

**A G H**

**AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE**  
**Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej**

## Praca dyplomowa

### *Projekt Robotu Trójnożnego Three-legged Robot*

Autor: *Jakub Mazur*  
Kierunek studiów: *Automatyka i Robotyka*  
Opiekun pracy: *dr. inż. Łukasz Więckowski*

Kraków, 2020



*Serdecznie dziękuję ... tu ciąg dalszych podziękowań np. dla promotora, itd  
TODO*



# **Spis treści**

<b>1. Wprowadzenie .....</b>	7
1.1. Historia robotów kroczących .....	7
1.2. Istniejące konstrukcje trójnożne .....	8
1.2.1. STriDER.....	9
1.2.2. Triped Martian .....	10
1.2.3. Inne konstrukcje.....	11
<b>2. Model Matematyczny .....</b>	13
2.1. Noga robota typu RRR .....	13
2.1.1. Kinematyka Prosta .....	14
2.1.2. Kinematyka prosta - metoda Denavita Hartenberga [DH_AA_article].....	15
2.1.3. Invert kinematics.....	17
2.2. Cały Robot .....	19
2.2.1. Matematyka kroku .....	19



# 1. Wprowadzenie

## 1.1. Historia robotów kroczących

Pierwsza idea robota kroczącego pojawiła się już pod koniec XV wieku, a narodziła się w głowie nikogo innego jak Leonarda Da Vinci. Od tamtej pory wielu naukowców próbowało tworzyć konstrukcje, które używały nóg zamiast kół. Jednakże, pierwsze faktycznie udane roboty tego typu datuje się dopiero na początek lat 60 ubiegłego wieku. Pojawiły się wtedy pierwsze działające konstrukcje, na przykład robot czteronożny zbudowany przez Josepha Shingleya oraz roboty sześciu i ośmionożne zbudowane przez "Space General Corporation".  
[history]

Od tamtej pory powstało wiele różnych projektów, które kategoryzuje się w zależności od ilości nóg posiadanych przez robota:

- jednonożne,
- dwunożne (Humanoid, chicken-walkers),
- czteronożne (Quadrupedal),
- sześcionożne (Hexapod),
- ośmionożne,
- gąsiennicowe.

Można tu zaobserwować pewną tendencję spadkową, wraz z upływem czasu widać wzrost udanych konstrukcji o mniejszej liczbie nóg. Konstrukcje takie wymagają większej wiedzy naukowców, lepiej dobranych algorytmów ale za to można je skonstruować mniejszym nakładem materiałowym. Stąd naturalne jest dążenie do ograniczania liczby nóg w konstrukcjach robotów kroczących

Pojawia się także inna tendencja w ilości nóg robotów. Prawie wszystkie konstrukcje (poza jednonożnymi) opierają się na anatomii zwierząt. Jest to raczej logiczna tendencja, jako że do takich robotów mamy już algorytmy chodu opracowane przez miliony lat ewolucji. Biologiczne "konstrukcje", które nie mają sensu nie przetrwałyby do dziś. [[history](#)]

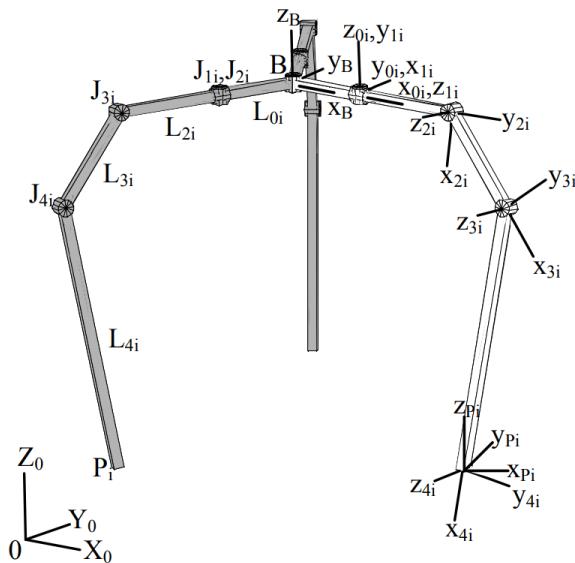
Co natomiast z konstrukcjami robotów trójnożnych? Można się zastanowić czy konstrukcje takie nie powstają ponieważ faktycznie nie mają sensu, czy może dlatego że temat bardziej "klasycznych", prostszych w implementacji, konstrukcji nie został po prostu jeszcze wyyczerpany przez naukowców. Jeżeli rozejrzymy się dookoła siebie możemy zaobserwować wiele przedmiotów codziennego użytku które posiadają właśnie trzy nogi, od wszelakich taboretów, przez wieszaki na ubrania po stoły. Są to jednak przedmioty statyczne i dla takich rozwiązań trzy nogi są wymaganym minimum aby dany przedmiot stał stabilnie. Co jednak z konstrukcjami dynamicznymi? Jeżeli robot trójnożny podniesie nogę, straci stabilność, zacznie się przewracać. Czyni to z niego dość ciekawą konstrukcję, gdzie w momencie stania w miejscu zachowuje się bardziej jak roboty o większej ilości nóg, nie przewraca się, a podczas ruchu zachowuje się jak roboty dwunożne, musi odpowiednio szybko odstawić nogę w odpowiednie miejsce aby się nie przewrócić.

