

HexapodKinematics

Hexapod Bionik

Luty 2024

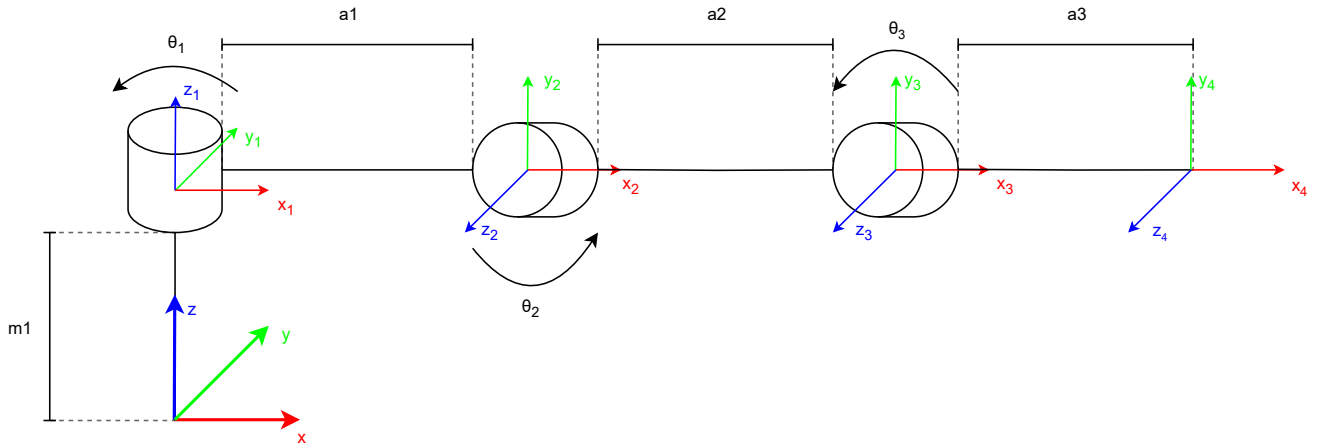


Spis treści

1	Diagram kinematyczny pojedynczego odnóża	3
2	Proste zadanie kinematyki dla odnóża	3
2.1	Macierzowa metoda rozwiązywania prostego zadania kinematyki	4
2.2	Analityczna metoda rozwiązywania prostego zadania kinematyki	4
3	Odwrotne zadanie kinematyki dla odnóża	5
4	Testowanie implementacji w Pythonie	6

1 Diagram kinematyczny pojedynczego odnóża

Rozwiązanie zarówno prostego zadania kinematyki jak i kinematyki odwrotnej dla odnóża oparto na poniższym diagramie kinematycznym, który spełnia wszystkie 4 zasady Denavita-Hartenberga.



Rysunek 1: Diagram kinematyczny odnóża

Jak widać na powyższym diagramie nodze robota występują 3 stawy, których dodatni kierunek ruchu jest zgodny z regułą prawej dłoni. Każdy z nich jest ustawiony w pozycji 0 deg, natomiast zakresy ruchów są następujące:

- J1 - zakres ruchu (−90 deg; 90 deg)
- J2 - zakres ruchu (−90 deg; 90 deg)
- J3 - zakres ruchu (−180 deg; 0 deg)

Na bazie powyższego diagramu możemy wyznaczyć zmodyfikowane parametry Denavita-Hartenberga, które wyglądają następująco:

i	$\alpha_i - 1$	$a_i - 1$	d_i	θ_i
1	0	0	d_1	θ_1
2	$\frac{\pi}{2}$	a_1	0	θ_2
3	0	a_2	0	θ_3
4	0	a_3	0	0

