

**KATEDRA MECHANIKI TEORETYCZNEJ I STOSOWANEJ**  
**WYDZIAŁ MECHANICZNY TECHNOLOGICZNY**  
**POLITECHNIKA ŚLĄSKA**

**Projekt inżynierski**

Projekt i budowa ramienia robota sterowanego mikroprocesorowo

Autor: Paweł Misztal


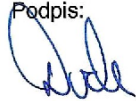
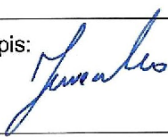
Kierunek studiów: Mechatronika

Prowadzący pracę: Dr hab. Inż. Sławomir Kciuk prof. PŚ

Opiekun: Dr inż. Mariola Jureczko

Gliwice, luty 2024



POLITECHNIKA ŚLĄSKA					
KATEDRA MECHANIKI TEORETYCZNEJ I STOSOWANEJ					
WYDZIAŁ MECHANICZNY TECHNOLOGICZNY					
PROJEKT INŻYNIERSKI					
Student: Paweł Misztal	Kierunek: Mechatronika	Poziom, profil i forma studiów: Studia pierwszego stopnia, profil ogólnoakademicki, stacjonarne	Semestr: VII	Ścieżka dyplomowania/specjalność: -	Rok akademicki: 2023/2024
<p><b>Temat pracy:</b> Projekt i budowa ramienia robota sterowanego mikroprocesorowo</p> <p><b>Thesis title:</b> Design and construction of a microprocessor-controlled robot arm</p> <p><b>Zakres pracy:</b></p> <ul style="list-style-type: none"> <li>- Dokonać przeglądu obecnego stanu wiedzy i rozwiązań technicznych dotyczących tematu projektu.</li> <li>- Przyjąć założenia projektowe.</li> <li>- Zaprojektować ramię robota sterowanego mikroprocesorowo.</li> <li>- Wykonać ramię robota sterowanego mikroprocesorowo.</li> <li>- Przeprowadzić testy poprawności działania wykonanego ramienia robota.</li> <li>- Opracować wyniki pracy i wnioski.</li> <li>- Wykonać sprawozdanie końcowe.</li> <li>-</li> </ul> <p>Czy zakres pracy będzie obejmował informacje podlegające ochronie na podstawie przepisów o ochronie informacji niejawnych: TAK/NIE *</p> <p>Czy przedmiot pracy dyplomowej będzie objęty inną tajemnicą prawnie chronioną: TAK/NIE*</p>					
Prowadzący pracę: Dr hab. inż. Sławomir Kciuk prof. PŚ	Podpis: 	Kierownik jednostki wewnętrznej: Dr hab. inż. Sławomir Duda, prof. PŚ	Podpis: 		
Opiekun: Dr inż. Mariola Jureczko	Podpis: 	Data wydania tematu: <b>06.2023</b>	Planowany termin złożenia pracy dyplomowej: <b>02.2024</b>		



# Spis treści

Streszczenie .....	1
Summary.....	2
1. Cel i zakres pracy .....	4
2. Wstęp teoretyczny .....	6
2.1. Silniki krokowe.....	6
3. Projekt ramienia robota .....	8
3.1. Kinematyka robota.....	8
3.2. Komponenty robota .....	9
3.3. Modele CAD komponentów robota dostosowane do druku 3D.....	11
3.4. Przełożenie przekładni .....	14
3.5. Model robota w programie Inventor .....	15
3.6. Model chwytaka w programie Inventor .....	15
3.7. Projekt układu elektrycznego.....	16
4. Wykonanie ramienia robota.....	19
5. Oprogramowanie .....	21
5.1. Analizator komend.....	21
5.2. Kinematyka prosta i odwrotna.....	22
5.3. Implementacja interpolacji liniowej .....	26
5.4. Transformacja układu współrzędnych .....	27
5.5. Dedykowane oprogramowanie do sterowania robotem.....	29
6. Wyniki przeprowadzonych testów .....	31
7. Podsumowanie.....	32
Bibliografia.....	i
Spis rysunków .....	iii



---

# Streszczenie

W niniejszej pracy został przedstawiony proces projektowania ramienia robota, jego wykonanie, tworzenie oprogramowanie oraz wyniki przeprowadzonych testów.

Przy projektowaniu ramienia robota skupiano się na dostosowaniu modelu pod wytwarzanie metodami przyrostowymi i wykorzystaniu odpowiednich materiałów. Ważną kwestią było też obliczenie przełożenia przekładni czy wybranie odpowiedniego modelu kinematycznego robota do powierzonego zadania. Zaprojektowano również odpowiedni układ elektroniczny sterowania ramieniem robota.

W ramach pracy pokazano efekty wykonania ramienia robota technikami przyrostowymi oraz jego montażu. Zaprezentowano również wykonanie układu elektronicznego.

W opisie oprogramowania przybliżono sposób analizowania komend poprzez mikrokontroler robota z wykorzystaniem komunikacji przy pomocy portu szeregowego. Przedstawiono zarówno sposób obliczania kinematyki prostej i odwrotnej, jak i obliczenia transformacji układu współrzędnych. Pokazano sposób implementacji interpolacji liniowej toru ruchu ramienia robota.