## 1.2. Istniejące konstrukcje trójnożne

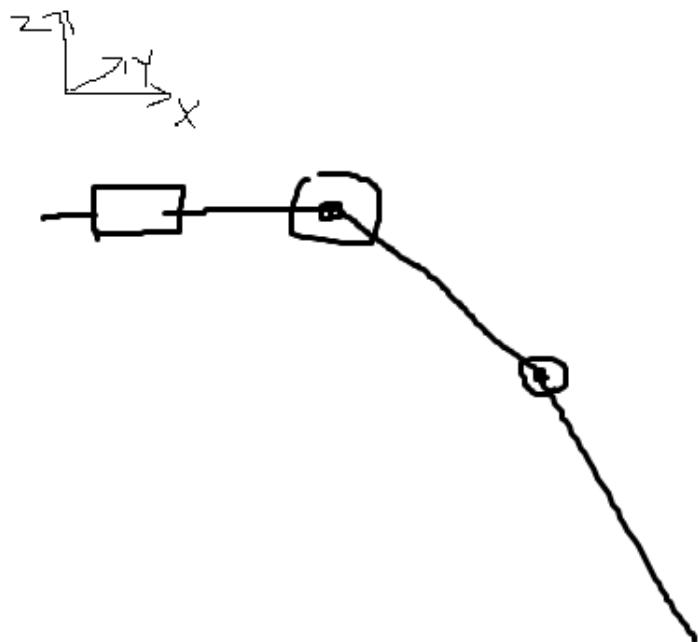
Konstrukcje trójnożne pojawiały się w dziełach science fiction już od dawna, od "The War of the Worlds" z 1898 aż po "Mroczne Widmo" z 2001 i regularnie pojawiają się naukowcy, którzy próbują udowodnić że nie trzeba oganiczać tego typu robotów do dzieł z obszaru science fiction.

### 1.2.1. STriDER

Został zbudowany w 2007 na Uniwersytecie Stanowym w Virginii. Jego celem były eksperymenty z algorytmami chodu i doclewo, prowadzenie obserwacji. Miały to umożliwić długie nogi, które znacznie podwyższały konstrukcję i sprawiały że górną platformą była idealna do instalowania wszelkiego rodzaju urządzeń typu kamery. Kinematykę robota można określić jako  $3 - SRRR$ , a kinematykę jednej nogi jako  $RRR$ . Przy czym "pierwsze"  $R$  oznacza obrót dookoła osi  $x$  a dwa kolejne  $R$  oznaczają obrót wokół osi  $y$  (zgodnie z oznaczeniami na rysunku 1.2) [[strider](#)]



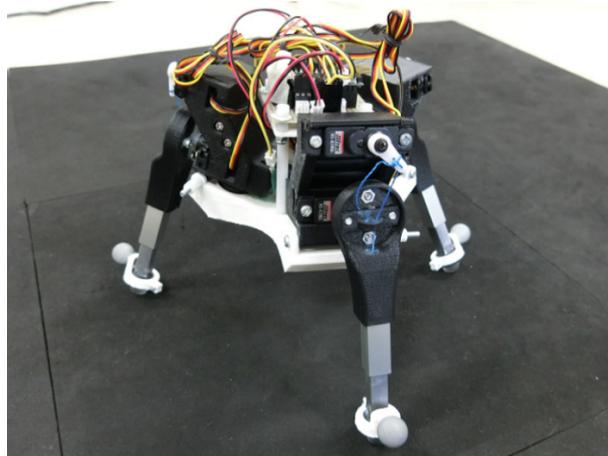
Rys. 1.1. Schemat kinematyczny robota Strider źródło: [strider]



