

Krzysztof Posiewała
Hubert Stachnik
Aleksandra Styk

Kinematyka i symulacja nóg humanoida Poppy

Niniejszy artykuł przedstawia projekt, symulację oraz produkcję niskonakładowego robota antropomorficznego. Inspiracją przedsięwzięcia jest projekt Poppy Humanoid, który został upubliczniony dla celów popularyzacji wizji robotów, jako potężne narzędzie do nauki. W poniższym opracowaniu przeanalizowano i skupiono uwagę na samym problemie konstrukcji dolnej części robota Poppy. Dzięki zastosowaniu pakietu MATLAB Multibody oraz programu Autodesk Inventor, został wykonany model robota humanoidalnego, a dodatkowo wyliczona zarówno kinematyka prosta, jak i odwrotna, co umożliwiło i ułatwiło dalszą pracę nad ukończeniem projektu.

Wprowadzenie

Postęp, który jest zauważalny na przestrzeni ostatnich lat, związany z wytwarzaniem wysoce wyspecjalizowanych robotów, przyczynia się do automatyzacji różnych aspektów życia codziennego, która może częściowo lub w pełni zastąpić ludzi w przyszłości. To właśnie rozwój androidów jest jednym z najważniejszych rozwiązań dla zwiększenia funkcjonalności działań w środowisku człowieka. Roboty te zazwyczaj przypominają wyglądem ludzi, wyróżniają się konstrukcją ciała o zbliżonym do ludzkiego kształcie, czy również porównywalną zasadą chodzenia. To za sprawą znaczącego elementu robota antropomorficznego, jakim jest miednica z kończynami dolnymi, która formuje dolną część ciała i nadaje te rzeczywiste cechy. W ciągu ostatnich dekad społeczeństwo robotyków zaproponowało kilka udanych modeli dwunożnych, o podobnej strukturze z sześcioma stopniami swobody na nogę tj. 3 będące dla biodra, 1 stopień swobody na kolano oraz 2 dla kostki. Model ten zastosowany jest w znanych androidach o wielkościach zbliżonych prawdziwemu człowiekowi, takich jak Honda ASIMO [1], WABIAN z Uniwersytetu Waseda [2], Boston Dynamics Atlas [3], Android Technics AR-600 [4], itp. Wśród robotów o znacznie mniejszym rozmiarze popularne są ROBOTIS Darwin oraz Poppy Humanoid, który jest analizowany w naszym projekcie. Jego konstrukcja odpowiadająca budowie nóg posiada pięć stopniami swobody.

Ogólnym celem projektu Poppy utworzonego w 2012 r. początkowo było zbadanie właściwości morfologicznych, zwłaszcza skupiając się na poznaniu mechanizmu sensoryczno-motorycznego. Obecnym założeniem jest promowanie kreatywności w społeczeństwie, w której naukowcy, nauczyciele, artyści i entuzjaści robotów mogą dzielić się swoją pracą i pomysłami. Poppy – jako open source'owy dwunożny robot humanoidalny zaprojektowany jest tak, by zapewniać łatwe użycie i łatwą integrację elementów w zestawie, które swobodnie można zmontować i ponownie ustawić.

W tym artykule przedstawione zostały przeprowadzone kroki nad badaniem robota, by móc w przyszłości rozbudować projekt i pracować nad większym przedsięwzięciem, jak zbudowanie własnego humanoida, który posiadać będzie zdolność samodzielnego poruszania się i pokonywania trudnego terenu.

Ogólny projekt robota

Główną cechą robota, który posiada dwie nogi połączone miednicą, jest umiejętność chodzenia, stawiania nóg w sposób ludzki, dlatego też ciało człowieka jest traktowane jako źródło odniesienia dla zaawansowanych mechanizmów w tego typu koncepcjach. Uwzględnione zostało w obliczeniach również nachylenie kości udowej o 6 do 7 stopni w konstrukcji, by jak najbardziej urzeczywistnić wygląd podczas wykonywanego ruchu.