2 Proste zadanie kinematyki dla odnóża

Wykorzystując parametry Denavita-Hartenberga wyznaczone w poprzednim punkcie możemy znaleźć macierze przekształcenia jednorodnego ${}^0_1\mathbf{T}$, ${}^1_2\mathbf{T}$, ${}^2_3\mathbf{T}$, które posłużą nam do ostatecznego wyznaczenia macierzy ${}^3_4\mathbf{T}$. Macierze przekształceń pomiędzy układami wyznaczone są zgodnie z następującym wzorem:

$${}^{n-1}_n\mathbf{T} = \text{Rot}_{z_{n-1}}(\alpha_{n-1}) \cdot \text{Trans}_{z_{n-1}}(a_{n-1}) \cdot \text{Rot}_{z_n}(\theta_n) \cdot \text{Trans}_{z_n}(d_n)$$

Po podstawieniu danych, a następnie wymnożeniu macierzy składowych, ostateczne macierze przekształceń jednorodnych wyglądają następująco:

$${}^0_1\mathbf{T} = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1_2\mathbf{T} = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & a_1 \\ 0 & 0 & -1 & 0 \\ \sin \theta_2 & \cos \theta_2 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2_3\mathbf{T} = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & a_2 \\ \sin \theta_3 & \cos \theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3_4\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & a_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.1 Macierzowa metoda rozwiązywania prostego zadania kinematyki

Po ich wymnożeniu otrzymujemy macierz przekształcenia jednorodnego ${}^0_4\mathbf{T}$, którą możemy wykorzystać do wyznaczenia położenia środka układu współrzędnych 4, będącego jednocześnie środkiem stopy odnóża. Robimy to w następujący sposób:

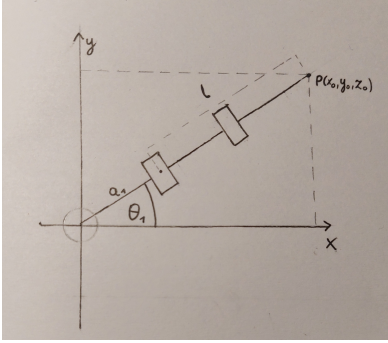
$$P = {}^0_1\mathbf{T} {}^1_2\mathbf{T} {}^2_3\mathbf{T} {}^3_4\mathbf{T} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = {}^0_4\mathbf{T} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

2.2 Analityczna metoda rozwiązywania prostego zadania kinematyki

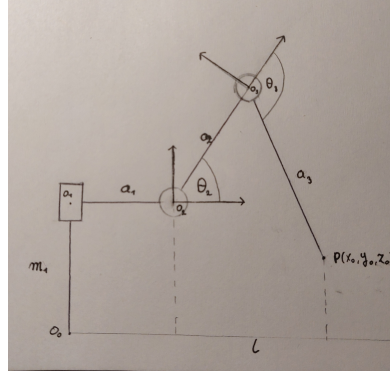
Korzystając z faktu, że współrzędne punktu $P = [p_x, p_y, p_z]^T$, które poszukujemy, są jednocześnie ostatnią kolumną macierzy ${}^0_4\mathbf{T}$, możemy znaleźć symboliczną reprezentację tej kolumny. Zamiast podstawiać dane do macierzy poszczególnych przekształceń, a następnie je wymnażać w celu uzyskania macierzy ${}^0_4\mathbf{T}$ wypełnionej danymi wejściowymi, możemy bezpośrednio wymnożyć te macierze, aby otrzymać symboliczną reprezentację macierzy ${}^0_4\mathbf{T}$. Dzięki temu otrzymujemy gotowe wzory na współrzędne p_x , p_y oraz p_z punktu P , które wyglądają następująco:

$$\begin{aligned} p_x &= \cos \theta_1 (a_1 - a_3 (\sin \theta_2 \sin \theta_3 - \cos \theta_2 \cos \theta_3) + a_2 \cos \theta_2) \\ p_y &= \sin \theta_1 (a_1 - a_3 (\sin \theta_2 \sin \theta_3 - \cos \theta_2 \cos \theta_3) + a_2 \cos \theta_2) \\ p_z &= d_1 + a_3 (\cos \theta_2 \sin \theta_3 + \cos \theta_3 \sin \theta_2) + a_2 \sin \theta_2 \end{aligned}$$