Rys. 1.2. Uproszczony model matematyczny nogi robota STriDER

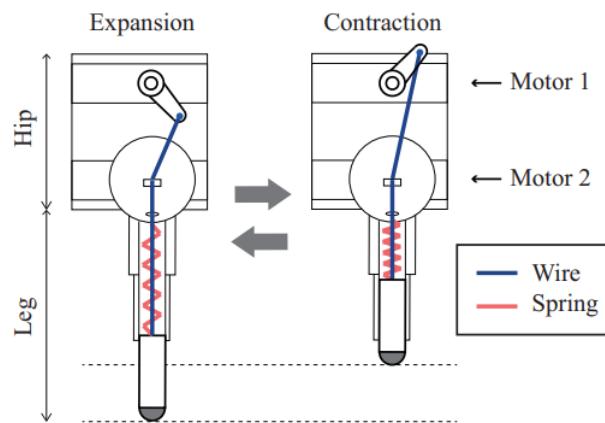
### 1.2.2. Triped Martian

Jest to robot zbudowany przez Yoichi Masudę na uniwersytecie w Osace w 2017 roku. Kinematyka tego robota opiera się na mechanizmie SLIP (Spring-Loaded-Inverted Pendulum). SLIP ma w pewien sposób symulować sposób poruszania się stosowany przez ludzi i zwierzęta. Sprężyna wewnętrz nogi robota jest naciągana, co powoduje skracanie się nogi, a zwalnianie sprężyny z powrotem wydłuża człon. Dodatkowo dodane jest serwo, które może obracać nogę



Rys. 1.3. Zdjęcie robota Martian źródło: [Triped\_Martian]

dookoła osi  $y$ . Czyni to z nogi robota mechanizm o kinematyce typu  $RL$ . Został także dodany czujnik naprężenia, który jest w stanie zmierzyć siłę naciągu nici kompresującej sprężynę. [Triped\_Martian]



Rys. 1.4. Model nogi robota Triped Martian źródło: [Triped\_Martian]

### 1.2.3. Inne konstrukcje

W internecie można także znaleźć kilka różnych, niezbyt dobrze udokumentowanych konstrukcji zakończonych mniejszym lub większym sukcesem:

- Makerfaire 3-legged walking robot
- Missel tripod robot

Niestety konstrukcje te nie posiadają żadnej dokumentacji technicznej i nie da się dokładnie określić zasad ich działania. Nie można nawet mieć pewności że konstrukcje te faktycznie

istnieją a nie są oszustwem lub inną formą naginania rzeczywistości, na co mogłyby wskazywać jakość filmików i uboga ilość materiałów.



## 2. Model Matematyczny

### 2.1. Noga robota typu RRR

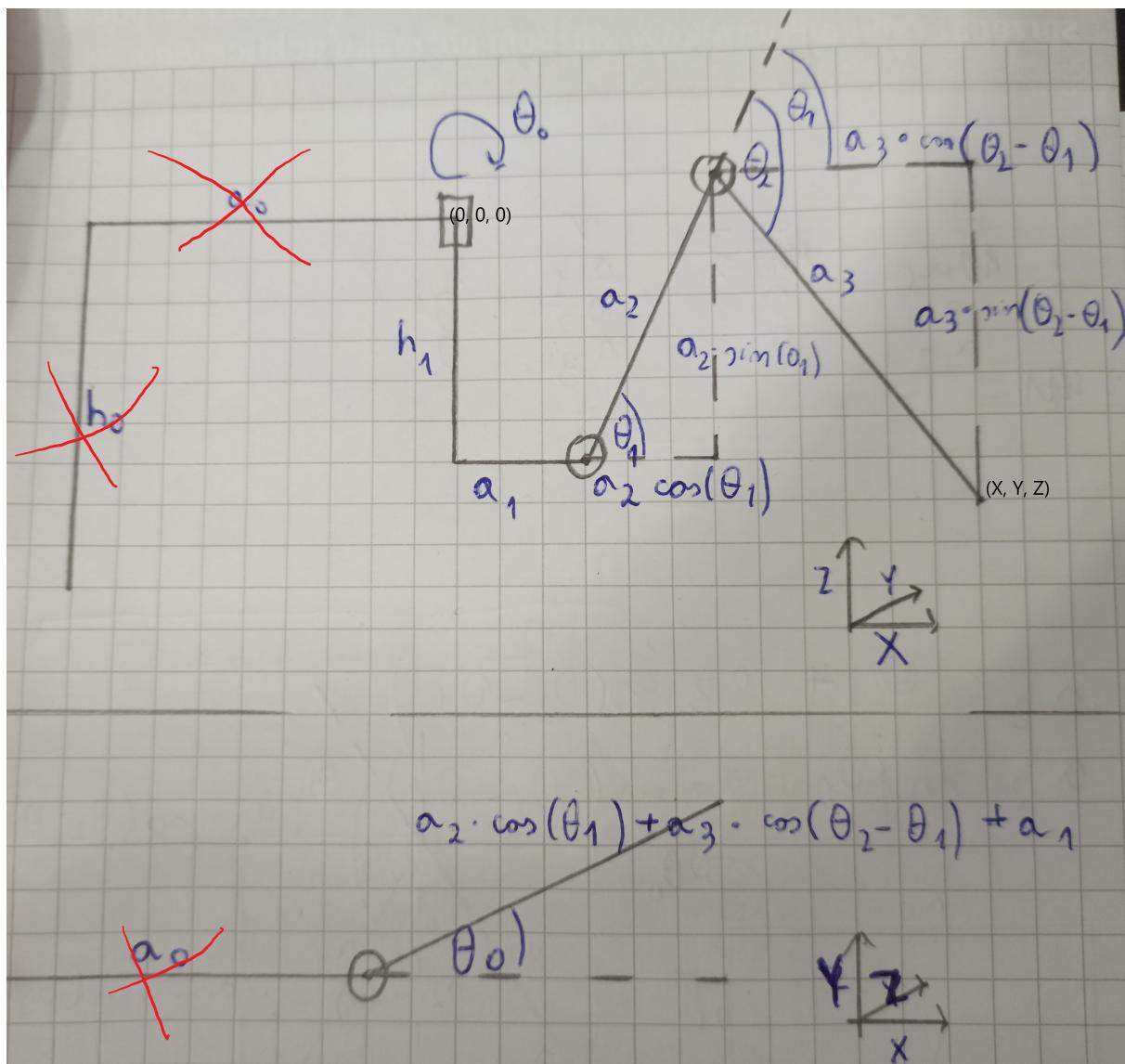
Noga ma trzy stopnie swobody. Wszystkie są typu obrotowego, co czyni z niej konstrukcję typu *RRR*. Przy czym dwa elementy rotacyjne obracają się dookoła osi poziomej (*X*) (podobnie jak w przypadku robota STriDER), a jedna dookoła osi pionowej (*Z*). Są to też te same osi obrotu co w przypadku ramienia robotycznego typu antromorficznego (zwanego także "angular" bądź "jointed"). Co za tym idzie, cały model matematyczny jest w zasadzie identyczny jak w przypadku manipulatora tego typu.