**Słowa kluczowe:** ramię robota, mikrokontroler, kinematyka odwrotna, druk 3D

---

# Summary

In this thesis, the process of designing a robot arm, its fabrication, software development, and the results of conducted tests have been presented.

The design of the robot arm focused on designing the model for incremental manufacturing methods and utilizing suitable materials. Calculating the gear ratio and choosing the appropriate kinematic model for the assigned task were also crucial aspects. An appropriate electronic control system for the robot arm was also designed.

The thesis showcases the results of constructing the robot arm using incremental techniques and its assembly. The implementation of the electronic control system is also demonstrated.

In the software description, the method of analyzing commands through the robot's microcontroller using communication via a serial port is outlined. Both forward and inverse kinematics calculations, as well as coordinate system transformation computations, are presented. The implementation of linear interpolation for the motion trajectory of the robot arm is also showcased.

**Key words:** robot arm, microcontroller, inverse kinematics, 3D printing





# 1. Cel i zakres pracy

Celem pracy jest zaprojektowanie, wykonanie i zaprogramowanie, a następnie przetestowanie wykonanego ramienia robota sterowanego mikroprocesorowo. W pierwszym etapie należy zaprojektować ramię robota, dobrać jego wymiary, określić kształt i rodzaj użytych materiałów. W drugim etapie należy wykonać ramię robota według opracowanego projektu. W trzecim etapie będzie wykonywane oprogramowanie robota na mikrokontroler oraz komputer osobisty. W ostatnim etapie wykonany robot zostanie poddany testom sprawdzającym jego sprawność.

W celu późniejszej oceny jakości wykonania projektu konieczne jest wcześniejsze określenie założeń tego, jakie wymagania ma spełniać dany projekt. Dlatego w przypadku tego projektu postawiono następujące wymagania:

- ramię robota ma mieć możliwość przenoszenia obiektów o masie do 100g,
- orientacja obrotu przenoszonych obiektów nie jest wymagana,
- zakres pracy ramienia robota ma pozwalać na poruszanie się kiści robota w polu roboczym o wymiarach 300x200 [mm] oraz na przeniesieniu złapanych obiektów poza wyznaczone pole robocze,
- komunikacja z robotem ma polegać na przesyłaniu komend z komputera osobistego przy pomocy portu szeregowego,
- układ sterowania robota ma mieć możliwość wyznaczenia lokalnych układów współrzędnych,
- układ sterowania ma pozwalać na poruszanie kiścią robota przy pomocy kinematyki odwróconej,
- układ sterowania ma pozwalać na liniowy ruch kiści.



## 2. Wstęp teoretyczny

Ramię robota może być poruszane wieloma różnymi rodzajami napędów, aktualnie najlepsze osiągi oferują serwonapędy, jednak przez ich wysoką cenę nie będą one wykorzystane w tej pracy. Alternatywą mogą się okazać silniki krokowe, które zostaną opisane w następnej części.

### 2.1. Silniki krokowe

Gdy potrzebna jest możliwość dokładnego pozycjonowania silnika elektrycznego, przy jednoczesnym prostym sterowaniu, rozwiązaniem jest zastosowanie silników krokowych, zwanych też silnikami skokowymi. Silniki te charakteryzują się możliwością zmiany położenia wału silnika o znaną wartość, dzięki czemu nie ma konieczności stosowania układów sprzężenia zwrotnego. Jest to zaleta tych silników, ponieważ sterowniki tych silników są zarówno dużo prostsze w obsłudze, jak i tańsze. Natomiast wadą tych silników jest to, że przy ich przeciążeniu obracają się one o nieznaną wartość. Przez to, że nie mają one układu sprzężenia zwrotnego, to nie mamy informacji o tym, kiedy został taki silnik przeciążony. Dlatego silniki te należy dobierać z zapasem momentu obrotowego względem warunków brzegowych, aby nie doszło do przypadkowego zgubienia położenia początkowego [1].

Silniki krokowe można podzielić na dwie kategorie, unipolarne oraz bipolarne. Pierwsze charakteryzują się gorszym momentem obrotowym względem silników bipolarnych, jednak sposób ich sterowania jest dużo prostszy. Zaletą silników unipolarnych jest to, że podłączając je odpowiednio do sterownika można otrzymać silniki bipolarne. Taka modyfikacja w drugą stronę nie jest możliwa. Prostym a zarazem szybkim sposobem na sprawdzenie z jakim typem silnika ma się do czynienia jest sprawdzenie ilości wyprowadzonych przewodów z silnika. Gdy jest ich 4, to jest to silnik bipolarny, gdy 6 to unipolarny [1].