Tabela 1 Zakresy ruchu przegubów dla każdej z części

Przegub	Zakres ruchu (°)
Hip Y	<-25, 45>
Hip Z	<-40, 60>
Hip X	<-60, 100>
Knee X	<-6, 112>
Ankle X	<-35, 30>

Tabela 2 Wartości masy i rozmiar

Część ciała	Masa (kg)	Rozmiar (mm)
Thigh	0,239	181
Shin	0,108	177
Foot	0,047	145 x 50
Całość	1,14	145 x 187 x 495

Dane przedstawione w tabelach 1 oraz 2 zostały pozyskane dzięki programowi Autodesk Inventor poprzez analizę każdego z elementów Poppy. Jak również pomocna w odczycie parametrów okazała się strona projektowa założycieli [5].

Odnosząc się do spisanych wartości zauważyć można zakresy ruchów upodobnione do sprawności nóg człowieka. Problem jaki został napotkany wiąże się z rotacją wokół osi X. Zasięg ten jest znacząco mniejszy, mimo iż wystarczający przy chodzeniu robota, to jednak ograniczający manewrowość innych działań na tych przegubach.

Całe ciało robota Poppy mierzy ok. 80 cm i waży 3,5 kg. Zgodnie z tabelą 2, wymiary samych kończyn dolnych odpowiadają proporcjonalnie do całości konstrukcji (Rysunek 1). Na rysunku 1 można również zauważyć charakterystyczne stopy o prawidłowym rozmiarze, podczas gdy bliźniacze projekty zostawały zazwyczaj obdarzane stopami o znacznie większych wymiarach, dla celów zwiększenia stabilności i wsparcia konstrukcji. Ten osobniczy element zapewnił tym samym dodatkowe trudności w algorytmach i jednocześnie zwiększył ilość wymaganych ruchów podczas kroczenia rozważanego robota.



Rysunek 1 Model nóg zbudowany w programie Autodesk Inventor

Kinematyka prosta

Zadaniem kinematyki prostej jest określenie położenie i orientacje efektora końcowego, poprzez podanie konfiguracji robota. Kinematyka nóg robota Poppy może być zamodelowana jako łańcuch kinematyczny 6 części połączonych 5 przegubami. Bazowy układ współrzędnych znajduje się na środku miednicy. Do rozwiązania problemu kinematyki prostej została zastosowana notacja Denavita-Hartenberga (Tabela 3). Jako, że prawa i lewa noga są identyczne, to obliczenia będą wykonywane tylko dla jednej nogi.

Tabela 3 Parametry kinematyczne jednej nogi robota Poppy

i	Θ_i (rad)	d_i (mm)	a_{i-1} (mm)	α_{i-1} (rad)
1	Θ_1	0	19 (l_0)	$\pi/2$
2	$\Theta_2 + \pi/2$	0	52 (l_1)	$\pi/2$
3	$\Theta_3 + \pi/2$	0	0 (l_2)	$\pi/2$
4	Θ_4	0	181 (l_3)	0
5	Θ_5	0	177 (l_4)	0

Macierz tranzycji A_i dla układu lokalnego wyznacza się poprzez wymnożenie macierzy rotacji oraz translacji [6]. Ta macierz jest reprezentowana przez cztery przekształcenia podstawowe:

$$A_{i-1,i} = Rot_{z,\theta_i} * Trans_{z,d_i} * Trans_{x,a_i} * Rot_{x,\alpha_i}$$

$$A_{i-1,i} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ponieważ, macierz A_i jest funkcją jednej zmiennej, wynika z tego, że trzy z czterech zmiennych są stałe, a czwarty parametr Θ_i , dla przegubu obrotowego lub d_i , dla przegubu przesuwne, jest wielkością zmienną. W przypadku badania nogi humanoida występuje tylko zmienna Θ_i , która odpowiada za ruch obrotowy stawów robota.

