



WYDZIAŁ ELEKTRYCZNY

Katedra Automatyki i Robotyki

Raport z realizacji pracy przejściowej

Marcin Przekop

Projekt manipulatora o strukturze OOP

Prowadzący

Dr hab. Inż. Zbigniew Kulesza

Białystok 2023

Streszczenie

W niniejszej pracy wykonany zostanie projekt robota typu OOP. Przedstawiona zostanie metodyka pracy, opis robota i kinematyki z nim związanej, wykonane zostaną zadania kinematyki prostej oraz odwrotnej a wyniki zweryfikowane będą przy użyciu symulacji. Odpowiednie założenia konstrukcyjne robota uwzględnione zostaną w jego projektowaniu i dobrane zostaną odpowiednie napędy pneumatyczne oraz elektryczne.

Spis treści

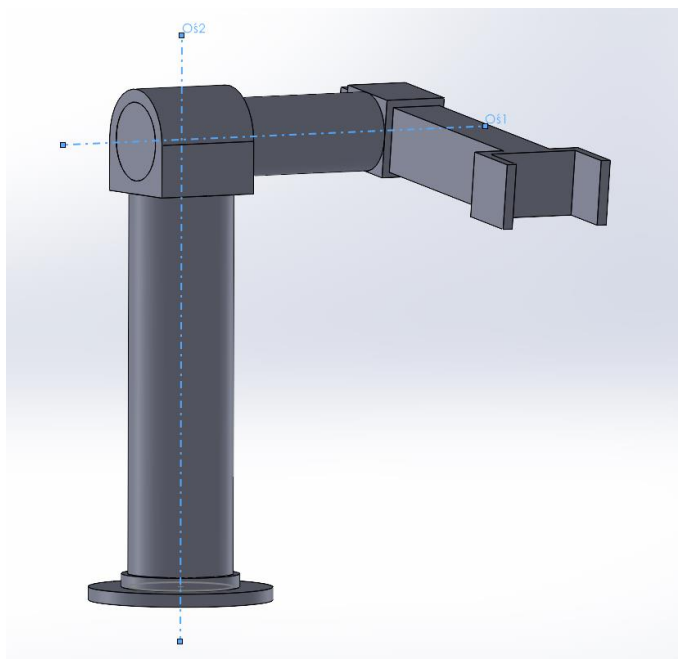
Spis ważniejszych oznaczeń	4
1. Wstęp	5
1.1. Cel pracy i zakres pracy	6
1.2. Metodyka realizacji	6
1.3. Charakterystyka pracy	7
1.4. Podsumowanie	7
2. Opis szczegółowy robota	8
3. Opis kinematyki	10
3.1. Zadanie kinematyki prostej	11
3.2. Zadanie kinematyki odwrotnej	14
3.3. Opis matematyczny toru ruchu robota	15
4. Badania symulacyjne	16
5. Założenia projektowe do realizacji prototypu robota	20
6. Trójwymiarowy model geometryczny robota wykonany w programie SolidWorks	21
7. Dobór napędów	25
7.1. Wymagania dotyczące momentów obrotowych	25
7.2. Dobór napędu pneumatycznego	26
7.3. Dobór napędów elektrycznych	27
8. Schemat połączeń	29
9. Szacunkowe zestawienie kosztów	30
10. Wnioski końcowe	31
Bibliografia	32

Spis ważniejszych oznaczeń

a	– długość wspólnej normalnej
α	– obrót wokół wspólnej normalnej, od starej osi z do nowej osi z
d	– przesunięcie wzdłuż poprzedniego z do wspólnej normalnej
θ	– obrót względem poprzedniego z , od starej osi x do nowej osi x
l_x	– stała wymiarowa robota
q_x	– zmienna konfiguracyjna robota
x_yT	– macierz przejścia

1. Wstęp

Postęp robotyzacji spowodował, że dotychczas wykonywane prace, za które odpowiadał człowiek realizowane są za pomocą różnych manipulatorów a najpowszechniejsze stały się roboty typu SCARA ze względu na ich dużą precyzję. Istotne zatem jest zrozumienie w jaki sposób one funkcjonują i jak można je sprawnie wykorzystać. Celem tej pracy przejściowej jest zaprojektowanie manipulatora o strukturze OOP. Rozpatrzone zostaną wszelkie zagadnienia dotyczące zarówno kinematyki jak i dynamiki tego typu robota w oparciu o literaturę. Dodatkowo pod uwagę zostanie wzięty dobór elementów robota. Przykładowy model robota o strukturze OOP (na podstawie [2]) ukazano na rysunku 1.1.



Rys.1.1. Przykładowy manipulator o strukturze OOP

1.1. Cel pracy i zakres pracy

Celem niniejszej pracy jest wykonanie projektu robota o strukturze OOP mającego za zadanie wykonywać zadanie typu “pick and place”. Zostaną wykonane niezbędne do realizacji zadania obliczenia oraz dobrane zostaną odpowiednie napędy.

Zakres pracy obejmuje wykonanie następujących prac:

- sporządzenie szczegółowego opisu robota,
- wykonanie opisu kinematyki prostej i odwrotnej robota,
- sporządzenie matematycznego opisu toru ruchu robota,
- przeprowadzenie symulacji mających na celu weryfikację poprawności działań,
- sporządzenie założeń projektowych do realizacji robota,
- zaprojektowanie modelu trójwymiarowego,
- dobór odpowiednich podzespołów do robota,
- przygotowanie schematu połączeń elektrycznych
- sporządzenie szacunkowego zestawienia kosztów

1.2. Metodyka realizacji

W niniejszej pracy badania symulacyjne zostaną wykonane w oprogramowaniu MATLAB a ich wizualizacja zrealizowana będzie przy pomocy dodatku Robotics Toolbox. Do projektowania modelu trójwymiarowego robota wykorzystany zostanie program SolidWorks a dodatek Solidworks Motion użyty zostanie do określenia wymaganych momentów obrotowych.

1.3. Charakterystyka pracy

W drugim rozdziale przedstawiony zostanie opis szczegółowy robota, pokazany zostanie jego model koncepcyjny oraz przedstawione zostaną oznaczenia wykorzystywane w dalszej części pracy.

Rozdział trzeci zawiera opis kinematyczny robota oraz obejmuje wykonanie zadania kinematyki prostej i odwrotnej wraz z przedstawieniem kodu do programu MATLAB.

W rozdziale czwartym wykonane zostały badania symulacyjne przeprowadzone w celu weryfikacji poprawności pracy wykonanej w poprzednim rozdziale.

Rozdział piąty dotyczy założeń projektowych dotyczących realizacji prototypu robota.

W rozdziale szóstym przedstawiono zaprojektowany model trójwymiarowy wykonany w programie SolidWorks.

Rozdział siódmy zawiera zestawienia i porównania napędów wykorzystanych w realizacji pracy.

Ósmy rozdział dotyczy schematu połączeń elementów robota.

Rozdział dziewiąty zawiera zestawienie szacunkowych kosztów robota.