#### 2.1.1. Kinematyka Prosta

Obliczanie kinematyki prostej polega na uzyskaniu równań końcówki ramienia robotycznego we współrzędnych kartezjańskich względem współrzędnych konfiguracyjnych. Inaczej mówiąc, wejściem algorytmu jest zbiór współrzędnych konfiguracyjnych, a na wyjściu otrzymamy współrzędne kartezjańskie.**[robot\_manipulators]**

W przypadku tego konkretnego manipulatora, mamy do czynienia z trójwymiarowym układem współrzędnych kartezjańskich i trzema stopniami swobody. Da to liniowo niezależny układ trzech równań, w którym parametrami będą kąty na jakich mają ustawić się serwomechanizmy a wynikiem wektor współrzędnych kartezjańskich.

Najprostszą metodą liczenia kinematyki prostej jest rozrysowanie modelu matematycznego i geometryczne wyprowadzenie potrzebnych równań. Model taki dla tego manipulatora został przedstawiony na rysunku 2.1. Przyjęty początek układu współrzędnych został oznaczony jako  $(0, 0, 0)$ , a punkt którego współrzędne kartezjańskie są poszukiwane został oznaczony jako  $(X, Y, Z)$ .  $h_1$  i  $a_{1-3}$  to stałe długości poszczególnych członów nogi. Natomiast  $\theta_{0-2}$  to właśnie pozycje serwomechanizmów - parametry algorytmu. Na ich podstawie zostaną obliczone współrzędne końcówki manipulatora w systemie kartezjańskim. Same obliczenia



Rys. 2.1. Model Matematyczny

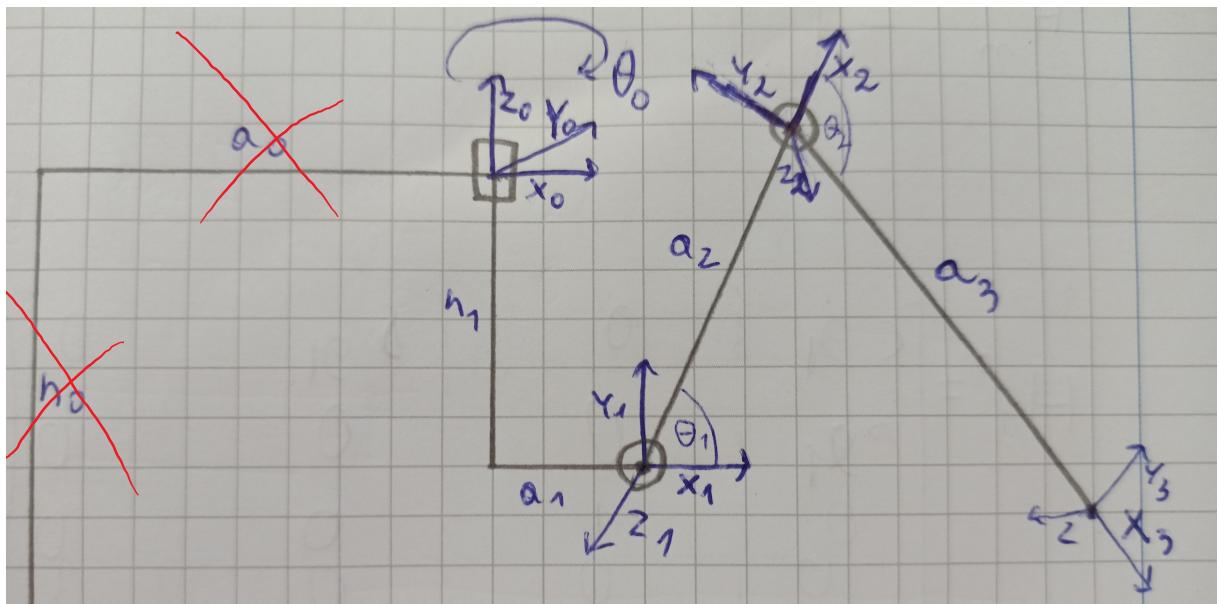
geometryczne są już w tym momencie dość trywialne, wystarczy do każdego członu  $a_{1-3}$  przenieść jego długość na osią  $X$ ,  $Y$  lub  $Z$  za pomocą trygonometrii (sinus lub cosinus) i zsumować odpowiednie długości. Da to układ równań 2.1.