Końcowa macierz tranzycji od podstawy do końcowego efektora może być uzyskana poprzez wymnożenie wszystkich tranzycji dla układów lokalnych.

$$T_{05} = A_{01} * A_{12} * A_{23} * A_{34} * A_{45} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & Tx \\ r_{21} & r_{22} & r_{23} & Ty \\ r_{31} & r_{32} & r_{33} & Tz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Jak widać w macierzy T_{05} jest wyróżniona część odpowiadająca za rotację efektora końcowego (miejsca oznaczone literą „r”) oraz położenie w układzie kartezjańskim (miejsca z literą „T”).

Kinematyka odwrotna

Celem kinematyki odwrotnej jest znalezienie zmiennych przegubowych w zależności od pozycji i orientacji końcówki manipulatora. W ogólnym przypadku jest ona trudniejsza od kinematyki prostej, ponieważ często nie istnieje jednoznaczne rozwiązanie. Kinematykę odwrotną można rozwiązać na dwa sposoby: analitycznie i numerycznie. Rozwiązanie analityczne dla manipulatora o 5 stopniach swobody może być trudne do znalezienia, dlatego dla Poppy zostaną wykorzystane dwa sposoby numeryczne:

- Algorytm Cyclic Coordinate Descent (CCD)

Zasada działania powyższego algorytmu opiera się na zmianie wartości przegubów zaczynając od ostatniego [6]. W pierwszej kolejności algorytm oblicza wartość kąta pomiędzy prostą łączącą początek ostatniego rozważanego ramienia oraz końcówkę, a prostą przebiegającą również przez początek ostatniego rozważanego ramienia oraz docelowy punkt, który ramię powinno osiągnąć. Następnie wartość ostatniego przegubu Θ_i jest zmieniana o obliczony kąt. Potem podobne przekształcenie wykonywane jest dla wcześniejszego przegubu, przy czym kąt, o który należy zmienić wartość w tym przegubie obliczany jest pomiędzy prostą przechodzącą przez ten przegub oraz przez punkt końcówki manipulatora, a drugą prostą, która również łączy rozważany przegub oraz punkt docelowy.

- Algorytm jacobianowy

Zasada działania tego algorytmu polega na wielokrotnym wykonywaniu obliczeń według wzoru: $q_{n+1} = q_n - \gamma J^{-1}(P_A - P_D)$, gdzie q jest macierzą o jednej kolumnie zawierającej wszystkie zmienne przegubowe, γ oznacza stałą będącą współczynnikiem odpowiedzialnym za szybkość zmian w równaniu, a macierz jacobianowa J składa się z pochodnych położenia współrzędnych efektora po każdej zmiennej przegubowej, macierze P_A oraz P_D oznaczają odpowiednio macierze aktualnego położenia końcówki manipulatora oraz docelowego punktu, który ma osiągnąć.