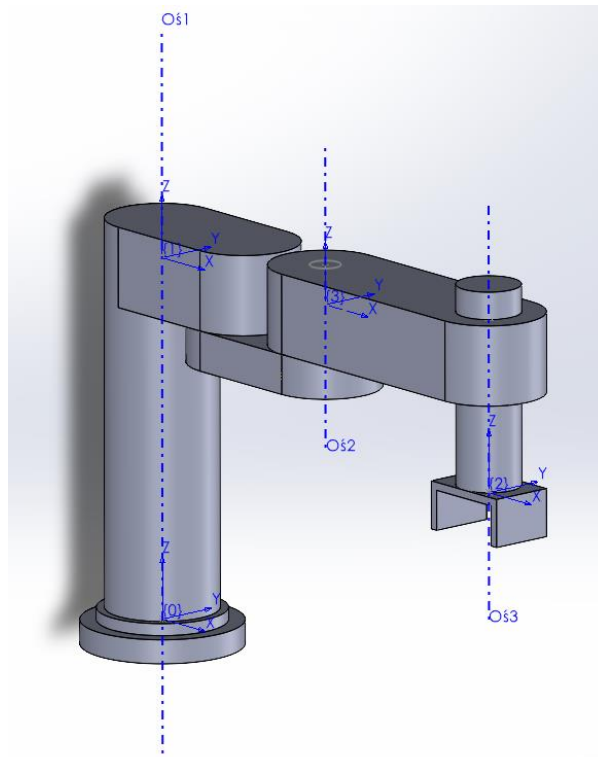
W dziesiątym rozdziale przedstawione zostaną wnioski dotyczące realizacji całego projektu.

1.4.Podsumowanie

Niniejsza praca polegać będzie na zaprojektowaniu robota typu SCARA począwszy od sporządzenia odpowiednich opisów matematycznych po dobór elementów i zaprojektowanie modelu trójwymiarowego robota.

2. Opis szczegółowy robota

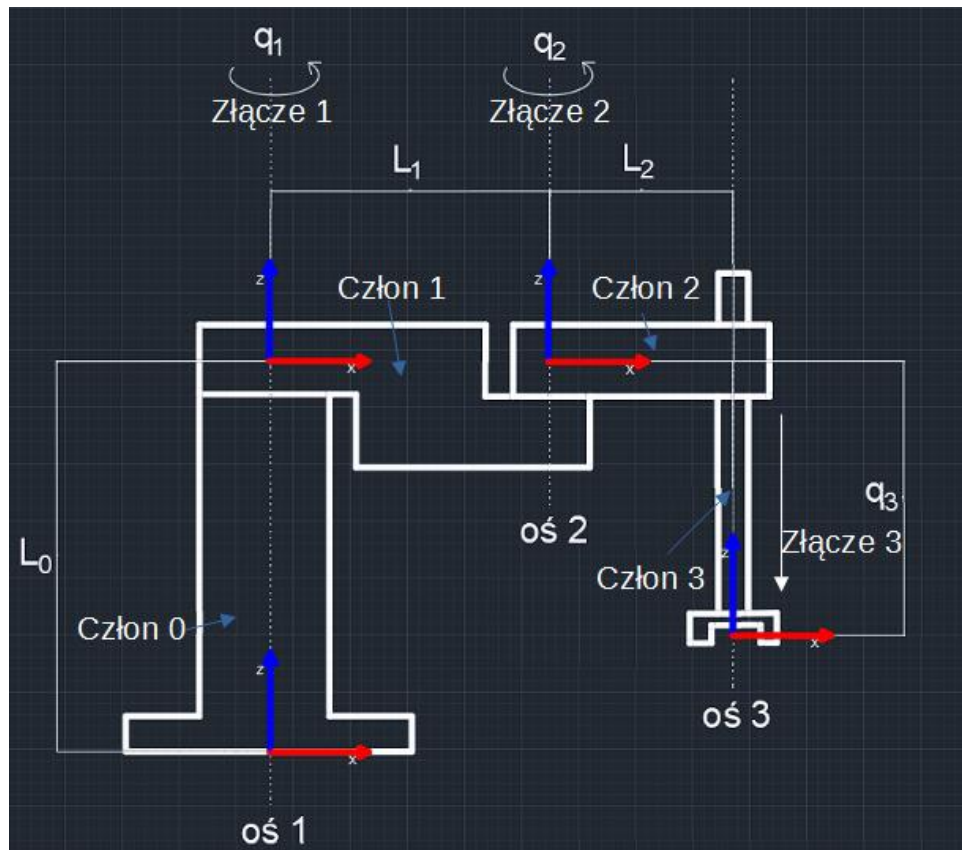
Projektowany robot będzie składać się z dwóch równoległych osi obrotowych i jednej osi przesuwnej (także równoległej do osi obrotowych). Model koncepcyjny robota ukazano na rysunku 2.1.



Rys.2.1. Model realizowanego robota (na podstawie [6])

Zadanie robota polegać będzie na układaniu szklanych butelek (każda o masie około 320 g) pojemnika na taśmę produkcyjną. Z technicznego punktu widzenia istotne więc będzie jak powtarzalność robota i jego precyzja. Do jego konstrukcji wykorzystany zostanie jeden obrotowy napęd pneumatyczny (obróć wokół osi 1), jeden serwomotor (odpowiadający za obrót wokół osi 2), jeden siłownik liniowy z enkoderem (odpowiedzialny za przesuw wzdłuż osi 3) oraz dodatkowy serwomotor wykonujący operację chwytania.

Układy współrzędnych oznaczenia zmiennych oraz numerację członów i przegubów przyjęto tak jak na rysunku 2.2.



Rys.2.2. Schemat oznaczeń robota

3. Opis kinematyki

Kinematyka odgrywa istotną rolę w projektowaniu i analizie ruchu robota, ponieważ zajmuje się opisem jego geometrii, położenia i orientacji w przestrzeni oraz relacji między różnymi elementami. Poniżej opisane zostaną sposoby realizacji zarówno kinematyki prostej jak i odwrotnej dla robota o strukturze OOP. Kluczowe do realizacji zadań kinematycznych jest odpowiednie odczytanie parametrów D-H, które zapisano w tabeli 3.1 [4].

Tab. 3.1. Tabela zawierająca parametry D-H

i	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	l_0	$q1$
2	l_1	0	0	$q2$
3	l_2	0	$-q3$	0

3.1. Zadanie kinematyki prostej

Zadanie kinematyki prostej polega na określeniu położenia efektora końcowego na podstawie zadanej konfiguracji. Do rozwiązania tego zagadnienia wykorzystane zostanie przekształcenie D-H. Macierz przekształcenia pomiędzy układami a można wyznaczyć jako szereg przekształceń [3]:

$${}^0_i T = {}^0_1 T \cdot {}^1_2 T \cdot \dots \cdot {}^{i-1}_i T \quad (3.1)$$

Macierze przekształceń wyznaczono poniżej.

