Projekt robota trójnożnego

Jakub Mazur

26 października 2022

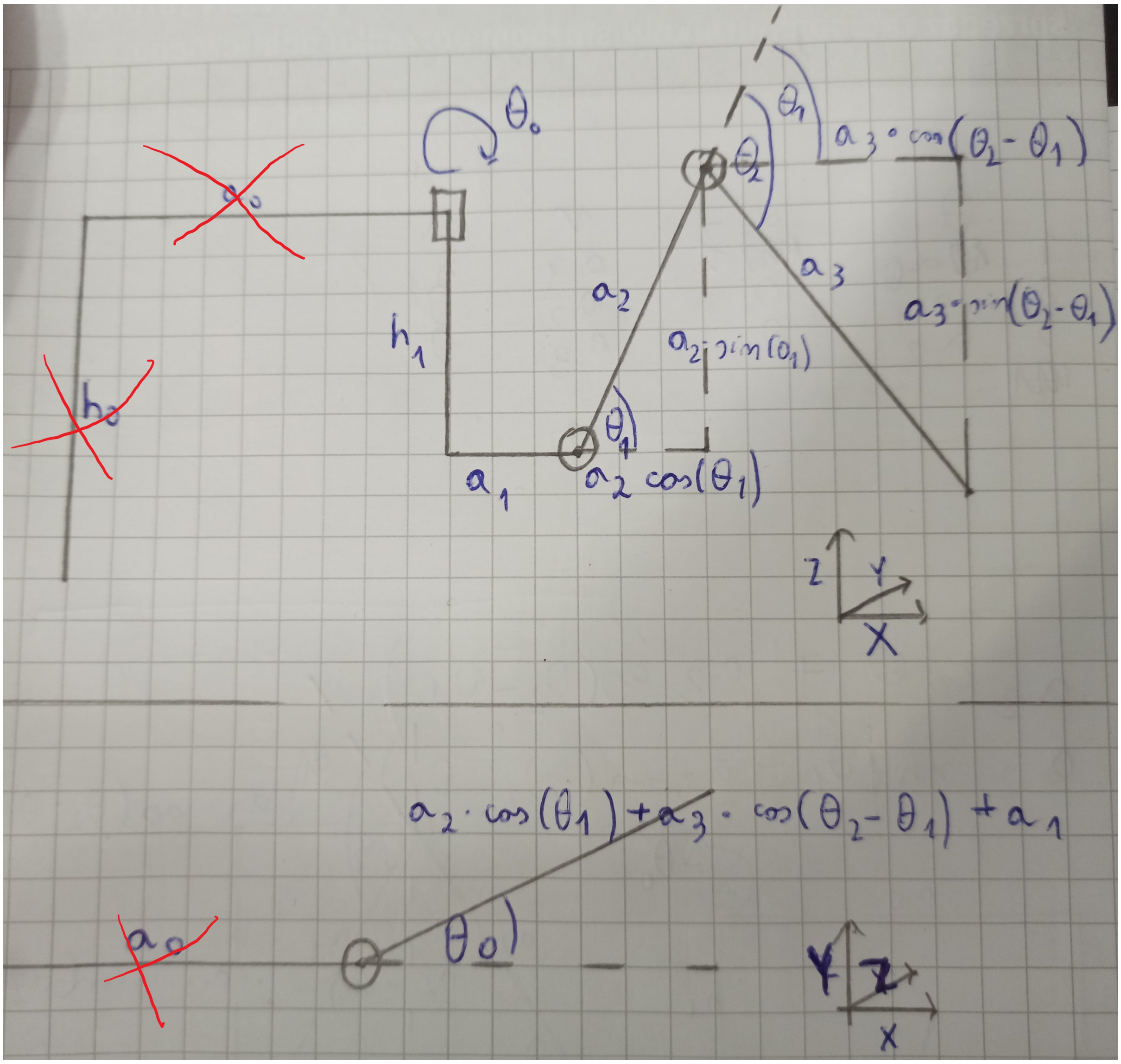
Spis treści

1 Model Matematyczny

Noga ma 3 stopnie swobody. Wszystkie są typu obrotowego, przy czym dwie obracają się dookoła osi poziomej, a jedna dookoła osi pionowej. Są to te same osi obrotu co w przypadku ramienia robotycznego typu antromorficznego.