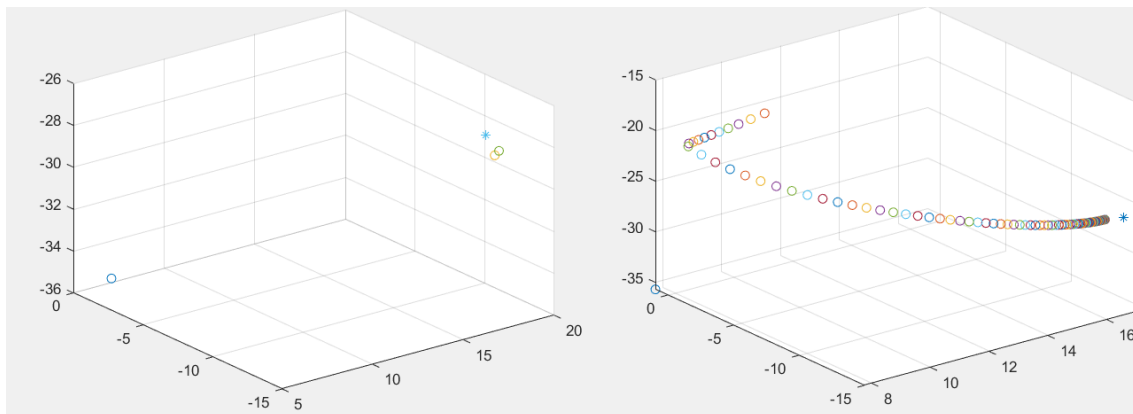
Porównanie algorytmów

Każdy z powyższych algorytmów został napisany w środowisku Matlab. Do kinematyki odwrotnej humanoida została wykorzystana wcześniej wyliczona kinematyka prosta. Dla obydwu algorytmów warunki stopu zostały tak samo określone, czyli 10 tys. iteracji lub odległość końca efektora od zadanego punktu mniejsza niż 1.

Tabela 4 Zestawienie przykładowych wyników wywołania algorytmów

Punkt docelowy	Ilość iteracji	
	CCD	Jakobianowy
(17,-14,-27)	3	98
(13,-11,-25)	Nie znaleziono ustawienia przegubów	569
(22,18,28)	Nie znaleziono ustawienia przegubów	1164

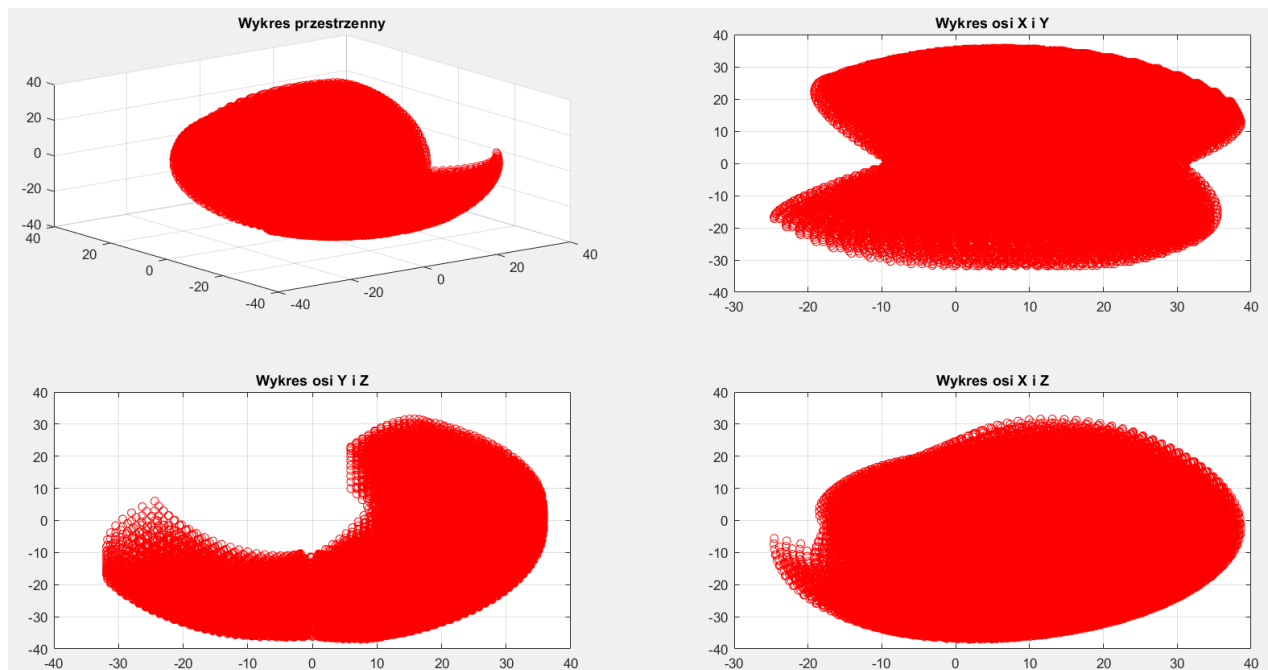
Jak zostało przedstawione w tabeli 4, algorytm CCD w niektórych przypadkach działa o wiele szybciej niż algorytm jakobianowy (Rysunek 2), ale dla trudniejszych przypadków, gdzie jest potrzeba wykonania większej ilości obliczeń, drugi sposób okazuje się skuteczniejszy. Wadą algorytmu Cyclic Coordinate Descent jest to, że czasami może utknąć w lokalnym minimum, przez co algorytm zatrzymuje się w jednym punkcie i uniemożliwia wykonanie dalszych przemieszczeń.