Macierz przejścia z układu współrzędnych {0} do układu współrzędnych {1} [3]:

$${}^0_1 T = \begin{bmatrix} \cos(q_1) & -\sin(q_1) & 0 & 0 \\ \sin(q_1) & \cos(q_1) & 0 & 0 \\ 0 & 0 & 1 & l_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.2)$$

Macierz przejścia z układu współrzędnych {1} do układu współrzędnych {2} [3]:

$${}^1_2 T = \begin{bmatrix} \cos(q_2) & -\sin(q_2) & 0 & l_1 \\ \sin(q_2) & \cos(q_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.3)$$

Macierz przejścia z układu współrzędnych {2} do układu współrzędnych {3} [3]:

$${}^2_3 T = \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.4)$$

Wzór na macierz przekształceń z globalnego układu współrzędnych do układu współrzędnych efektora końcowego [3]:

$${}^0_3 T = {}^0_1 T \cdot {}^1_2 T \cdot {}^2_3 T \quad (3.5)$$

Po prawidłowym wymnożeniu wszystkich macierzy pośrednich przekształceń otrzymujemy macierz 3.6 [3]:

$${}^0_3T = \begin{bmatrix} t_1 & t_2 & 0 & t_3 \\ t_4 & t_5 & 0 & t_6 \\ 0 & 0 & 1 & l_0 - q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

gdzie:

$$t_1 = [\cos(q_1) \cdot \cos(q_2) - \sin(q_1) \cdot \sin(q_2)], \quad (3.7)$$

$$t_2 = [-\cos(q_1) \cdot \sin(q_2) - \cos(q_2) \cdot \sin(q_1)], \quad (3.8)$$

$$t_3 = [l_2 \cdot \cos(q_1) \cdot \cos(q_2) - \sin(q_1) \cdot \sin(q_2) + l_1 \cos(q_1)], \quad (3.9)$$

$$t_4 = [\cos(q_1) \cdot \sin(q_2) + \cos(q_2) \cdot \sin(q_1)], \quad (3.10)$$

$$t_5 = [\cos(q_1) \cdot \cos(q_2) - \sin(q_1) \cdot \sin(q_2)], \quad (3.11)$$

$$t_6 = [l_2 \cdot \cos(q_1) \cdot \sin(q_2) + \cos(q_2) \cdot \sin(q_1) + l_1 \sin(q_1)]. \quad (3.12)$$

Na wydruku 3.1 (opracowanym na podstawie [1]) zamieszczono kod programu realizowany w oprogramowaniu MATLAB odpowiedzialny za wykonywanie odpowiednich obliczeń i wyświetlenie wyników. Po podaniu parametrów z tabeli 3.1 oraz stałych program oblicza odpowiednie przekształcenia zgodnie ze wzorem 3.5.

Wydruk 3.1. Kod odpowiedzialny za wykonywanie zadania kinematyki prostej

```

clc; clear all; close all;
syms l0 l1 l2 q1 q2 q3;
%Wprowadzanie stałych:
L0=70; l1=100; l2=80;
%Wprowadzanie danych konfiguracyjnych:
q1=deg2rad(45); q2=deg2rad(-90); q3=20;
%Wprowadzenie danych %z tabeli (3.1):
a0 = 0; alpha0 = 0; d1 = l0; theta1 = q1;
a1 = l1; alpha1 = 0; d2 = 0; theta2 = q2;
a2 = l2; alpha2 = 0; d3 = q3; theta3 = 0;
T_0_1 = DH(a0, alpha0, d1, theta1) %przekształcenie (3.2)
T_1_2 = DH(a1, alpha1, d2, theta2) %przekształcenie (3.3)
T_2_3 = DH(a2, alpha2, d3, theta3) %przekształcenie (3.4)
T_0_3 = T_0_1 * T_1_2 * T_2_3 %przekształcenie (3.6)
function T = TwistX(a, alpha)
    ca = cos(alpha);
    sa = sin(alpha);
    %macierz translacji wokół 'a' z DH (na podstawie [5])
    T = [
        1 0 0 a
        0 ca -sa 0
        0 sa ca 0
        0 0 0 1];
end
function T = TwistZ(d, theta)
    ct = cos(theta);
    st = sin(theta);
    %macierz translacji wokół 'd' z DH (na podstawie [5])
    T = [
        ct -st 0 0
        st ct 0 0
        0 0 1 d
        0 0 0 1];
end
function T = DH(a, alpha, d, theta)
    T = TwistX(a, alpha) * TwistZ(d, theta);
End

```

3.2. Zadanie kinematyki odwrotnej

By wykonać zadanie kinematyki odwrotnej musimy przyjąć q_1 , q_2 oraz q_3 jako szukane, natomiast wartości x , y i z są podane i znane. Mamy trzy niewiadome więc potrzebne są trzy równania, aby móc je wyznaczyć. Otrzymujemy je z wykonanej wcześniej macierzy przekształceń 0_3T (3.1.) i otrzymujemy poniższe równania:

$$x = l_2 \cdot (\cos(q_1) \cdot \cos(q_2) - \sin(q_1) \cdot \sin(q_2)) + l_1 \cdot \cos(q_1), \quad (3.14)$$

$$y = l_2 \cdot (\cos(q_1) \cdot \sin(q_2) + \sin(q_1) \cdot \cos(q_2)) + l_1 \cdot \sin(q_1), \quad (3.15)$$

$$z = l_0 - q_3. \quad (3.16)$$

Po rozwiązaniu układu równań otrzymujemy funkcję do wyznaczenia poszczególnych współrzędnych. Na wydruku 3.2 przedstawiono kod programu MATLAB (opracowany na podstawie [8]). wykonujący odpowiednie obliczenia do realizacji zadania kinematyki odwrotnej. Po wprowadzeniu stałych oraz żądanej pozycji końcowej program oblicza parametry konfiguracyjne zgodnie ze wzorami 3.6; 3.7; oraz 3.8 umożliwiające jej osiągnięcie. Dodatkowo kod posiada funkcję symulującą pozycję robota o zadanych stałych i parametrach konfiguracyjnych.

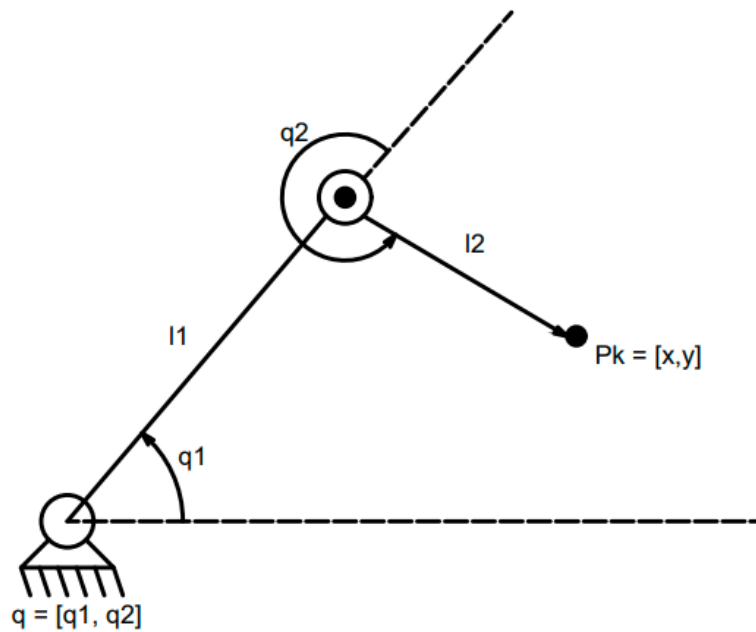
Wydruk 3.2. Kod odpowiedzialny za realizację zadania kinematyki odwrotnej

```
clc; clear all; close all;
syms l0 l1 l2 q1 q2 q3 ik_x ik_y ik_z ikq1 ikq2 ikq3;
%Współrzędne docelowe:
ik_x=60;
ik_y=90;
ik_z=80;
%Stałe:
l0=70; l1=100; l2=80;
%Układ równań:
ik=[ %wzór 3.14
    ik_x==l2*(cos(q1)*cos(q2) - sin(q1)*sin(q2)) + l1*cos(q1);
    %wzór 3.15
    ik_y==l2*(cos(q1)*sin(q2) + cos(q2)*sin(q1)) + l1*sin(q1);
    %wzór 3.16
    ik_z==l0 - q3;
];
%Funkcja Matlab'a rozwiązująca układy równań:
S=solve(ik, [q1 q2 q3]);
%przypisanie wartości poszczególnych zmiennych konfiguracyjnych
do zmiennej oraz zamiana radianów na stopnie
ikq1=eval(subs(S.q1))*180/pi;
ikq2=eval(subs(S.q2))*180/pi;
ikq3=eval(subs(S.q3));
%Wyświetlenie rozwiązań q1
disp('q1:');
disp(ikq1);
%Wyświetlenie rozwiązań q2
disp('q2:');
disp(ikq2);
%Wyświetlenie rozwiązań q3
```

```
disp('q3:');  
disp(ikq3(1,1));
```

3.3. Opis matematyczny toru ruchu robota

Tor ruchu robota można wyznaczyć uzależniając wartości konfiguracyjne poszczególnych członów od czasu. Położenie (x,y) można wyznaczyć algebraicznie patrząc na robota z góry, wtedy oznaczenia wyglądają jak na rysunku 3.1.