Silniki krokowe posiadają charakterystyczne parametry. Jednym z nich jest działka elementarna oznaczana  $\phi$ , czyli liczba kroków na obrót silnika. Najczęściej wartość tego parametru wynosi 200 kroków/obrót, co przekłada się na  $1.8^\circ$ , gdzie jest to często niewystarczająca dokładność do wielu zadań. Jednym ze sposobów na radzenie sobie z tym problemem jest stosowanie mikrokroków, lub też zwanego inaczej sterowania amplitudowego. Sposób ten polega na sterowaniu amplitudami, nie tylko w sposób zero jedynkowy, ale też poprzez możliwość podania na uzwojenia silnika np. połowy maksymalnej amplitudy. Mikrokroki mogą być wielkości  $1/2$ ,  $1/4$ ,  $1/8$  itd. Przykładowy poprzedni silnik skokowy o wartości 200 skoków/obrót po sterowaniu mikro krokowym  $1/16$  pozwala na osiągnięcie teoretycznej dokładności pozycjonowania na poziomie  $0.1125^\circ$ , co jest znacznie większą dokładnością [1].

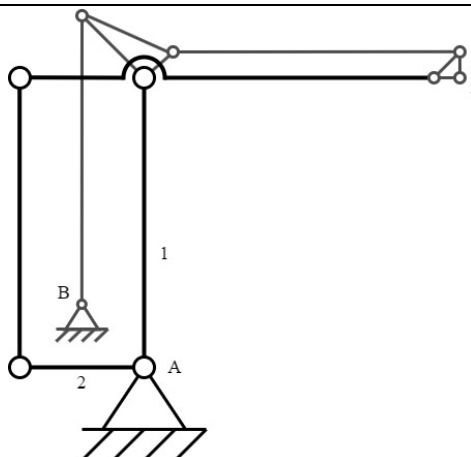
## 3. Projekt ramienia robota

W tym rozdziale opisano na początku wybór odpowiedniej kinematyki robota. W kolejnym etapie dokonano wyboru odpowiednich komponentów do budowy robota. Później opisano sposoby dostosowywania modeli CAD pod technologię wytwarzania przyrostowego. Dokonano również niezbędnych obliczeń do wyznaczenia przełożenia przekładni robota. Na końcu przybliżono efekty projektowania modelu CAD robota oraz jego układu sterowania.

### 3.1. Kinematyka robota

W celu przenoszenia obiektów, gdzie ich obrót nie jest wymagany, robot powinien posiadać minimum 3 stopnie swobody. Jest to podyktowane poruszaniem elementu w wyznaczonym polu roboczym, gdzie przenoszony obiekt musi zostać przesunięty w dwóch osiach oraz podniesiony na odpowiednią wysokość. Kolejnym wymogiem stawianym przed robotem było stosunkowo duże pole robocze, ponieważ oprócz poruszania się kiści robota w polu o wymiarach 300x200 [mm], robot miał mieć możliwość przenoszenia elementów poza wyznaczony obszar. Z tego względu dokonano wybór konstrukcji robota jako robota antropomorficznego, który charakteryzuje się większym polem roboczym niż zajmowana przez niego podstawa. Następnie zdecydowano się na konstrukcję robota tzw. „do paletyzacji”, która charakteryzuje się mechanicznym unieruchomieniem czwartej osi, dzięki czemu do sterowania położeniem kiści robota potrzebne są tylko 3 silniki. Dzięki zastosowaniu odpowiedniej geometrii ramion możliwa jest równoległa praca kiści robota względem podstawy. Dodatkowym atutem tego rozwiązania jest możliwość umieszczenia silników w podstawie robota, co pozwala na zredukowanie masy ramion.

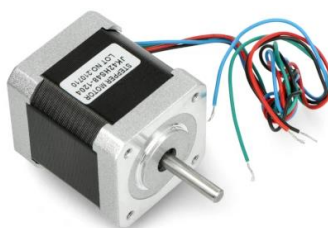
Na rysunku 1 przedstawiono model kinematyczny robota, gdzie literą A oznaczono główną podstawę robota, stanowiącą podporę całego robota, z której będą przekazywane momenty obrotowe odpowiednio na ramiona 1 i 2. Z podstawy oznaczonej B będzie wychodzić wodzik stabilizujący obrót kiści robota oznaczonej numerem 3.



Rysunek 1 Model kinematyczny robota

## 3.2. Komponenty robota

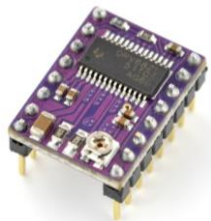
Kluczowym elementem pozwalającym na poruszanie się robota jest jego element napędowy. W tym celu konieczny jest dobór odpowiedniego silnika. Ze względu na łatwość sterowania i niskie koszty zdecydowano się na wybór silników krokowych jako głównych silników zapewniających możliwość zmiany pozycji robota. Modelem silnika, na który się zdecydowano jest silnik JK42HS48-1204, który przedstawiono na rysunku 2. Jest to silnik bipolarny w obudowie Nema 17, charakteryzujący się 200 krokami na obrót, momentem na poziomie 0,47Nm i maksymalnym prądem zasilania 1,2A [2].