1.1 Noga robota typu RRR

Jest to noga o 3 rotacyjnych stopniach swobody, przy czym 1 stopień obraca się dookoła osi z, a 2 stopnie dookoła osi y. Model matematyczny takiej nogi, a co za tym idzie także jej kinematyka, będzie identyczny jak model ramienia robotycznego typu antropomorficznego (zwanego także "angular" bądź "jointed").



Rysunek 1: Model Matematyczny

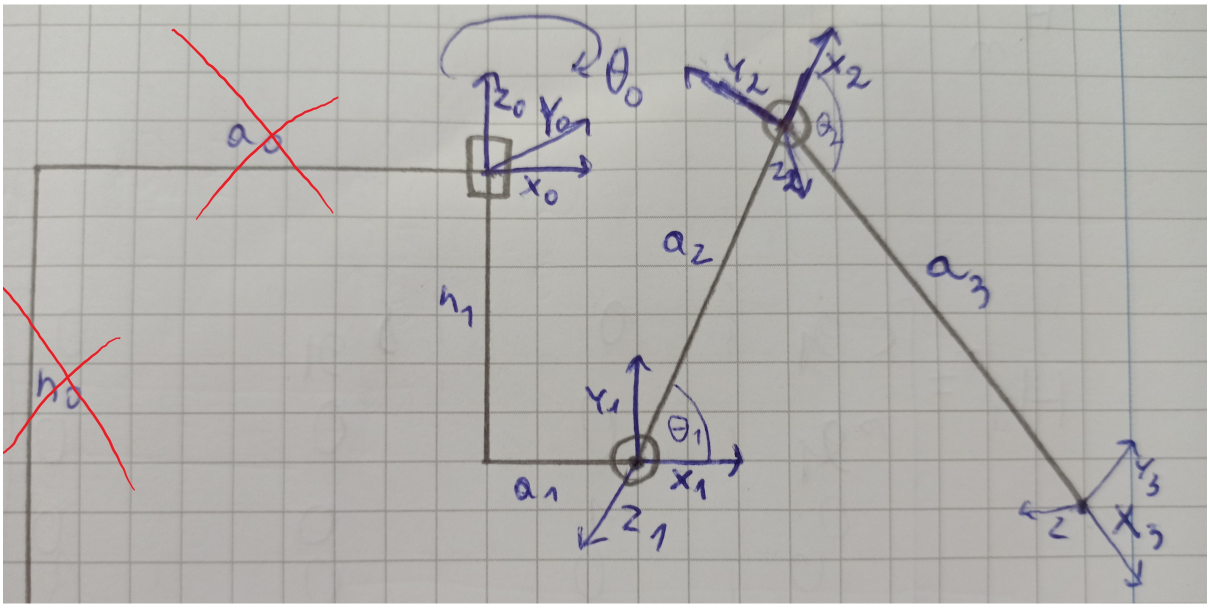
1.1.1 Forward kinematics

Obliczanie kinematyki prostej polega na TODOTODO

$$
\begin{aligned}
a_{temp} &= a_2 \cos \theta_1 + a_3 \cos(\theta_2 - \theta_1) + a_1 \\
Y &= a_{temp} \cdot \sin \theta_0 \\
X &= a_{temp} \cdot \cos \theta_0 \\
Z &= a_2 \sin \theta_1 - a_3 \sin(\theta_2 - \theta_1)
\end{aligned}
\tag{1}
$$

1.1.2 Forward kinematics - denavit hartenberg [1]

Alternatywą dla zwykłych obliczeń geometrycznych jest metoda Denvaita Hartenberga - TODOTODO-TODO