Rys.3.1. Widok robota z góry

Zatem:

$$x(t) = l_1 \cos q_1 + l_2 \cos(q_1 + q_2), \quad (3.17)$$

$$y(t) = l_1 \sin q_1 + l_2 \sin(q_1 + q_2). \quad (3.18)$$

Przesunięcie wzdłuż osi Z jest zależne jedynie od zmiennej konfiguracyjnej q_3 , czyli zgodnie z rysunkiem 2.2 otrzymano:

$$z(t) = l_0 - q_3. \quad (3.19)$$

4. Badania symulacyjne

W celu sprawdzenia poprawności wykonanych do tej pory działań wykonano po trzy symulacje dla zadań kinematyki prostej i odwrotnej. Dane konfiguracyjne, wyniki oraz symulacje zestawiono w odpowiednich tabelach.

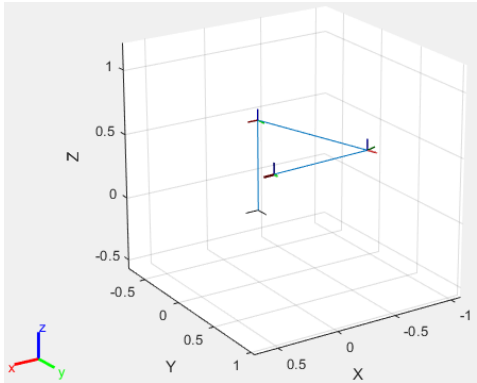
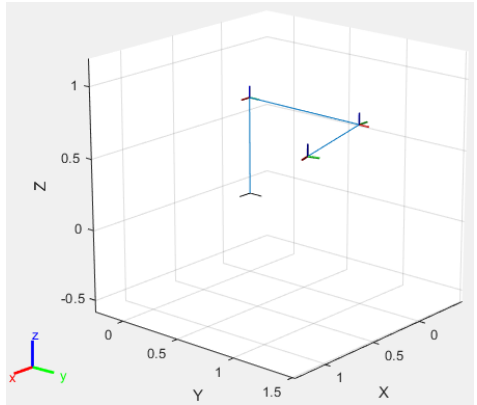
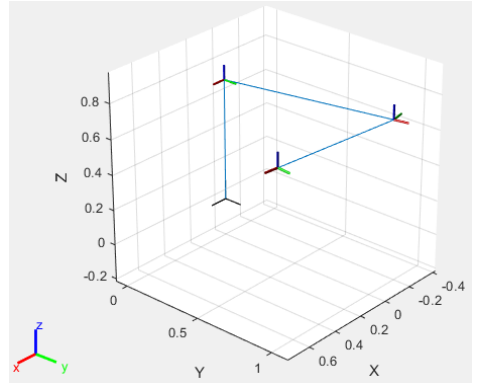
W pierwszej kolejności przeprowadzono symulacje dla zadania kinematyki prostej, do których wykorzystano kod programu z wydruku (3.1) a wyniki przedstawiono w tabeli 4.1.

Tab. 4.1. Trzy przykładowe realizacje zadania kinematyki prostej

Stałe	Zmienne	Wynik	Wizualizacja
$l_0 = 70,$ $l_1 = 100,$ $l_2 = 80$	$q_1 = 30,$ $q_2 = 30,$ $q_3 = 10$	$T_{0_3} =$ <pre> 0.5000 -0.8660 0 126.6025 0.8660 0.5000 0 119.2820 0 0 1.0000 60.0000 0 0 0 1.0000 </pre> $x=126.602540$ $y=119.282032$ $z=60.000000$ Obrót wokół osi Z = 30.000000 stopni Rys.4.1. Wynik pierwszej symulacji	 Rys.4.2. Wizualizacja pierwszej symulacji
	$q_1 = 40,$ $q_2 = 60,$ $q_3 = 25$	$T_{0_3} =$ <pre> -0.1736 -0.9848 0 62.7126 0.9848 -0.1736 0 143.0634 0 0 1.0000 50.0000 0 0 0 1.0000 </pre> $x=62.712590$ $y=143.063381$ $z=50.000000$ Obrót wokół osi Z = -10.000000 stopni Rys.4.3. Wynik drugiej symulacji	 Rys.4.4. Wizualizacja pierwszej symulacji
	$q_1 = 80,$ $q_2 = 10,$ $q_3 = 15$	$T_{0_3} =$ <pre> 0 -1.0000 0 17.3648 1.0000 0 0 178.4808 0 0 1.0000 55.0000 0 0 0 1.0000 </pre> $x=17.364818$ $y=178.480775$ $z=55.000000$ Obrót wokół osi Z = 0.000000 stopni Rys.4.5. Wynik trzeciej symulacji	 Rys.4.6. Wizualizacja pierwszej symulacji

Następnie w tabeli 4.2 umieszczono wyniki symulacji dla zadań kinematyki odwrotnej wykonanych na podstawie kodu programu z wydruku 3.2.

Tab. 4.2. Trzy przykładowe realizacje zadania kinematyki odwrotnej

Stałe	Zadane współrzędne	Wynik	Wizualizacja
$l_0 = 70,$ $l_1 = 100,$ $l_2 = 80$	$x = 50,$ $y = 100,$ $z = 40$	<pre> q1: 107.3793 19.4906 q2: -104.1080 104.1080 q3: 30 </pre> <p>Rys.4.7. Wynik pierwszej symulacji</p>	 <p>Rys.4.8. Wizualizacja pierwszej symulacji</p>
	$x = 80,$ $y = 120,$ $z = 60$	<pre> q1: 88.5395 24.0803 q2: -74.0380 74.0380 q3: 10 </pre> <p>Rys.4.9. Wynik drugiej symulacji</p>	 <p>Rys.4.10 Wizualizacja pierwszej symulacji</p>
	$x = 60,$ $y = 90,$ $z = 80$	<pre> q1: 101.2989 11.3210 q2: -107.0826 107.0826 q3: -10 </pre> <p>Rys.4.12 Wynik trzeciej symulacji</p>	 <p>Rys.4.13 Wizualizacja pierwszej symulacji</p>

W wydruku 4.1 zamieszczono kod programu MATLAB wykonujący symulację pozycji poszczególnych członów robota [7]. Kod pobiera z tabeli 3.1 dane dotyczące ilości członów oraz ich parametrów konfiguracyjnych i tworzy wizualizację pozycji robota.