Rysunek 2 Zestawienie wykresów dla punktu (17, -14, -27,) z zaznaczonymi położeniami końcówki manipulatora po wykonaniu każdego kroku algorytmu. Po prawej algorytm CCD, a po lewej algorytm Jakobianowy

Przestrzeń robocza

Obszar roboczy manipulatora zdefiniowany jest jako przestrzeń, do której może dotrzeć efektor. Wyznaczenie przestrzeni roboczej jest kluczowe podczas badania robota, ponieważ pozwala to zobaczyć w jakim zakresie może działać robot. W przypadku badania nóg humanoida jest ona pomocna w późniejszym planowaniu ruchu. Dzięki przestrzeni roboczej, można zaplanować trajektorie ruchu podczas stawiania kroku. Na rysunku 3 można zaobserwować, że zakres ruchu nogi jest duży i manipulator może przybrać wiele pozycji.



Rysunek 3 Wykresy przestrzeni roboczej dla jednej nogi Poppy

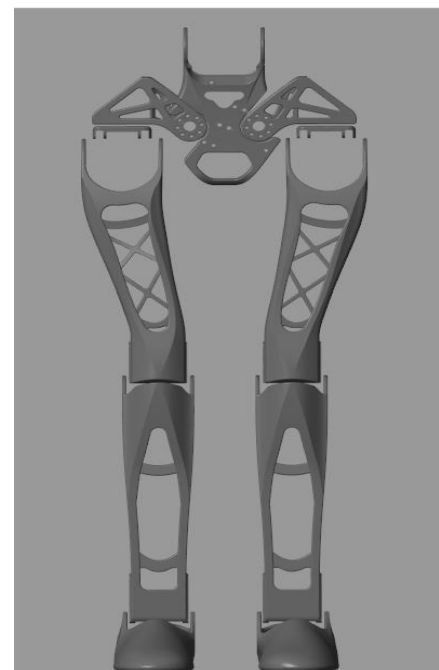
Symulacja

Symulacje są powszechnie wykorzystywane do odtwarzania zjawisk lub zachowań obiektów rzeczywistych. Został utworzony schemat samych nóg Poppy, w programie Multibody (Rysunek 4), który jest dodatkiem do pakietu Simulink Matlab. Jego celem jest sprawdzenie jak zachowuje się dynamika robota i sprawdzenie czy jest możliwe, aby humanoid samoistnie utrzymywał równowagę.