Rysunek 2: Model Denavit Hartenberg

$$\begin{array}{c|cccc} \text{Joint } i & \theta_i & \alpha_i & r_i & d_i \\ \hline 1 & \theta_0 & \frac{\pi}{2} & a_1 & h_1 \\ 2 & \theta_1 & 0 & a_2 & 0 \\ 3 & \theta_2 & 0 & a_3 & 0 \end{array} \tag{2}$$

Gdzie:
θ_i - angle from x_{n-1} to x_n around z_{n-1}
α_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.
d_i - distance from x_{n-1} to x_n along the z_{n-1} direction

Następnie macierze przejść pomiędzy ramkami $n-1$ i n oblicza się zgodnie z następującym wzorem:

$$H_n^{n-1} = \left[\begin{array}{ccc|c} & \text{R} & & \text{T} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cdot\cos\alpha_i & \sin\theta_i\cdot\sin\alpha_i & a_i\cdot\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cdot\cos\alpha_i & -\cos\theta_i\cdot\sin\alpha_i & a_i\cdot\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} [2] \tag{3}$$

Gdzie R oznacza macierz Rotacji a T macierz Transformacji.

A macierz przejścia pomiędzy ramką 0 a ramką 3 zgodnie z następującym wzorem:

$$H_n^{n-1} = H_1^0\cdot H_2^1\cdot H_3^2 =$$
$$\begin{bmatrix} c\theta_0\cdot(c\theta_1\cdot c\theta_2 - s\theta_1\cdot s\theta_2) & -c\theta_0\cdot(c\theta_1\cdot s\theta_2 + s\theta_1\cdot c\theta_2) & s\theta_0 & c\theta_0\cdot(a_1 + a_2\cdot c\theta_1 + a_3\cdot(c\theta_1\cdot c\theta_2 - s\theta_1\cdot s\theta_2)) \\ s\theta_0\cdot(c\theta_1\cdot c\theta_2 - s\theta_1\cdot s\theta_2) & -s\theta_0\cdot(c\theta_1\cdot s\theta_2 + s\theta_1\cdot c\theta_2) & -c\theta_0 & s\theta_0\cdot(a_1 + a_2\cdot c\theta_1 + a_3\cdot(c\theta_1\cdot c\theta_2 - s\theta_1\cdot s\theta_2)) \\ c\theta_1\cdot s\theta_2 + s\theta_1\cdot c\theta_2 & c\theta_1\cdot c\theta_2 - s\theta_1\cdot s\theta_2 & 0 & h_1 + a_2\cdot s\theta_1 + a_3\cdot(c\theta_1\cdot s\theta_2 + s\theta_1\cdot c\theta_2) \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$
$$\begin{bmatrix} c\theta_0\cdot c(\theta_1+\theta_2) & -c\theta_0\cdot s(\theta_1+\theta_2) & s\theta_0 & c\theta_0\cdot(a_1 + a_2\cdot c\theta_1 + a_3\cdot c(\theta_1+\theta_2)) \\ s\theta_0\cdot c(\theta_1+\theta_2) & -s\theta_0\cdot s(\theta_1+\theta_2) & -c\theta_0 & s\theta_0\cdot(a_1 + a_2\cdot c\theta_1 + a_3\cdot c(\theta_1+\theta_2)) \\ c\theta_1\cdot s\theta_2 + s\theta_1\cdot c\theta_2 & c(\theta_1+\theta_2) & 0 & h_1 + a_2\cdot s\theta_1 + a_3\cdot s(\theta_1+\theta_2) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

Następnie, aby otrzymać właściwe równania, należy z równania 4 wziąć część odpowiedzialną za transformację.

$$\begin{aligned} X &= c\theta_0 \cdot (a_1 + a_2 \cdot c\theta_1 + a_3 \cdot c(\theta_1 + \theta_2)) \\ Y &= s\theta_0 \cdot (a_1 + a_2 \cdot c\theta_1 + a_3 \cdot c(\theta_1 + \theta_2)) \\ Z &= h_1 + a_2 \cdot s\theta_1 + a_3 \cdot s(\theta_1 + \theta_2) \end{aligned} \tag{5}$$