Wydruk 4.1. Fragment kodu odpowiedzialny za wizualizację robota

```
%Wizualizacja
dhparams = [0          0          10          q1;
            11         0          0          q2;
            12         0          0          0;
            0          0         -q3          0];

robot = rigidBodyTree;

bodies = cell(4,1);
joints = cell(4,1);
for i = 1:4
    bodies{i} = rigidBody(['body' num2str(i)]);
    if i== 4
        joints{i}=rigidBodyJoint(['jnt'num2str(i)],"prismatic");
    else
        joints{i}=rigidBodyJoint(['jnt'num2str(i)],"revolute");
    end
    setFixedTransform(joints{i},dhparams(i,:), "dh");
    bodies{i}.Joint = joints{i};
    if i == 1 % Pierwszy element jako baza
        addBody(robot,bodies{i},"base")
    else
        addBody(robot,bodies{i},bodies{i-1}.Name)
    end
end
showdetails(robot)
config = homeConfiguration(robot);
show(robot);
%Wprowadzenie danych konfiguracyjnych do symulacji
config(2).JointPosition =q1;
config(3).JointPosition =q2;
config(4).JointPosition =q3;
show(robot,config);
```

W celu dodatkowej weryfikacji poprawności wykonania zadań kinematyki prostej i odwrotnej należy podać dane konfiguracyjne do danego punktu dla zadania kinematyki odwrotnej a następnie wynik q_1, q_2, q_3 podać jako dane dla zadania kinematyki prostej. Jeżeli pozycja końcówki robota będzie zgodna z tą do obliczania kinematyki odwrotnej oznaczać to będzie, że wszystkie operacje wykonane zostały prawidłowo.

Jako pozycję końcową podano:

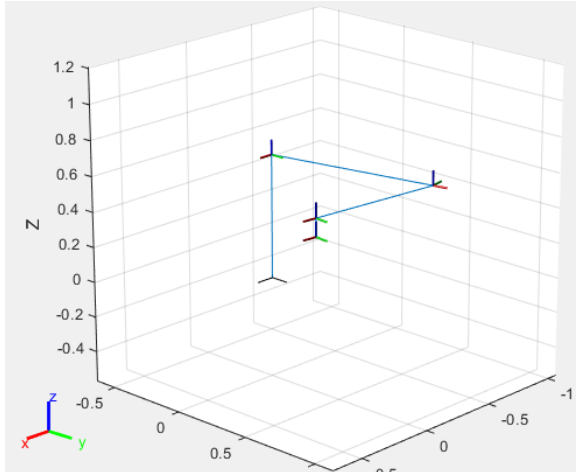
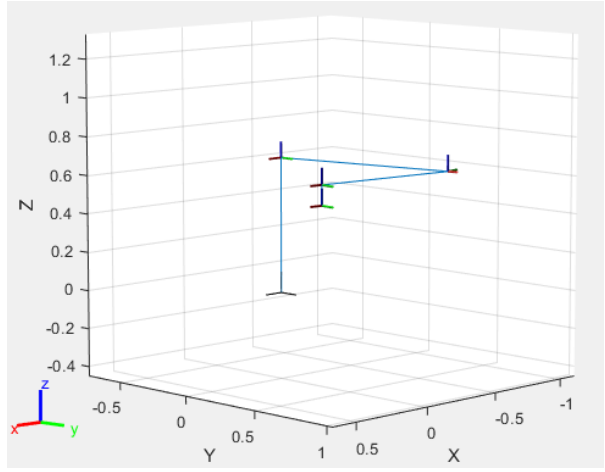
$$x = 60, y = 90, z = 80 \quad (4.1)$$

Otrzymano poniższe wartości:

$$q_1 = 101.2989, q_2 = -107.0826, q_3 = -10 \quad (4.2)$$

Po podaniu powyższych parametrów jako dane do zadania kinematyki prostej otrzymano tą samą pozycję końcówki jak na początku co ukazano w tabeli 4.3 na rysunku 4.14 oraz na rysunku 4.15.

Tab. 4.3. Weryfikacja działania obu zadań kinematyki

Pozycja robota dla zadania kinematyki prostej	Pozycja robota dla zadania kinematyki odwrotnej
 <p>Rys.4.14. Wizualizacja dla zadania kinematyki prostej</p>	 <p>Rys.4.15. Wizualizacja dla zadania kinematyki odwrotnej</p>

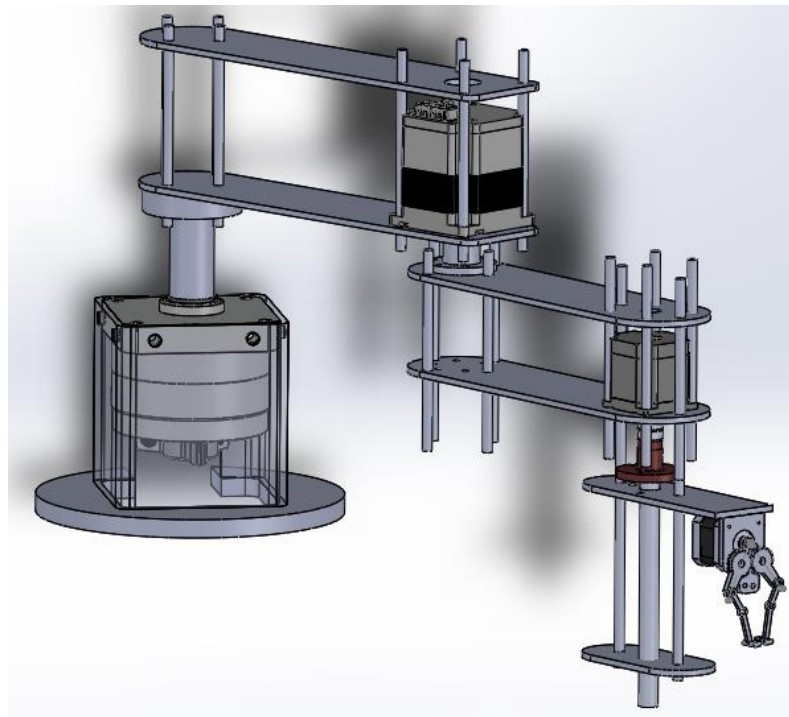
5. Założenia projektowe do realizacji prototypu robota

Do realizacji prototypu robota przyjęto następujące założenia projektowe:

- zostanie wykorzystany jeden obrotowy napęd pneumatyczny znajdujący się w podstawie manipulatora (obrót wokół osi 1),
- zostanie wykorzystany jeden wybrany silnik obrotowy (obrót wokół osi 2),
- zostanie wykorzystany jeden napęd liniowy (obrót wzdłuż osi 3),
- zostanie wykorzystany jeden napęd odpowiedzialny za realizację funkcji chwytania,
- zasięg maksymalny robota wynosić będzie 0,9 metra,
- masa pojedynczej szklanej butelki wynosi 320 g,
- butelki odbierane będą z pojemnika i odkładane na taśmę produkcyjną w celu ich napełnienia.

