód statyczny trójpodporowy lub pięciopodporowy. W rozdziale tym przedstawiony został algorytm chodu trójpodporowego po płaskiej powierzchni.



Rysunek 8: Fazy kroku

#### 4.1 Trajektoria ruchu końca nogi

Trajektoria końca nogi zadana jest w przestrzeni kartezjańskiej i złożona jest z sześciu faz:

- 0 — faza środkowa niepodparta,
- 1 — faza wykroczna niepodparta,
- 2 — faza wykroczna podparta,
- 3 — faza środkowa podparta,
- 4 — faza zakroczna podparta,
- 5 — faza zakroczna niepodparta.

Rysunek 8 przedstawia konfigurację nogi w poszczególnych fazach kroku.

Tablica 1: Etapy kroku

Etapy	0	1	2	3
noga 1,3,5	0	1	2	3
noga 2,4,6	3	4	5	0

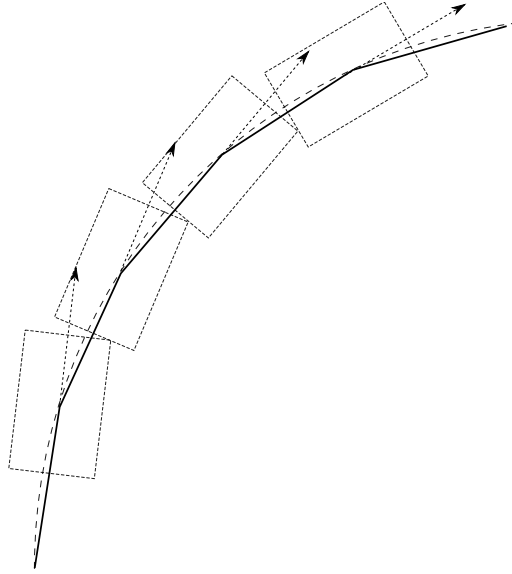
Część trajektorii uczestniczącej w przemieszczeniu korpusu to fazy od 2 do 4. Jako że rozpatrywany jest ruch trójpodporowy, trzy nogi współpracują ze sobą naprzemiennie w zadaniu przemieszczenia korpusu.

Tablica 4.1 przedstawia kolejne etapy przemieszczenia robota o jeden krok. Kompletny krok składa się z 4 etapów. Etap zerowy jest etapem wyjściowym, w którym nogi znajdują się w pozycjach środkowych. Przemieszczenie korpusu robota następuje w etapie 2–3 przez nogi nieparzyste

oraz w etapie 0–1 przez nogi parzyste. Kolejno nogi parzyste i nieparzyste przemieszczają robota o połowę długości kroku. Zapętlenie tych etapów powoduje, że robot zaczyna chodzić.

## 4.2 Trajektoria ruchu końca nogi przy zmianach kierunku ruchu

W fazie podporowej nogi stykają się z nieruchomym podłożem, dlatego też trajektorie końców nóg względem korpusu odpowiadają trajektorii ruchu korpusu względem podłoża. W trakcie ruchu do przodu końce nóg przemieszczają się w fazie podporowej wzdłuż osi  $OX$ , natomiast podczas przemieszczenia w bok — wzdłuż osi  $OY$ . Problem pojawia się w przypadku skręcania w ruchu postępowym. Trajektorie ruchu końców nóg w fazie podporowej powinny być odpowiednimi łukami, natomiast ich długość musi odpowiadać długości kroku. Rozwiązanie takie jest kłopotliwe i częściej stosuje się metodę przybliżenia łuku skreću odcinkami. Metoda ta nie wymaga zmian kształtu trajektorii końca nogi z wyjątkiem modyfikacji współrzędnych początku i końca fazy podporowej.



Rysunek 9: Obrót korpusu

Koniec nogi wyrażony jest w lokalnym układzie współrzędnych danej nogi  $O_i$  ( $i = 1, \dots, 6$ ), a ten z kolei wyrażony jest w układzie robota  $O_R$ . Aby zrealizować przemieszczenie korpusu robota w nową pozycję, należy wyznaczyć położenie końców nóg na końcu fazy podporowej.