TODO - znak się nie zgadza

1.1.3 Invert kinematics

Kinematyka odwrotna to TODOTODOTODO

Zwykle odwrotną kinematykę można obliczyć poprzez rozwiązanie równań kinematyki prostej. Jest to w tym przypadku także teoretycznie możliwe, jako że mamy do czynienia z trzema niewiadomymi (θ_{0-2}) i układem trzech równań nr. 1 lub 1. (Równanie na a_{temp} jest jedynie pomocnicze, trzeba je traktować jak część równań Y i X).

Jest to jak najbardziej wykonalne w przypadku θ_0, można to zrobić łącząc wzór na X i Y, dzieląc go obustronnie, zamieniając $\frac{\sin}{\cos}$ na tan i wyciągając θ_0 na lewą stronę. Daje to wzór 6

$$\theta_0 = \arctan \frac{Y}{X} \tag{6}$$

Większy problem pojawia się jednak w przypadku obliczania θ_1 i θ_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 nie będzie możliwa poprawna implementacja tych równań.

Można natomiast zastosować pewnego rodzaju trik. Jeżeli wyznaczanie równań θ_1 i θ_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 θ_2 można powrócić do rysunku 2 i zastosować na trójkącie utworzonym z a_2 i a_3 twierdzenie cosinusów połączone z twierdzeniem pitagorasa. Daje to wzór 7.

$$(x - a_1)^2 + (z - h_1)^2 = a_2^2 + a_3^2 - 2 \cdot a_2 \cdot a_3 \cdot \cos(180^o - \theta_2) \tag{7}$$

Następnie na podstawie równania 7 aby ostatecznie otrzymać θ_2 należy zastosować wzór $\cos(180^o - \theta) = -\cos\theta$ i za pomocą prostych przekształceń wyciągnąć z równania θ_2. Daje to wzór 8

$$\theta_2 = \arccos\left(\frac{(x - a_1)^2 + (z - h_1)^2 - a_2^2 - a_3^2}{2a_2a_3}\right)[4] \tag{8}$$

Natomiast obliczenia dla θ_1 TODO

$$\theta_1 = \arctan\frac{z - h_1}{x - a_1} - \arcsin\left(\frac{a_3 \sin\theta_2}{\sqrt{(x - a_1)^2 + (z - h_1)^2}}\right)[4] \tag{9}$$

Ostatecznie zbierając równania 6, **??**, 8 otrzymujemy układ równań który stanowi kinematykę odwrotną opisywanej nogi robotycznej.

TODO - układ równań

TODO - ograniczenia.

1.2 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.

1.2.1 Matematyka kroku

Aby móc w prosty sposób napisać później algorytmy chodu i móc się skupić na kolejności przestawiania nóg przez robota, długości i wysokości kroku i innych tego typu aspektach, należy teraz dobrze ten krok sparametryzować. Na podstawie moich poprzednich doświadczeń z podobnymi projektami, uważam że najbardziej intuicyjne będzie sparametryzowanie kroku do:

- wartości kąta pomiędzy kierunkiem "do przodu" kroku a osią x danej nogi
- długości kroku
- wysokości kroku

2 Model CAD i druk 3D

3 Środowisko ROS

3.1 Schemat implementacji

4 Algorytm Chodu

Literatura

[1] How to Find Denavit-Hartenberg Parameter Tables, blogpost by Automatic Addison

[2] Homogeneous Transformation Matrices Using Denavit-Hartenberg, blogpost by Automatic Addison

[3] Matlab Denavit Hartenberg calculations

[4] Adam Labuda, Janusz Pomirski, Andrzej Rak (2009) Model manipulatora o dwóch stopniach swobody

[5] Alexander Wallen Kiessling, Niclas Maatta (2020) Anthropomorphic Robot Arm