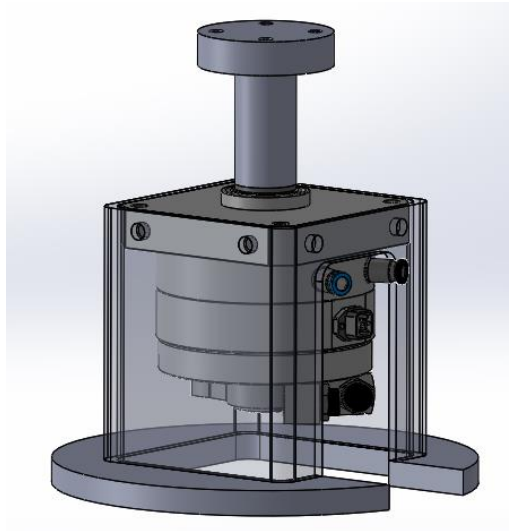
6. Trójwymiarowy model geometryczny robota wykonany w programie SolidWorks

Cały model robota został wykonany w programie SolidWorks, poszczególne elementy takie jak silniki pobrane zostały bezpośrednio z witryn internetowych producentów. Przy projektowaniu uwzględniono odpowiednią technologię wykonania każdego elementu. Całkowite złożenie przedstawiono na rysunku 6.1.



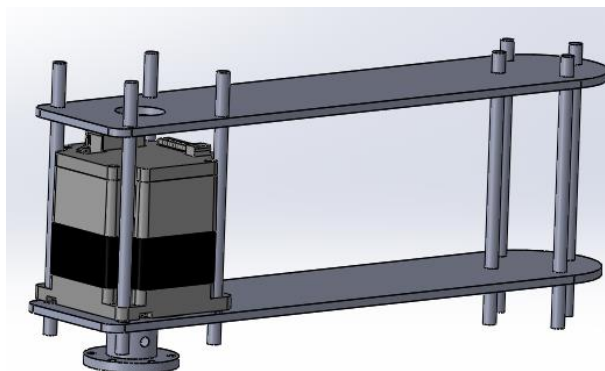
Rys.6.1. Całkowity model robota

Podstawa i obudowa wykonane są z blachy aluminiowej o grubości 3 mm, silnik pneumatyczny pochodzi od producenta FESTO, wał wykonany został z wałka stalowego o średnicy 40 mm a mocowanie dla ramienia wykonano z wałka o średnicy 80mm. Model pokazano na rysunku 6.2.



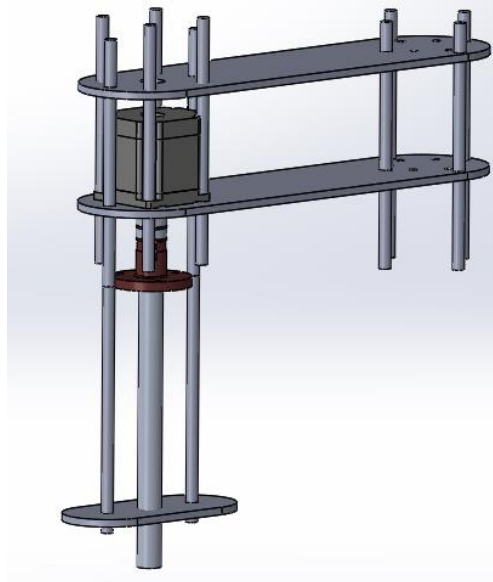
Rys.6.2. Podstawa robota

Człon pierwszy robota widnieje na rysunku 6.3 i składa się z dwóch elementów wyciętych z blachy aluminiowej o grubości 5 mm, silnika umożliwiającego obrót wokół osi {2}, wału montażowego o średnicy 14 mm oraz śrub trapezowych o średnicy 8 mm wraz z odpowiednimi nakrętkami i podkładkami.



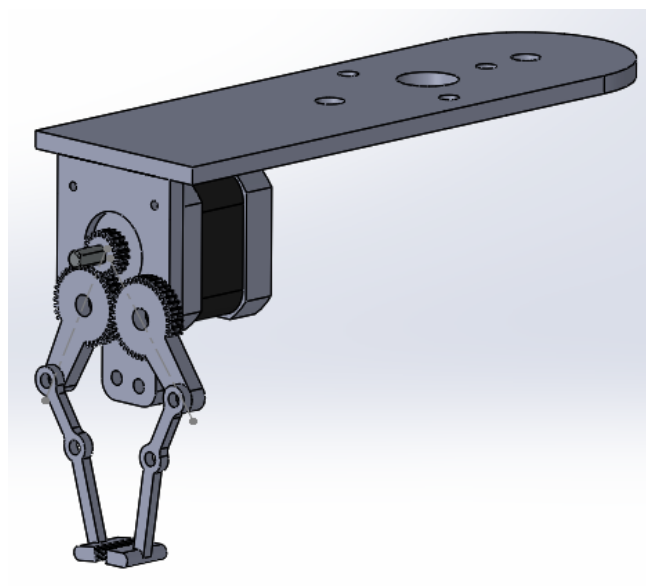
Rys.6.3. Człon pierwszy robota

Na rysunku 6.4. przedstawiono drugi człon robota, który zawiera trzy elementy wykonane z blachy aluminiowej o grubości 5 mm, napęd liniowego oraz śruby trapezowe z odpowiednimi podkładkami i nakrętkami.



Rys.6.4. Człon drugi robota

Elementy chwytaka zostały wykonane z blach aluminiowych o grubości 5mm, użyty silnik jest w standardzie NEMA 17. Na rysunku 6.5 pokazano model złożeniowy chwytaka.



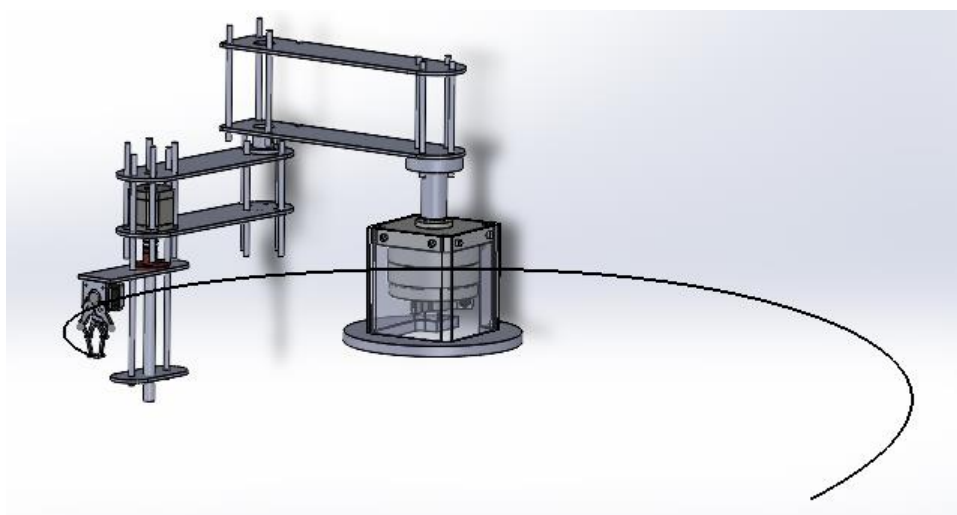
Rys.6.5. Chwytnik

7. Dobór napędów

Istotnym elementem manipulatora są jego napędy odpowiedzialne za ruch. Przy ich doborze należy uwzględnić samą masę robota jak i obiektu, który będzie on przenosił.