Niech wektor  $p_{i3}$  określa pozycję końca nogi wyrażoną w układzie  $O_i$  w fazie 3. Określa on zatem pozycję nogi, gdy robot nie wykonuje żadnego ruchu. Dla przykładu:  $p_{i3} = [100, 0, -100, 1]^T$ . Niech  $O'_R$  oznacza pozycję korpusu robota na końcu fazy podporowej, natomiast pojedynczy krok niech będzie opisany parametrami:

- $offset_{up}$  — wysokość, na jaką jest podnoszona noga;
- $offset_x$  — przemieszczenie robota wzdłuż osi  $OX$  w trakcie jednego kroku;
- $offset_y$  — przemieszczenie robota wzdłuż osi  $OY$  w trakcie jednego kroku;
- $\alpha_z$  — skręt robota wzdłuż osi  $OZ$  w trakcie jednego kroku.

Translacja układu  $O'_R$  w  $O_R$  ma postać:

$$T' = \begin{bmatrix} \cos(\alpha_z) & -\sin(\alpha_z) & 0 & -offset_x \\ \sin(\alpha_z) & \cos(\alpha_z) & 0 & offset_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (6)$$

Aby wyznaczyć pozycję nogi na końcu fazy podporowej  $p_{i4}$  wyrażoną w układzie lokalnym nogi  $O_i$ , należy wyrazić pozycję końca nogi  $p_{i3}$  w układzie  $O_R$ , dokonać translacji  $T'$ , a następnie wyrazić otrzymaną pozycję w układzie  $O_i$ :

$$\begin{aligned} p_{i3}^R &= O_i * p_{i3}, \\ p_{i4}^R &= T' * p_{i3}^R, \\ p_{i4} &= O_i^{-1} * p_{i4}^R. \end{aligned} \tag{7}$$

Konieczne jest jeszcze obliczenie pozycji końca nogi w pozostałych fazach kroku:

$$\begin{aligned} P_i &= p_{i4} - p_{i3}, \\ up &= [0 \quad 0 \quad offset_{up} \quad 1]^T, \\ p_{i2} &= p_{i3} - P_i, \\ p_{i1} &= p_{i2} + up, \\ p_{i0} &= p_{i3} + up, \\ p_{i5} &= p_{i4} + up. \end{aligned} \tag{8}$$

### 4.3 Trajektoria ruchu końca nogi przy zmiennej rotacji i nutacji korpusu

W przypadku konieczności utrzymania korpusu w poziomie podczas kroczenia po pochyłych powierzchniach konieczna jest możliwość kroczenia ze zmiennym nachyleniem korpusu.

Wprowadzone zostaną dodatkowe parametry kroku:

- *roll* — (nutacja) obrót względem osi  $OY$ ,
- *pitch* — (rotacja) obrót względem osi  $OX$ ,
- *offset<sub>z</sub>* — wysokość korpusu nad powierzchnią.

Niech

$$T'' = Trans(Z, offset_z)Rot(X, pitch)Rot(Y, roll). \tag{9}$$

Pozycje  $p_{i3}^R$  [7] należy zastąpić przez

$$p_{i3}^R = T'' O_i p_{i3}, \tag{10}$$

wektor  $P_i$  [8]

$$P_i = Rot(X, pitch)Rot(Y, roll)(p_{i4} - p_{i3}) \tag{11}$$

oraz na nowo policzyć

$$p_{i4} = p_{i3} + P_i. \tag{12}$$

## 5 Elektronika

Elektronika w robocie nadal jest na etapie projektowania. Przeprowadzane są ciągle nowe testy i eksperymenty, które doprowadzić mają do powstania płyty głównej robota scalającej w sobie większość modułów. W raporcie tym przedstawiony zostanie ukończony już sterownik pojedynczej nogi robota oraz omówione zostaną planowane w przyszłości moduły.

Jednostką centralną robota jest 32 bitowy mikroprocesor firmy Freescale MC332. Wybrany on został głównie ze względu na rozbudowany moduł QSPI. W jednostce tej zaimplementowany został algorytm chodzenia, przy czym liczenie odwrotnego zadania kinematyki odbywa się w sterowniku nogi. Rozwiązanie takie pozwala odciążyć jednostkę centralną, oraz pozwala na sterowanie nogą robota poprzez proste wysyłanie pozycji końca nogi. Komunikacja pomiędzy jednostką centralną a poszczególnym sterownikiem nogi odbywa się przez magisteralę SPI, przy czym każdy sterownik nogi podpięty jest pod indywidualny Chip Select. Podczas komunikacji pomiędzy jednostką centralną a sterownikiem nogi konieczne okazało się wprowadzenie kontroli przesyłanych danych, aby wyeliminować błędne informacje (zastosowano sumę CRC).