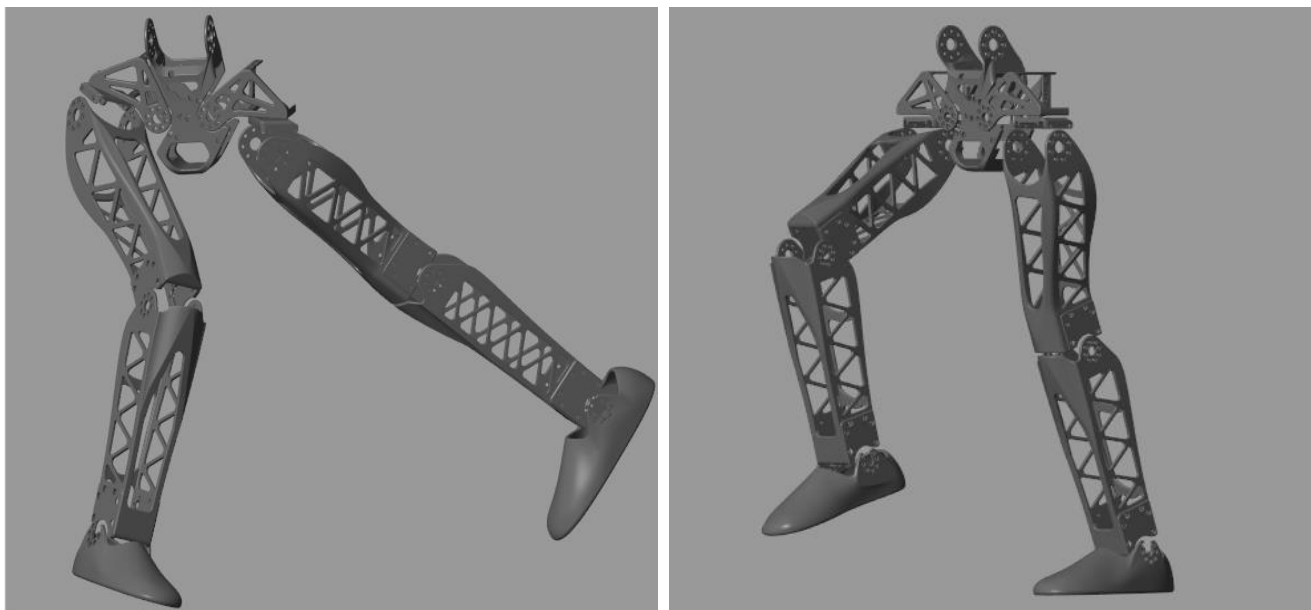
Każda z nóg ma pięć stopni swobody:

- między miednicą, a biodrem
- między biodrem, a podstawką łączącą z nim resztę dołu robota
- trzeci na górze uda
- czwarty w kolanie
- ostatni w kosce stopy

Do wprawiania nóg w ruch zastosowano wcześniej opisaną kinematykę odwrotną w której podawane są współrzędne X, Y i Z. Na ich podstawie odpowiednie funkcje obliczają wartości przegubów, które są przekazywane dalej do robota. Niestety zastosowany sposób do poruszania nogami nie zawsze działa. Czasem może zdarzyć się sytuacja w której noga znajdzie się w nietypowym ułożeniu.



Rysunek 4 Model nóg zbudowany w pakiecie Matlab Multibody



Rysunek 5 Efekt symulacji wykorzystującej kinematykę odwrotną. Po lewej przykład złego ustawienia nogi, po prawej przykład poprawnie ustawionych nóg.

Na początku niemożliwe było zasymulowanie ruchu w programie Matlab, ponieważ po stworzeniu modelu program ten nie potrafił obliczyć masy, momentów bezwładności oraz współrzędnych środka masy dla żadnego elementu. Każda część po kolei została zaimplementowana do programu Autodesk Inventor. Dzięki temu można było odczytać niezbędne dane i przypisać je do odpowiednich elementów w schemacie stworzonym w Simulinku, aby móc kontynuować dalsze prace.

Literatura:

1. F. Sun, H. H. Ju, P. Y. Cui, *A new 12 DOF biped robot's mechanical design and kinematic analysis*, Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology, 2011, Harbin, China
2. Y. Ogura, H. Aikawa, K. Shimomura, H. Kondo, A. Morishima, H. Lim, A. Takanishi, *Development of a new humanoid robot WABIAN-2*, Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006, Orlando, FL, USA
3. S. Faraji, S. Pouya, C. G. Atkeson, A. J. Ijspeert, *Versatile and robust 3D walking with a simulated humanoid robot (Atlas): A model predictive control approach*, 2014 IEEE International Conference on Robotics and Automation, 2014, Hong Kong, China
4. R. Khusainov, I. Shimchik, I. Afanasyev, E. Magid, *Toward a human-like locomotion: Modelling dynamically stable locomotion of an anthropomorphic robot in simulink environment*, 2015 12th International Conference on Informatics in Control, Automation and Robotics, 2015, Colmar, France
5. <https://www.poppy-project.org>, data dostępu: 15.04.2019
6. R. Leniowski, *Podstawy Robotyki*, Uniwersytet Rzeszowski, 2005, Rzeszów