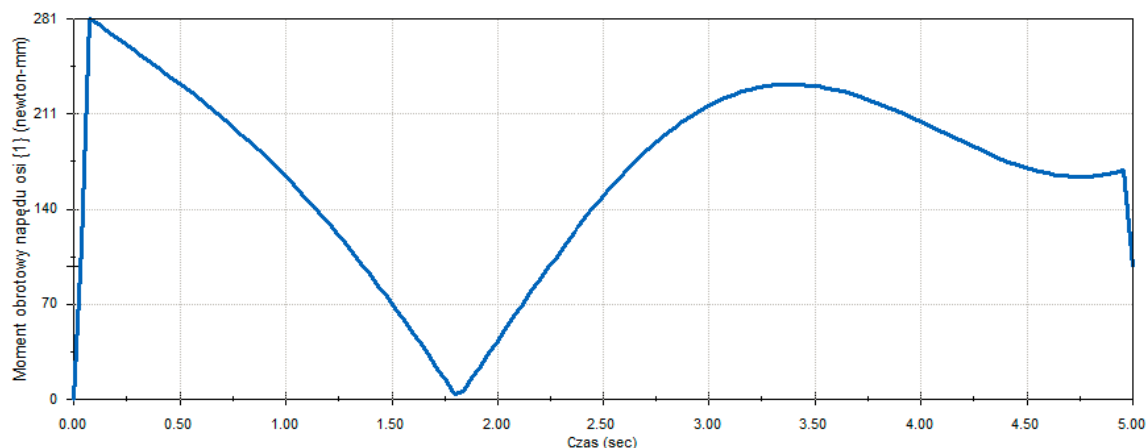
7.1. Wymagania dotyczące momentów obrotowych

Przed doбором napędów wykonano analizę ruchu modelu w programie SolidWorks oraz dodano odpowiednie materiały konstrukcyjne. Tor ruchu robota wykorzystany do badań przedstawiono na rysunku 7.1.



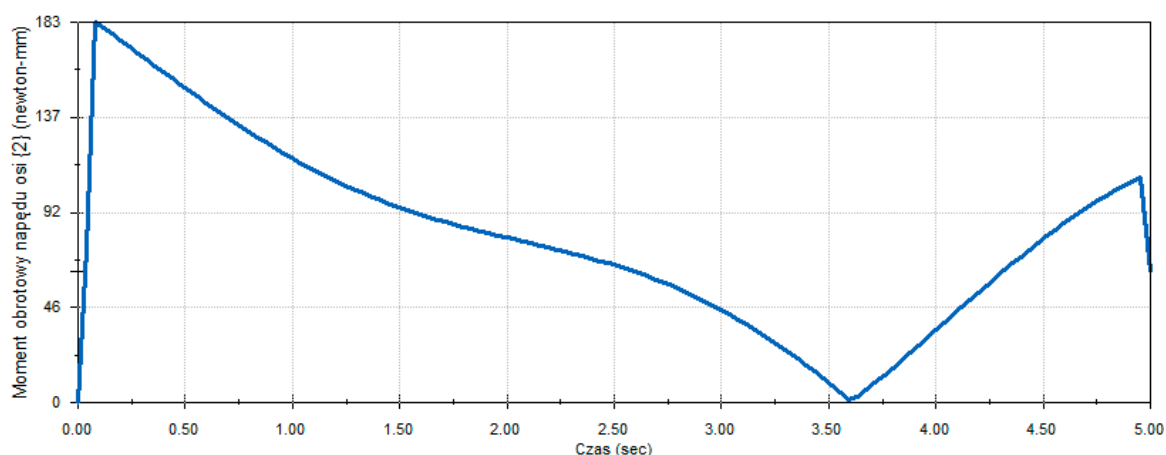
Rys.7.1. Tor ruchu do analizy

Wykonano badania momentu obrotowego wymaganego do poruszania się robotem dla poszczególnych osi obrotowych robota. Rysunek 7.2 przedstawia moment obrotowy wokół osi {1} a oznaczenia są zgodne z rysunkiem 2.2.



Rys.7.2. Moment obrotowy napędu wokół osi {1}

Na rysunku 7.3 przedstawiono wynik badania momentu obrotowego wokół osi {2}



Rys.7.3. Moment obrotowy napędu wokół osi {2}

Z rysunku 7.2 wynika, że napęd odpowiadający za obrót wokół osi {1} musi mieć moment obrotowy co najmniej 0,28 Nm aby móc się poruszać natomiast z rysunku 7.3 odczytać można, że minimalny moment obrotowy wymagany, aby robot się poruszał wynosi 0,183 Nm. Należy jednak uwzględnić masę butelki, którą robot będzie przekładać, która wynosi około 0,32 kg na ramieniu 0,9m co daje dodatkowo moment obciążający o wartości 0,288 Nm.

7.2. Dobór napędu pneumatycznego

Dobór napędu pneumatycznego do obrotu wokół osi {1} zgodnie z rysunkiem 2.2 przedstawiono w tabeli 7.1.

Tab. 7.1. Propozycje napędu pneumatycznego

Model	Zakres ruchu	Minimalne ciśnienie [MPa]	Maksymalne ciśnienie [Mpa]	Moment obrotowy [Nm]	Powtarzalność pozycji końcowej
SRU-plus-D1 40-W-180-3-8-M12-AS	180°	0,4	0,8	19,2	+/-0,5°
DSMI-40-270-A-B	270°	0,2	1	20	+/- 0,3°

Wybrano napęd DSMI-40-270-A-B [9] z uwagi na większy zakres ruchu, lepszą powtarzalność pracy oraz większy moment obrotowy.

7.3. Dobór napędów elektrycznych

Dobór napędu elektrycznego{2} ukazano w tabeli 7.2.

Tab. 7.2. *Propozycje napędu elektrycznego*

Model	Moment trzymający [Ncm]	NEMA	Napięcie zasilające [VDC]	Typ silnika
PD4-C5918L4204-E-01	187	23	12-48	bez szczotkowy
PD6-C8918S6404-E-09	360	34	12-48	krokowy
PD6-CB80M048030-E-09	170	34	12-48	bez szczotkowy

Z uwagi na większą powtarzalność operacji oferowaną przez silnik krokowy jak i większy moment trzymający zdecydowano się wybrać silnik PD6-C9819S6404-E-09 [10].

Dobór napędu elektrycznego do przesuwu wzdłuż osi {3} przedstawiono w tabeli 7.3.

Tab. 7.3. *Propozycje liniowego napędu elektrycznego*

Model	Maksymalna siła [N]	NEMA	Napięcie zasilające [VDC]	Długość śruby [mm]
PSA56S-E-2-VQKE-152	351,3	23	12-48	152
LSR1601T	321,4	23	12-48	300-500

Ze względu na wystarczającą siłę, niższą cenę oraz możliwość dopasowania długości śruby na zamówienie wybrano model SLR1601T [11].

Dobór napędu do chwytaka ukazano w tabeli 7.4.