Rysunek 2 Silnik krokowy JK42HS48-1204 [2]

Ze względu na duży pobór prądu przez dobrany silnik krokowy nie jest możliwe sterowanie nim bezpośrednio z mikrokontrolera. Z tego powodu konieczne jest dokonanie wyboru odpowiedniego sterownika silnika krokowego. Wybrany został sterownik silnika krokowego oparty na układzie DRV8825, który przedstawiono na rysunku 3. Sterownik ten pozwala na obsługę silników krokowych o poborze prądu do 2,2A na cewkę przy zastosowaniu chłodzenia oraz może być zasilany napięciem do 24V. Dodatkowo posiada

możliwość pracy mikro krokowej do 1/32 i może być sterowany przy pomocy mikrokontrolerów o napięciu 3,3V i napięciu 5V [3].



Rysunek 3 Sterownik silnika krokowego DRV8825 [3]

W celu przytrzymania elementu przez chwytak konieczna jest możliwość zmiany jego rozwarcia. Do tego celu można zastosować wiele rodzajów aktuatorów, zasilanych różnymi rodzajami energii m.in.: energią elektryczną, pneumatyczną czy hydrauliczną. Ze względu na prostotę sterowania, zasilania i niską cenę zdecydowano się na wykorzystanie serwa modelarskiego jako elementu zmieniającego rozstaw chwytaka. Na rysunku 4 przedstawiono wybrane serwo MG-90S, które charakteryzuje się momentem 0,2Nm przy napięciu zasilania 4,8V [4].



Rysunek 4 Serwo MG-90S [4]

Elementem sterującym wszystkim podzespołami robota będzie mikrokontroler. Zdecydowano się na mikrokontroler ESP32-S2-WROWER, który przedstawiono na rysunku 5. Wybór ten został podyktowany głównie mocą obliczeniową urządzenia, ponieważ procesor może pracować z taktowaniem do 240MHz, a wysoka wydajność procesora będzie niezbędna w ciągłym wykonywaniu skomplikowanych obliczeń matematycznych związanych z zadaniem kinematyki odwrotnej robota [5].



Rysunek 5 Mikrokontroler ESP32-S2-WROWER [5]

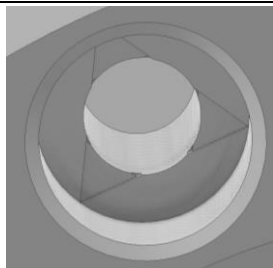


### **3.3. Modele CAD komponentów robota dostosowane do druku 3D**

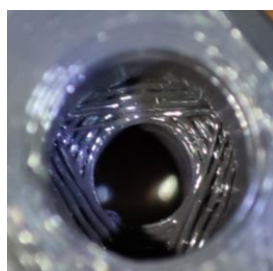
Do wykonania elementów ramienia robota konieczny był wcześniejszy wybór metody sposobu wykonania. Ze względu na dostępność, szybkość oraz stosunkowo niskie koszty produkcji zdecydowano się na wykonanie elementów robota metodą druku 3D w technologii FDM/FFM. Jest to obecnie najczęściej stosowana metoda przyrostowa [6].

W tej technologii druku dostępne jest szerokie spektrum rodzajów stosowanych filamentów. Konieczne jest wybranie odpowiedniego rodzaju do odpowiedniego zadania przy wykorzystaniu posiadanych możliwości. Wybrano filamenty PLA oraz PET-G. Filament PLA charakteryzuje się dużą sztywnością, niską temperaturą topnienia i łatwością druku, dlatego z tego tworzywa wykonano większość elementów konstrukcyjnych robota. Elementy narażone na kontakt z elementami o podwyższonej temperaturze, takimi jak silniki krokowe, zostały wykonane z PET-G. Filament ten posiada mniejszą sztywność i łatwość druku w porównaniu do PLA, ale jego temperatura mięknięcia jest wyższa i wynosi 78°C, gdzie temperatura mięknięcia PLA to 55°C [7] [8].

Przy projektowaniu elementów pod druk 3D konieczne jest odpowiednie zaprojektowanie geometrii w celu zapewnienia zarówno optymalnego procesu wydruku, jak i odpowiedniej wytrzymałości wydrukowanych komponentów. Przy projektowaniu elementów pod druk 3D powszechną praktyką jest projektowanie modelu tak, żeby ilość koniecznych podpór była jak najmniejsza. Czasami konieczne jest wydrukowanie osadzenia na np. łeb śruby walcowej od dołu modelu. W takim wypadku konieczne byłoby stosowanie podpór w celu poprawnego wydrukowania elementu. Drukarki 3D w technologii FDM potrafią dość dobrze drukować fragmenty zwane „mostami” czyli ciągle linie proste drukowane w powietrzu. Dlatego w tego typu otworach można zastosować specjalny rodzaj geometrii, gdzie w okrąg wrysowuje się trójkąt i go wyciąga o wysokość drukowanej warstwy, tak jak to pokazano na rysunku 6. Pozwala to wydrukować akceptowalnej jakości otwory w powietrzu bez użycia podpór, a efekty tego zostały pokazane na rysunku 7 [9].