$$\begin{aligned}
 a_{temp} &= a_2 \cos \theta_1 + a_3 \cos (\theta_2 - \theta_1) + a_1 \\
 Y &= a_{temp} \cdot \sin \theta_0 \\
 X &= a_{temp} \cdot \cos \theta_0 \\
 Z &= a_2 \sin \theta_1 - a_3 \sin (\theta_2 - \theta_1)
 \end{aligned} \tag{2.1}$$

### 2.1.2. Kinematyka prosta - metoda Denavita Hartenberga [DH\_AA\_article]

Alternatywą dla zwykłych obliczeń geometrycznych jest metoda Denavita Hartenberga. Polega ona na przedstawieniu całkowitego przekształcenia jako iloczynu przekształceń jednorodnych kolejnych członów. Pojedyncze przekształcenie jednorodne składa się natomiast z 6 przekształceń prostych (3 dla rotacji i 3 dla przesunięć). Wymnożenie tych przekształceń prostych da przekształcenie jednorodne. [DH\_wpaszke\_wyklad]

Metoda ta jest w szczególności użyteczna dla bardzo skomplikowanych manipulatorów, gdzie stopni swobody jest znacznie więcej niż ilość współrzędnych kartezjańskich.



Rys. 2.2. Model Denavit Hartenberg

Z praktycznego punktu widzenia obliczenia należy zacząć od stworzenia specjalnego rysunku (Rys. 2.2) z zaznaczonymi kolejnymi obrotami lokalnych układów współrzędnych, a następnie zebrać odpowiednie transformacje do tabelki (Tab. 2.1)

Gdzie:

$\theta_i$  - angle from  $x_{n-1}$  to  $x_n$  around  $z_{n-1}$

$\alpha_i$  - angle from  $z_{n-1}$  to  $z_n$  around  $x_n$

$r_i$  - distance between the origin of the  $n - 1$  frame and the origin of the  $n$  frame along the  $x_n$  direction.

Joint $i$	$\theta_i$	$\alpha_i$	$r_i$	$d_i$
1	$\theta_0$	$\frac{\pi}{2}$	$a_1$	$h_1$
2	$\theta_1$	0	$a_2$	0
3	$\theta_2$	0	$a_3$	0

**Tabela 2.1.** Tabela z parametrami DH

$d_i$  - distance from  $x_{n-1}$  to  $x_n$  along the  $z_{n-1}$  direction

Następnie macierze transformacji jednorodnej (pomiędzy ramkami  $n-1$  i  $n$ ) oblicza się zgodnie ze wzorem 2.2. Wzór ten jest właśnie obliczony poprzez wymnożenie wzorów na wyżej wspomniane 6 przekształceń prostych. Ale w zasadzie wzór ten sprowadza się do 2 zasadniczych elementów.  $R$  jest macierzą Rotacji o rozmiarze  $3 \times 3$  a  $T$  to macierz Transformacji  $1 \times 3$ .

$$H_n^{n-1} = \left[ \begin{array}{c|c} R & T \\ \hline 0 & 0 & 0 & 1 \end{array} \right] = \left[ \begin{array}{cccc} \cos \theta_i & -\sin \theta_i \cdot \cos \alpha_i & \sin \theta_i \cdot \sin \alpha_i & a_i \cdot \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cdot \cos \alpha_i & -\cos \theta_i \cdot \sin \alpha_i & a_i \cdot \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{array} \right] \quad (2.2)$$

#### [DH\_matrix\_AA\_article]

Następnie należy podstawić wartości dla każdego z 3 punktów i otrzymane macierze wymnożyć zgodnie ze wzorem 2.3.