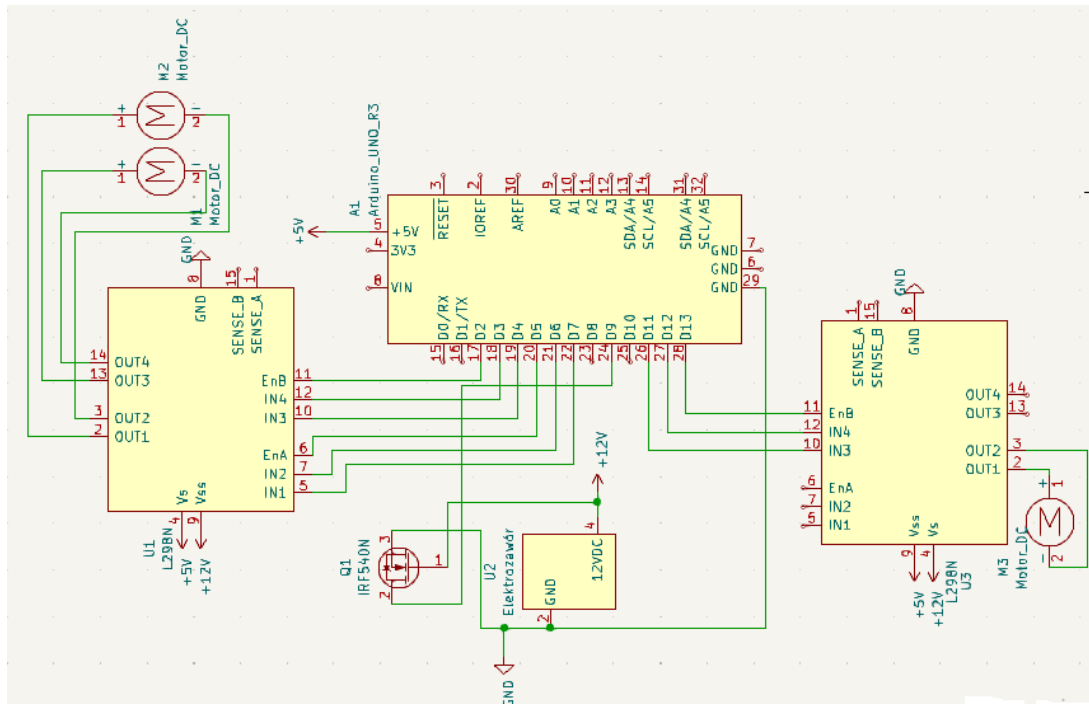
Tab. 7.4. *Propozycje silnika do chwytaka*

Model	Moment trzymający [Ncm]	NEMA	Napięcie zasilające	Typ silnika
PD2-CB	25	17	12-48	Bez szczotkowy
NEMA17	45	17	12-48	krokowy

Wybrano silnik krokowy NEMA17 [12] oferujący większy moment trzymający niż silnik bez szczotkowy. Dodatkowo pliki CAD poszczególnych napędów dostępne są bezpośrednio na witrynach internetowych producentów co będzie pomocne w dalszym projektowaniu robota.

8. Schemat połączeń

Na rysunku 8.1 pokazano schemat połączeń elementów robota. Sterowanie manipulatorem odbywa się z wykorzystaniem mikrokontrolera Arduino UNO R3. Napęd obrotowy pneumatyczny sterowany jest przy pomocy elektrozaworu służącego do kontroli ciśnienia powietrza a tym samym do obrotu wokół osi 1. Wszystkie serwonapędy są sterowane przy użyciu sterownika L298N.



Rys.8.1. Schemat połączeń elementów robota

9. Szacunkowe zestawienie kosztów

W tabeli 9.1 przedstawiono szacunkowe zestawienie kosztów elementów robota.

Tab. 9.1. Zestawienie kosztów

Produkt	Cena [zł]
DSMI-40-270-A-B	2460
PD6-C8918S6404-E-09	1200
LSR1601T	800
NEMA17	70
Materiały konstrukcyjne	600
Arduino UNO R3	120
Elektronika	100

Szacunkowy całkowity koszt robota wynikający z tabeli 9.1 to około 5350 złotych.

10. Wnioski końcowe

Praca nad projektem robota typu OOP przyniosła wiele wniosków, które są kluczowe dla skutecznej realizacji całego projektu. Pierwszym istotnym wnioskiem jest znaczenie zastosowania odpowiedniej metodyki pracy, obejmującej etapy projektowania, implementacji i weryfikacji. Systematyczne podejście do pracy i szczegółowe planowanie umożliwiają uniknięcie potencjalnych problemów oraz zapewniają osiągnięcie zamierzonych celów. Kolejnym istotnym wnioskiem jest istota opisu robota i związanej z nim kinematyki. Wcześniejsze określenie funkcjonalności manipulatora umożliwia pełne zrozumienie specyfikacji projektu natomiast sporządzenie opisu kinematyki, czyli sposobu ruchu i zdolności przemieszczania się robota jest niezwykle istotne do jego skutecznego projektowania. Wykonanie zadań kinematyki prostej oraz odwrotnej jest kluczowe dla prawidłowego określenia zdolności robota do precyzyjnego poruszania się i manipulowania przedmiotami.

Weryfikacja zadań kinematyki robota jest niezwykle ważna, w tym celu wykorzystano symulacje, które stanowią skuteczne narzędzie do weryfikacji wyników, umożliwiając potwierdzenie poprawności projektu. Analiza wyników symulacji w porównaniu z oczekiwanymi rezultatami pozwala na identyfikację ewentualnych rozbieżności oraz wprowadzenie odpowiednich poprawek do projektu. Dzięki symulacji możliwe jest także ocenienie zachowania robota w różnych scenariuszach, co przyczynia się do doskonalenia projektu i zoptymalizowania jego funkcjonalności.

Odpowiednie założenia konstrukcyjne stanowią kolejny kluczowy element w realizacji projektu. Dobór odpowiednich napędów pneumatycznych i elektrycznych ma istotny wpływ na zdolności ruchowe robota, takie jak prędkość, siła i precyzja. Uwzględnienie tych założeń w projekcie robota pozwala na optymalne dostosowanie konstrukcji do wymagań projektowych i zapewnienie osiągnięcia zamierzonych celów.

Bibliografia

- [1] Hughes C., Hughes T., Programowanie robotów: sterowanie pracą robotów autonomicznych. Helion, Gliwice 2017.
- [2] Kaczmarek W., Panasiuk J., Borys S., Dyczkowski R., Siwek M., Robotyzacja i automatyzacja przemysł 4.0. Wydawnictwo Naukowe PWN, Warszawa 2023.
- [3] Szkodny T., Zbiór zadań z podstaw robotyki. Wydawnictwo Politechniki Śląskiej, Gliwice 2013.
- [4] Szkodny T., Kinematyka robotów przemysłowych. Wydawnictwo Politechniki Śląskiej, Gliwice 2013.
- [5] Kusigerski B., Pandilov Z., Kusigerski G., Design and practical realization of a SCARA robot. *Acta Technica Corvininensis*, 15, 2022, 67-73.
- [6] Yamafuji K., Development of SCARA robots. *Journal of Robotics and Mechatronics*, 31, 2019, 10-15.
- [7] Ibrahim B. S. K. K., Ahmed M. A. Z., Modelling and control of SCARA manipulator. *Procedia Computer Science*, 42, 2014, 106-113.
- [8] Wen-Chuan Ch., Chin-Sheng Ch., Feng-Chi L., Ying-Shieh K., FPGA-realization of the kinematics IP for SCARA robot. *Microsystem technologies: sensors, actuators, systems integration*, 27, 2021, 1075-1090
- [9] <https://air-com.pl/p/naped-wahadlowy-dsmi-40-270-a-b-561691-festo,10207,286223>, 11.05.2023 r.
- [10] <https://en.nanotec.com/products/1906-pd6-c-stepper-motor-with-integrated-controller-nema-34>, 11.05.2023 r.
- [11] <https://us.rs-online.com/product/pbc-linear/lsr1601t-0300-rcf-231-nnn-n/72265292/>, 11.05.2023 r.
- [12] <https://www.pololu.com/product/2267>, 11.05.2023 r.