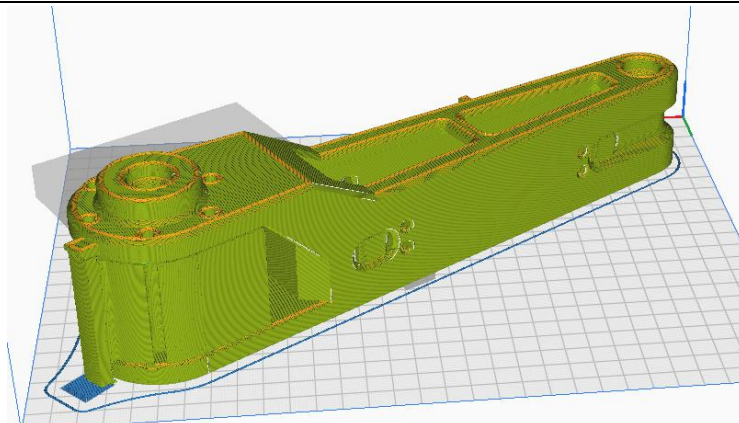
Rysunek 6 Geometria otworu drukowanego od spodu



Rysunek 7 Geometria wydrukowanego otworu od spodu

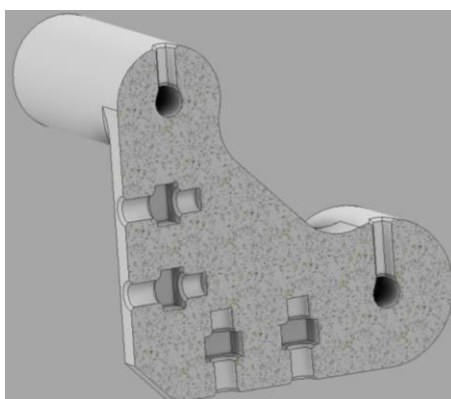
Kolejnym ważnym elementem przy projektowaniu elementów pod druk 3D jest projektowanie ich kształtu z myślą, że będą one drukowane w pewnej orientacji, a co za tym idzie warstwy wydrukowanego modelu będą ułożone w określony sposób, taki w jaki model został ułożony na drukarce 3D. Przez to, że wydruki elementów w technologii FDM charakteryzują się mniejszą wytrzymałością na rozciąganie w kierunku prostopadłym do warstw niż na rozciąganie w kierunku wzdłuż ułożenia warstw druku, to konieczne jest odpowiednie ustawienie modelu podczas druku [10]. Jako przykład poprawnego projektowania uwzględniającego ułożenie warstw druku przedstawiono projekt i ułożenie elementu ramienia głównego robota. Jak pokazano na rysunku 8 model ten ułożono w taki sposób, aby osie były prostopadłe do warstw wydruku. W ten sposób otrzymano optymalną wytrzymałość elementu na rozciąganie w najbardziej krytycznych kierunkach.

Przy projektowaniu kiści ramienia robota zastosowano również technikę osadzania elementów w wydruku. Proces ten polega na zaprojektowaniu elementu w taki sposób, aby móc zatrzymać wydruk na pewnej warstwie, ułożyć element w wydruku, a następnie kontynuować drukowanie [11]. Metoda ta była konieczna przy projektowaniu kiści robota, ponieważ do kiści robota będą mocowane różnego rodzaju końcówki. Możliwe jest wydrukowanie zadowalającej jakości otworu z gwintem, jednak tego typu rozwiązanie skutkuje szybkim zużywaniem się wydrukowanego gwintu. Dlatego zdecydowano się na zaprojektowanie kiści robota w taki sposób, że w trakcie wydruku 3D zostaną osadzone w nim cztery nakrętki M3, co pozwoli na wielokrotne przykręcanie i odkręcanie końcówek.



Rysunek 8 Ułożenie ramienia głównego robota w programie Cura

Na rysunku 9 pokazano przekrój projektu kiści robota. Posiada on w środku miejsca na nakrętki, a osadzenie nakrętek jest możliwe dzięki zatrzymaniu wydruku na określonej warstwie, co pokazano na rysunku 10.