$$H_n^{n-1} = H_1^0 \cdot H_2^1 \cdot H_3^2 =$$

$$\left[ \begin{array}{cccc} c\theta_0 \cdot (c\theta_1 \cdot c\theta_2 - s\theta_1 \cdot s\theta_2) & -c\theta_0 \cdot (c\theta_1 \cdot s\theta_2 + s\theta_1 \cdot c\theta_2) & s\theta_0 & c\theta_0 \cdot (a_1 + a_2 \cdot c\theta_1 + a_3 \cdot (c\theta_1 \cdot c\theta_2 - s\theta_1 \cdot s\theta_2)) \\ s\theta_0 \cdot (c\theta_1 \cdot c\theta_2 - s\theta_1 \cdot s\theta_2) & -s\theta_0 \cdot (c\theta_1 \cdot s\theta_2 + s\theta_1 \cdot c\theta_2) & -c\theta_0 & s\theta_0 \cdot (a_1 + a_2 \cdot c\theta_1 + a_3 \cdot (c\theta_1 \cdot c\theta_2 - s\theta_1 \cdot s\theta_2)) \\ c\theta_1 \cdot s\theta_2 + s\theta_1 \cdot c\theta_2 & c\theta_1 \cdot c\theta_2 - s\theta_1 \cdot s\theta_2 & 0 & h_1 + a_2 \cdot s\theta_1 + a_3 \cdot (c\theta_1 \cdot s\theta_2 + s\theta_1 \cdot c\theta_2) \\ 0 & 0 & 0 & 1 \end{array} \right]$$

$$\left[ \begin{array}{cccc} c\theta_0 \cdot c(\theta_1 + \theta_2) & -c\theta_0 \cdot s(\theta_1 + \theta_2) & s\theta_0 & c\theta_0 \cdot (a_1 + a_2 \cdot c\theta_1 + a_3 \cdot c(\theta_1 + \theta_2)) \\ s\theta_0 \cdot c(\theta_1 + \theta_2) & -s\theta_0 \cdot s(\theta_1 + \theta_2) & -c\theta_0 & s\theta_0 \cdot (a_1 + a_2 \cdot c\theta_1 + a_3 \cdot c(\theta_1 + \theta_2)) \\ c\theta_1 \cdot s\theta_2 + s\theta_1 \cdot c\theta_2 & c(\theta_1 + \theta_2) & 0 & h_1 + a_2 \cdot s\theta_1 + a_3 \cdot s(\theta_1 + \theta_2) \\ 0 & 0 & 0 & 1 \end{array} \right] \quad (2.3)$$

Następnie, aby otrzymać właściwe równania, należy z równania 2.3 wziąć część odpowiedzialną za transformację. Efektem jest wzór 2.4, czyli finalna wersja równań kinematyki

prostej obliczonej metodą Denavita Hartenberga

$$\begin{aligned} X &= c\theta_0 \cdot (a_1 + a_2 \cdot c\theta_1 + a_3 \cdot c(\theta_1 + \theta_2)) \\ Y &= s\theta_0 \cdot (a_1 + a_2 \cdot c\theta_1 + a_3 \cdot c(\theta_1 + \theta_2)) \\ Z &= h_1 + a_2 \cdot s\theta_1 + a_3 \cdot s(\theta_1 + \theta_2) \end{aligned} \quad (2.4)$$

TODO - znak się nie zgadza

### 2.1.3. Invert kinematics

Kinematyka odwrotna to TODOTODOTODO

Zwykle odwrotną kinematykę można obliczyć poprzez rozwiązywanie równań kinematyki prostej. Jest to w tym przypadku także teoretycznie możliwe, jako że mamy do czynienia z trzema niewiadomymi ( $\theta_{0-2}$ ) i układem trzech równań nr. 2.1 lub 2.4. (Równanie na  $a_{temp}$  jest jedynie pomocnicze, trzeba je traktować jak część równań  $Y$  i  $X$ ).

Jest to jak najbardziej wykonalne w przypadku  $\theta_0$ , można to zrobić łącząc wzór na  $X$  i  $Y$ , dzieląc go obustronnie, zamieniając  $\frac{\sin}{\cos}$  na tan i wyciągając  $\theta_0$  na lewą stronę. Daje to wzór 2.5

$$\theta_0 = \arctan \frac{Y}{X} \quad (2.5)$$

Większy problem pojawia się jednak w przypadku obliczania  $\theta_1$  i  $\theta_2$ , ponieważ wartości te da się obliczyć z wyżej wspomnianego równania, ale nie da się wyznaczyć na te wartości równań w postaci jawnej. A bez postaci jawnej poprawna implementacja tych równań będzie znacznie utrudniona.