Obecnie trwają prace nad modulem lokalizacji robota. Moduł składał się będzie z akcelero-  
metrów i żyroskopów oraz kompasu elektronicznego, o ile wpływ magnesów okaże się nieznaczący.  
Głównym zadaniem modułu będzie poziomowanie korpusu robota podczas pokonywania nierów-  
ności oraz lokalizacja robota. Komunikacja między modulem a jednostką centralną odbywać się  
będzie przez magistralę SPI. Cyfrowa filtracja sygnałów oraz reszta obliczeń zrealizowana zostanie  
wewnątrz jednostki centralnej. Z tego powodu istotne jest jej odciążenie.

Komunikacja z nadrzędnym komputerem zrealizowana zostanie za pomocą modułu bluetooth.  
W połączeniu z zasilaniem bateryjnym pozwoli to na wyeliminowanie przewodów ciągnących się  
za robotem. Cała warstwa nadrzędna dotycząca planowania ścieżki robota prawdopodobnie zreali-  
zowana zostanie w komputerze nadrzędnym. Zaimplementowana zostanie wizualizacja robota (w  
dużej mierze już istniejąca) oraz wizualizacja mapy terenu stworzonej przez robota.

## 5.1 Sterownik nogi robota

Ze względu na niezadowalającą pracę oryginalnych sterowników serw, mianowicie szarpanie i nie-  
równą pracę podczas powolnego przemieszczania, zbudowany on został od nowa. Niezależne ste-  
rowniki pojedynczych serw scalono w jednym układzie. Płytką umieszczoną jest wewnątrz nogi  
robota, co pozwoliło na minimalizację przewodów łączących nogę robota z korpusem. Obecnie są  
to jedynie przewody zasilające i ekranowane przewody magistrali SPI.

Sterownik oparty został na 16-sto bitowym mikrokontrolerze firmy Freescale HCS12C32. Wy-  
chylenie kątowe poszczególnych segmentów nogi określane jest za pomocą magnetycznych enkode-  
rów umieszczonych za serwem. Silniki sterowane są sygnałem PWM. Jako wzmacniacze wykorzy-  
stane zostały proste mostki H, mianowicie L293D. Wzmacniacze te są w dużych obudowach DIP,  
co pozwoliło łatwiej pozbyć się wydzielanego ciepła. Radiator służący tym celom to aluminiowa  
blacha zagięta wokół jednego z serw. Widoczna jest ona na przedstawionych uprzednio zdjęciach.

Na płytce sterownika umieszczone zostały również układy monitorujące prąd pobierany przez  
silniki. Zrealizowane są one na wzmacniaczach operacyjnych badających spadek napięcia na rezy-  
storze wzorcowym. Są to wzmacniacze firmy National Semiconductor dedykowane do takich wła-  
śnie zastosowań. Pozwalają one na pomiar prądu płynącego w obu kierunkach. Rezystor wzorcowy  
można umieścić bezpośrednio w szeregu z uzwojeniem silnika. Napięcie na wyjściu wzmacniacza  
przyjmuje wartości od 2.5 V do 0 V lub od 2.5 V do 5 V w zależności od tego, w którą stronę  
płynie prąd.

Wewnątrz mikrokontrolera zaimplementowany został sterownik w dużej mierze wzorowany na  
implementacji OpenServo. Jest to regulator PD połączony z generatorem trajektorii. Generator  
trajektorii ma za zadanie płynnie przeprowadzić serwo z jednej pozycji do drugiej w zadanym  
czasie. Generator trajektorii wykorzystuje krzywe Hermite'a, dzięki którym można zadawać różne  
wzorce prędkości, np. serwo może łagodnie startować na początku, zwalniać na końcu, natomiast  
maksimum prędkości osiąga w połowie odległości pomiędzy zadanymi pozycjami. Dzięki zastoso-  
waniu takiego sterowania udało się uzyskać płynność ruchu niemożliwą do osiągnięcia dla taniego  
serwa modelarskiego. Problem jest nadal podczas trzymania zadanej pozycji, ale powinno dać się  
to wyeliminować poprzez odpowiednie wyregulowanie wzmocnień regulatora PD.

Schemat ideowy sterownika nogi przedstawiony jest na rysunku 10.

## Literatura

- [1] Teresa Zielińska.: *Maszyny Kroczące*, PWN Warszawa 2003.
- [2] G. Figliolini, V. Ripa *Kinematics Model and Absolute Gait Simulation of a Six-Legged Walking Robot*, LARM, University of Cassino, Italy.
- [3] Dominik Belter *System sterowania ruchem sześcionożnego robota kroczącego*, Krajowa Konfe-  
rencja Robotyki 2008.