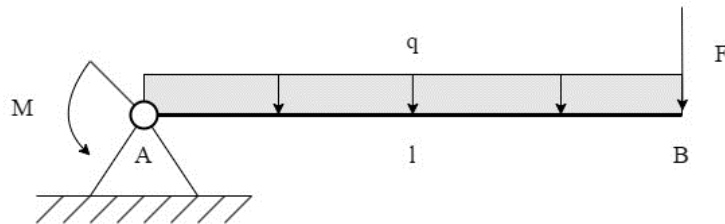
Rysunek 9 Przekrój kiści robota



Rysunek 10 Zatrzymany wydruk kiści robota z osadzonymi nakrętkami

### 3.4. Przełożenie przekładni

Po dobraniu silników krokowych konieczne było obliczenie przełożenia przekładni, w celu zapewnienia odpowiedniego momentu obrotowego do sterowania położeniem ramienia robota. W tym celu wykonano obliczenia statyczne. Na rysunku 11 przedstawiono schemat układu. Z założeń wynika, że na końcu ramienia powinna być siła skupiona o wartości  $F = 1[\text{N}]$  przyłożona w punkcie **B**. Dodatkowymi założeniami jest długość ramienia, która wynosi  $l = 0,5[\text{m}]$  oraz obciążenie ciągłe o wartości  $q = 12 [\text{N/m}]$ , która odpowiada rozkładowi masy ramienia. Poszukiwaną wartością była wartość momentu obrotowego w podstawie **A**. W równaniu (1) przedstawiono wzór sumy geometrycznej momentów wszystkich sił działających na układ względem punktu **A**. W równaniu (2) wyznaczono wartość momentu obrotowego  $M$ , która dla przyjętych wartości wyniosła  $2 [\text{Nm}]$ . W równaniu (3) obliczono minimalne przełożenie  $n$ , aby układ zachował stabilność. Do obliczenia przełożenia posłużono się wcześniej obliczonym momentem obrotowym  $M$  oraz wartością osiąganego momentu obrotowego silnika krokowego.



Rysunek 11 Uproszczony model ramienia z obciążeniem

$$\sum M_{iA} = M + Fl + \frac{ql^2}{2} = 0 \quad (1)$$

$$M = Fl + \frac{ql^2}{2} = 1 \cdot 0,5 + \frac{12 \cdot 0,5^2}{2} = 2 [\text{Nm}] \quad (2)$$

$$n = \frac{M}{M_s} = \frac{2}{0,47} = 4,25 \quad (3)$$

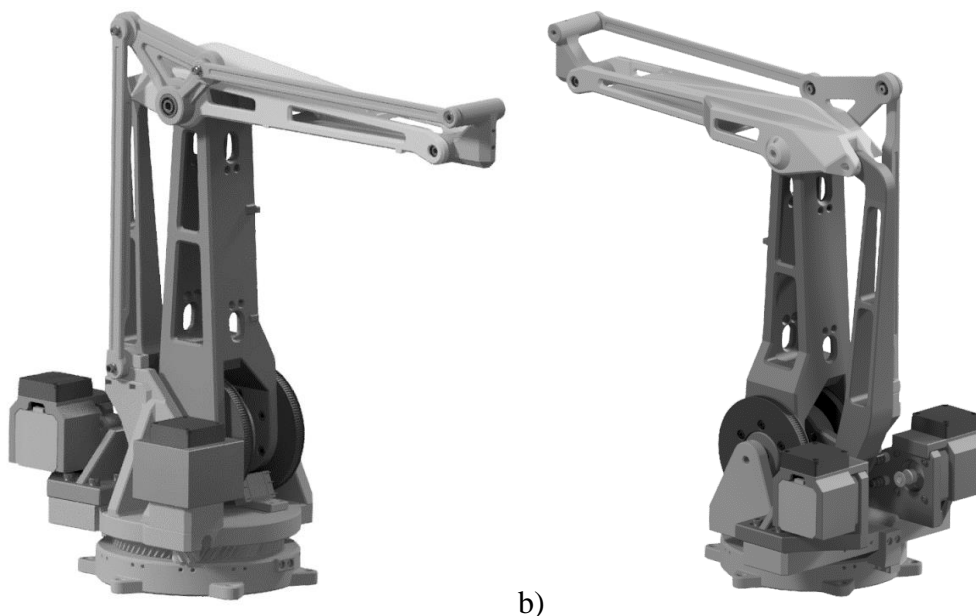
gdzie:

$M$  – moment obrotowy niezbędny do utrzymania ramienia w równowadze,

$M_s$  – moment obrotowy wybranego silnika krokowego.

### 3.5. Model robota w programie Inventor

Po dobraniu komponentów robota i wykonaniu niezbędnych obliczeń można było wykonać projekt ramienia robota w oprogramowaniu wspomagającym projektowanie. W tym celu wykorzystano oprogramowanie Inventor. Na rysunku 12 pokazano wygląd zaprojektowanego ramienia robota z różnych perspektyw. Przy projektowaniu ramienia robota starano maksymalizować wykorzystanie połączeń rozłącznych śrubowych. Dodatkowo przy osiach robota wykorzystywano odpowiednie łożyska toczne w celu zapewnienia niskich strat w wyniku tarcia. Dodatkowo starano się dzielić konstrukcję robota na moduły, czyli przykładowo silniki krokowe nie są montowane bezpośrednio do konstrukcji robota, ponieważ takie rozwiązanie mocno utrudnia potencjalne sposoby ulepszenia komponentów w przyszłości.