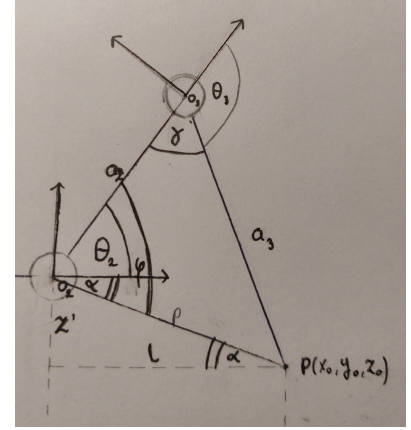
3 Odwrotne zadanie kinematyki dla odnóza



(a) Perspektywa od góry



(b) Przykładowe ustawienie



(c) Przybliżenie

Zakładamy, że szukamy rozwiązania dla punktu $P(x_0, y_0, z_0)$ względem o_0 . Korzystając z widoku od góry modelu zdefiniujemy długość l , jako rzut odcinka $|o_2o_4|$ na płaszczyznę X-Y, z modelu jasno wynika:

$$l = \sqrt{x_0^2 + y_0^2} - a_1$$

Do wyznaczenia zmiennych serwomechanizmów przydatne okaże się znalezienie różnicy wysokości między punktami o_2 i P , dalej będziemy zapisywać ją jako:

$$z' = z_0 - m_1$$

Wszystkie wyznaczone odcinki i długości przedstawiono na rysunkach.

Wyznaczanie θ_1 Przeglądając się modelowi od góry, możemy zauważyć, że składowa z_0 punktu P nie ma wpływu na ustawienie pierwszego serwomechanizmu, dodatkowo nie przewidujemy $x_0 < 0$, oznaczając:

$$\theta_1 = \arctan\left(\frac{y_0}{x_0}\right)$$

co daje nam przedział otrzymywanych wartości $\theta_1 \in [-90^\circ; 90^\circ]$.

Wyznaczanie θ_2 Korzystając z twierdzenia kosinusów dostajemy $\cos \varphi = \frac{p^2 + a_2^2 - a_3^2}{2pa_2}$. Dodatkowo $\tan \alpha = \frac{z'}{l}$. Znając te kąty otrzymujemy $\theta_2 = \varphi + \alpha$ oraz wiemy, że $\theta_2 \in [-90^\circ; 90^\circ]$

Wyznaczanie θ_3 Ponownie wykorzystamy twierdzenie kosinusów dla kąta γ :

$$\cos \gamma = \frac{a_2^2 + a_3^2 - p^2}{2a_2a_3}$$

γ i θ_3 są kątami przyległymi, przez co na koniec otrzymujemy $\theta_3 = \gamma - 180^\circ$ oraz przedział $\theta_3 \in [-180^\circ; 0^\circ]$.

Podsumowanie Ostatecznie wszystkie 3 zmienne serwomechanizmów możemy przedstawić jako:

$$\begin{aligned} \theta_1 &= \arctan \frac{y_0}{x_0} \\ \theta_2 &= \arccos \frac{p^2 + a_2^2 - a_3^2}{2pa_2} + \arctan \frac{z'}{l} \\ \theta_3 &= \arccos \frac{a_2^2 + a_3^2 - p^2}{2a_2a_3} - 180^\circ \end{aligned}$$

4 Testowanie implementacji w Pythonie

Powyżej opisany algorytm zaimplementowano w języku Python. Przygotowano również zestaw testów zabezpieczając przyszłe modyfikacje kodu. Testy sprawdzają działanie w podstawowych, banalnych sytuacjach oraz mniej konwencjonalnych. Sprawdzono działanie programu dla punktów z kilkoma rozwiązaniami odwrotnego zadania kinematycznego, punktów bardzo blisko początku układu współrzędnych i wykraczających poza dostępny zakres. Kod dostosowano, co ma również odzwierciedlenie w testach na scenariusze, gdzie w działaniach matematycznych może pojawić się dzielenie przez zero, czy wykraczanie poza dziedzinę dla odwrotnych funkcji trygonometrycznych.