Można natomiast zastosować pewnego rodzaju uproszczenie. Jeżeli wyznaczanie równań  $\theta_1$  i  $\theta_2$  sprowadzi się do problemu dwuwymiarowego, obliczenia stają się w zasadzie identyczne jak w przypadku obliczeń kinematyki odwrotnej dla robota typu SCARA, co było już robione wielokrotnie.

Aby obliczyć kąt  $\theta_2$  można powrócić do rysunku 2.1 i zastosować na trójkącie utworzonym z  $a_2$  i  $a_3$  twierdzenie cosinusów połączone z twierdzeniem pitagorasa. Daje to wzór 2.6.

$$(x - a_1)^2 + (z + h_1)^2 = a_2^2 + a_3^2 - 2 \cdot a_2 \cdot a_3 \cdot \cos(180^\circ - \theta_2) \quad (2.6)$$

Następnie na podstawie równania 2.6 aby ostatecznie otrzymać  $\theta_2$  należy zastosować wzór  $\cos(180^\circ - \theta) = -\cos \theta$  i za pomocą prostych przekształceń wyciągnąć z równania  $\theta_2$ . Daje to wzór 2.7

$$\theta_2 = \arccos \left( \frac{(x - a_1)^2 + (z + h_1)^2 - a_2^2 - a_3^2}{2a_2a_3} \right) \quad (2.7)$$

[SCARA\_model] Natomiast obliczenia dla  $\theta_1$  TODO

$$\theta_1 = \arctan \frac{z + h_1}{x - a_1} + \arcsin \left( \frac{a_3 \sin \theta_2}{\sqrt{(x - a_1)^2 + (z + h_1)^2}} \right) \quad (2.8)$$

[SCARA\_model]

Ostatecznie zbierając równania 2.5, 2.8, 2.7 otrzymujemy układ równań który stanowi kinematykę odwrotną opisywanej nogi robotycznej.

TODO - układ równań

TODO - ograniczenia.

Natomiast trzeba cały czas pamiętać że jest to tylko uproszczenie, które będzie owocować pewnymi drobnymi błędami. Trzeba zauważyc, że zmiana współrzędnej  $Y$  nie wpływa na wartości kątów  $\theta_1$  i  $\theta_2$ , tylko na  $\theta_0$ . Oznacza to, że wraz ze zmianą kąta  $\theta_0$ , na potrzeby uproszczenia obliczeń, obraca się także o ten kąt oś  $X$  układu współrzędnych. W praktyce oznacza to, że odsunięcie nogi od punktu  $(0, 0, 0)$  zawsze będzie rzutowane na cylinder o promieniu równym odsunięciu o takim samym  $X$  i  $Z$ , ale  $Y = 0$ . Problem jest zaprezentowany na rysunku XXX. Jednakże błąd powinien być pomijalnie mały a równania powinny umożliwić skuteczną implementację algorytmu chodu.

## 2.2. Cały Robot

Zwykle konstruując roboty wzorujemy się na zwierzętach występujących w naturze. W tym przypadku nie mamy jednak tego luksusu, dlatego na początek należy przyjąć jakieś najbardziej intuicyjne założenie. Dlatego też uznałem że najlepszym rozwiązaniem będzie rozmieścić nogi na jednej płaszczyźnie w równych odstępach - co 120 deg.

W poprzednim podrozdziale przyjąłem osie układu relatywne do ułożenia początkowego nogi robota. Oznacza to że robot trójnożny będzie miał 3 niezależne osie  $x$  i 3 niezależne osie  $y$ , tylko oś  $z$  zgadza się między kolejnymi nogami. Idzie za tym konieczność stworzenia pewnej metody "obracania" wszystkich tych osi  $x$  i  $y$  do jednego zunifikowanego układu współrzędnych. Pozwoli to obliczyć kinematykę całego robota.

### 2.2.1. Matematyka kroku

W celu uproszczenia zarówno obliczeń jak i późniejszej generacji kolejnych kroków algorytmu chodu, można pominąć liczenie całej kinematyki prostej i odwrotnej całego robota.

Zamiast tego wystarczy pojedynczy krok odpowiednio sparametryzować. Jeżeli dla każdego kroku pojedynczej nogi przyjmiemy:

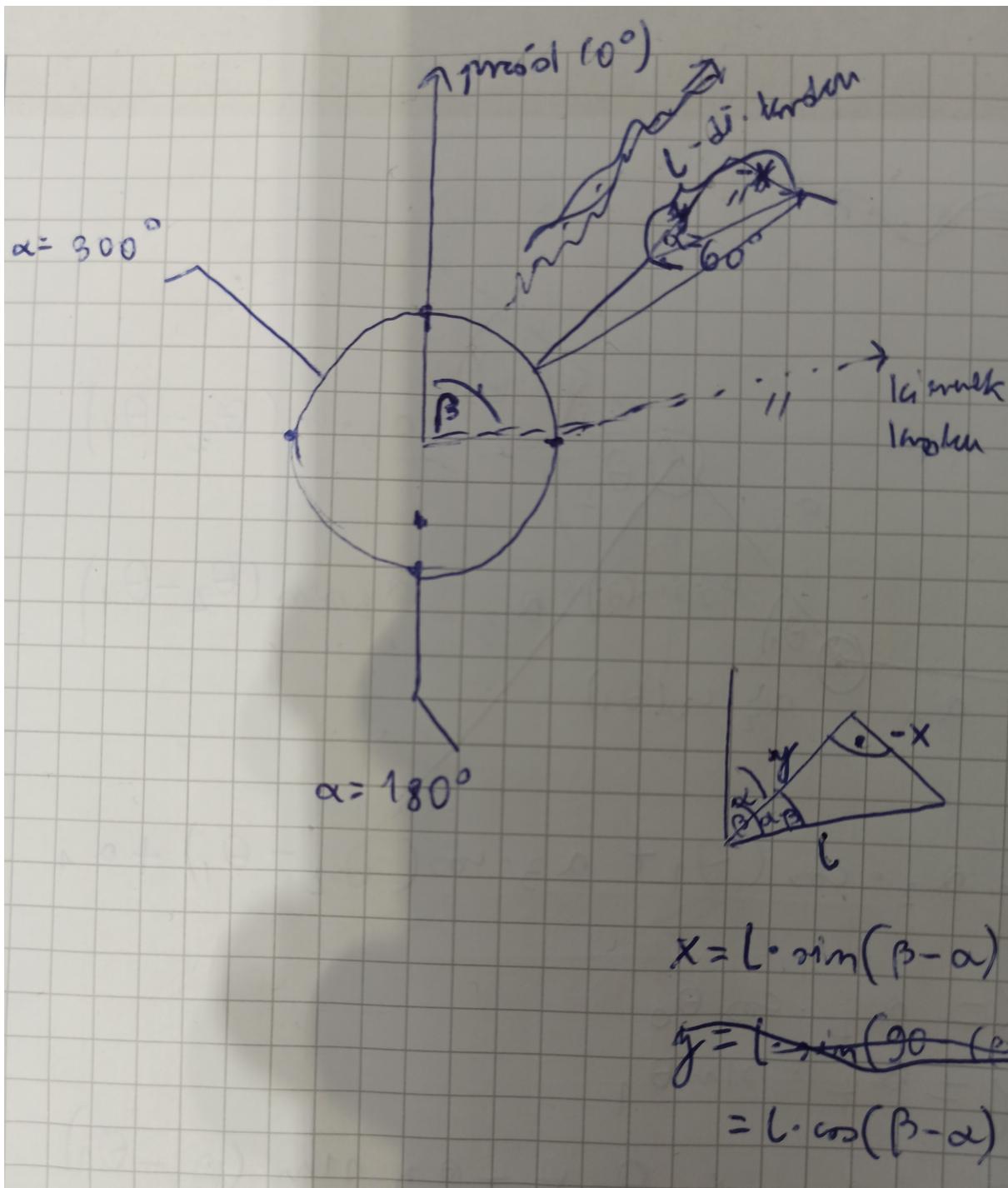
- kąta  $\alpha$  na "tarczy" robota, na którym ustawiona jest nogi
- kąta  $\beta$  względem "przodu" robota, w którą ma zostać wykonany krok
- długość kroku  $l$ , równolegle do osi kroku robota

To możemy łatwo otrzymać algorytm który z tych dwóch zmiennych i jednej stałej da pozycję zmianę pozycji końcówki robota ( $\Delta x, \Delta y$ ) we współrzędnych kartezjańskich względem nogi robota.

Zostało to przedstawione na rysunku 2.3 na przykładzie nogi na pozycji  $\alpha = 60^\circ$ . Na wspomnianym rysunku widać przód robota oznaczony jako  $0^\circ$ , kierunek kroku odsunięty o kąt  $\beta$  od osi "przodu" robota i wynikające z tego kroku przestawienie nogi. Przestawienie to składa się z długości korku  $l$  i dwóch pozycji nogi - przed krokiem i po kroku. Wynikające z tego zależności geometryczne zostały wyizolowane na [TODO - dać osobny obrazek na zależności]. Na podstawie tych zależności można napisać równanie 2.9

$$\begin{aligned}\Delta x &= l \cdot \sin(\beta - \alpha) \\ \Delta y &= l \cdot \cos(\beta - \alpha)\end{aligned}\tag{2.9}$$

Równanie 2.9 mówi tyle, że wykonanie nogą na pozycji  $\alpha$  kroku o długości  $l$  w kierunku  $\beta$  wymusza przestawienie końcówki nogi o  $(\Delta x, \Delta y)$



Rys. 2.3. Schemat matematyczny wykonywania kroku