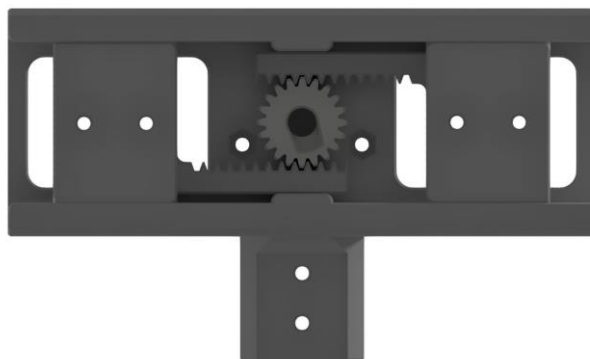


Rysunek 12 Widok robota: a) z przedniego lewego boku b) z tylnego prawego boku

### 3.6. Model chwytaka w programie Inventor

Dodatkowym elementem, który został zaprojektowany do ramienia robota jest chwytak. Jako element napędzający szczęki chwytaka wybrano serwo modelarskie. Serwo przy pomocy koła zębatego przekazuje moment obrotowy na zębatki szczęk chwytaka, dzięki

temu uzyskano liniowy ruch szczęk. Na rysunku 13 pokazano wygląd zaprojektowanego mechanizmu chwytaka.

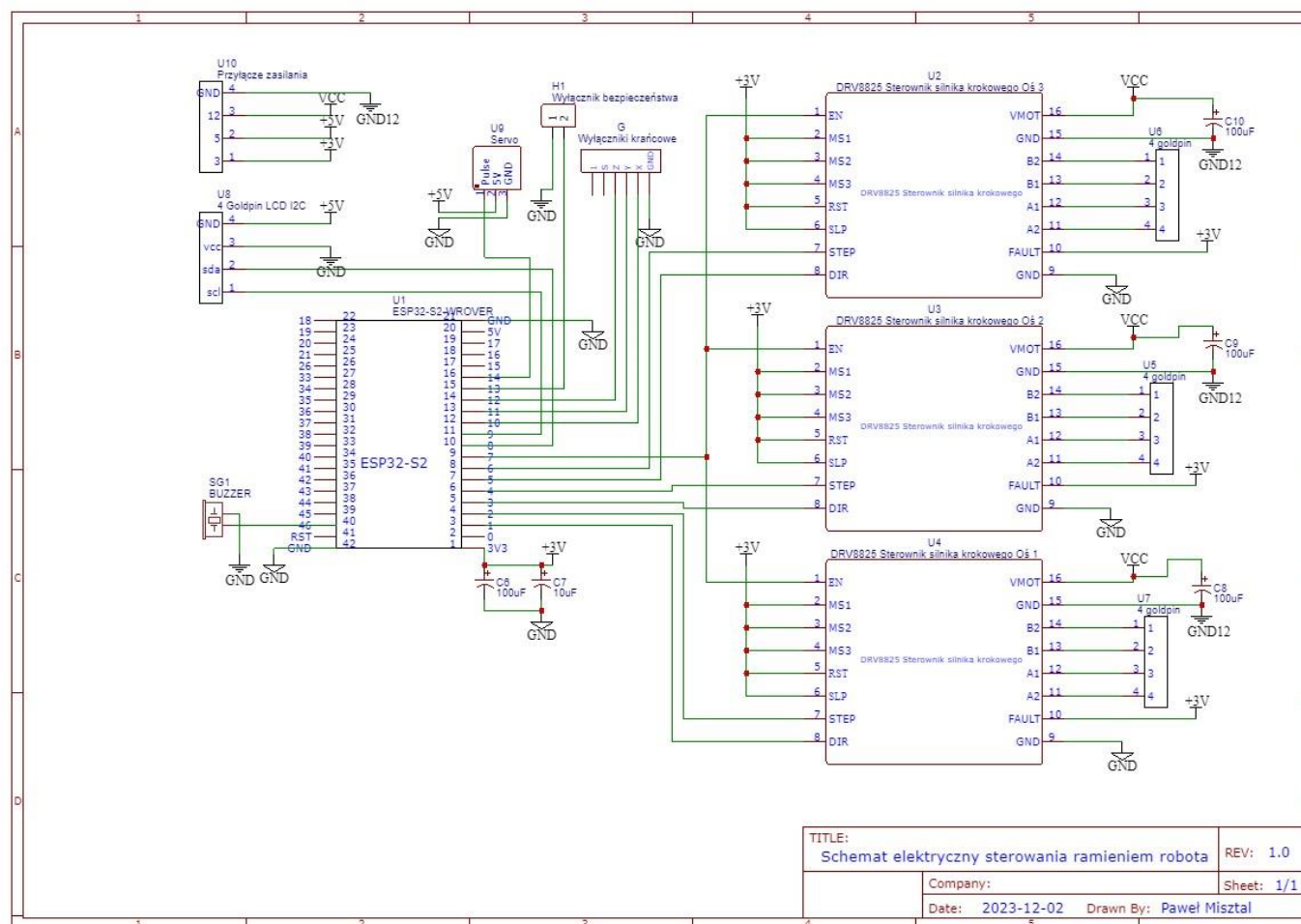


Rysunek 13 Mechanizm chwytaka

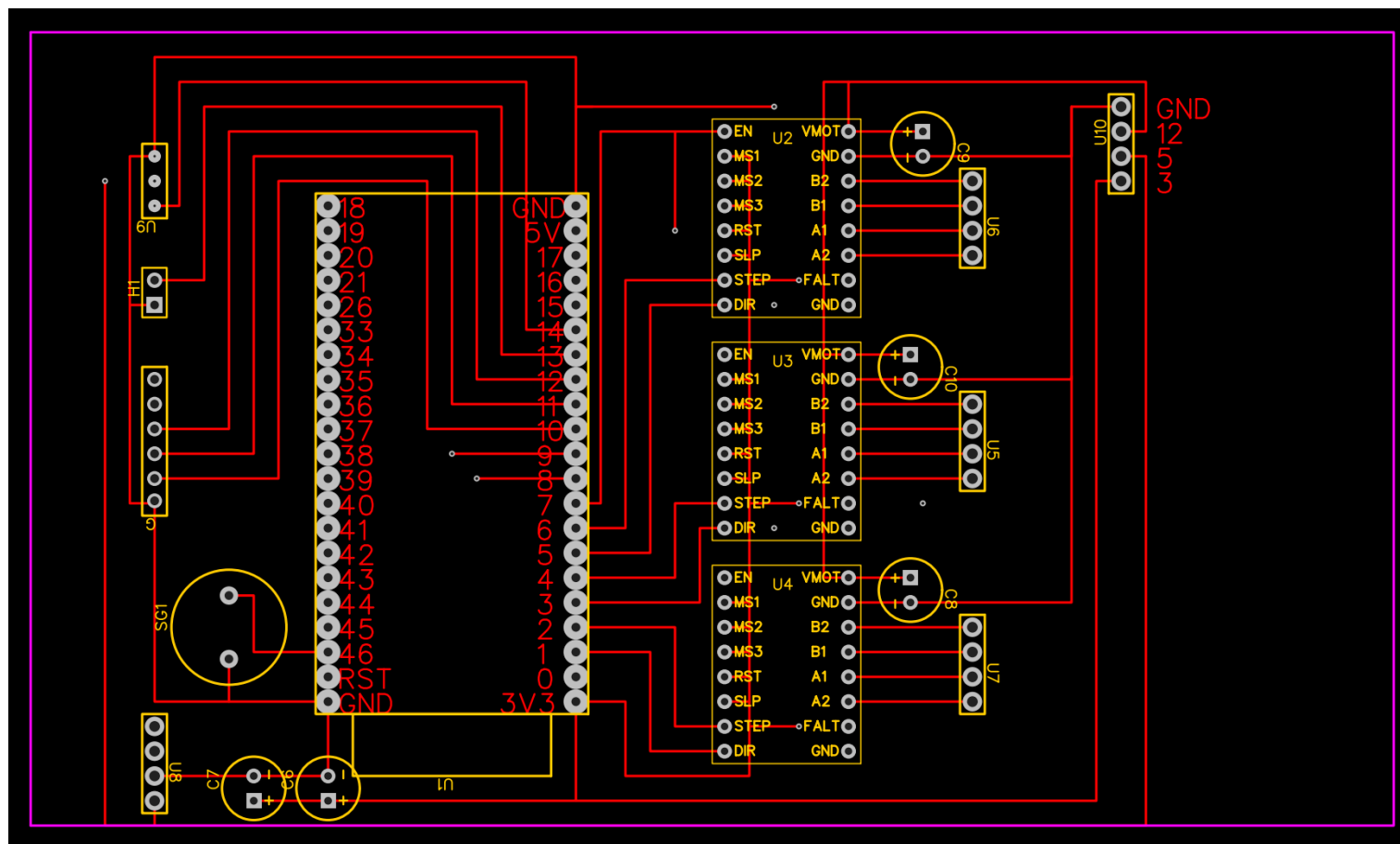
### 3.7. Projekt układu elektrycznego

W celu kontrolowania elementów wykonawczych ramienia robota konieczne było zaprojektowanie odpowiedniego schematu elektrycznego sterowania. Schemat ten przedstawiono na rysunku 14, przedstawia on połączenie mikrokontrolera ESP32S2 oznaczonego U1 ze sterownikami silników krokowych DRV8825 oznaczonymi U2,3,4 oraz innymi modułami, takimi jak listwy przyłączeniowe do serwa modelarskiego, wyłącznika bezpieczeństwa, wyłączników krańcowych, brzęczyka oraz złącz zasilania układu. Układ ten ponadto został wyposażony w odpowiednie kondensatory stabilizujące napięcie na sterownikach silników krokowych, zamontowanie tych kondensatorów oznaczonych C8,9,10 jest zalecane przez producenta sterowników silników krokowych [12]. Przy projektowaniu układu elektrycznego wykorzystywano oprogramowanie easyEDA.

Zaprojektowany układ elektryczny został wykonany na uniwersalnej płytce perforowanej. W tym celu konieczne było zaprojektowanie położenia ścieżek oraz położenia elementów. Zaprojektowaną płytkę elektryczną przedstawiono na rysunku 15.



Rysunek 14 Schemat elektryczny układu sterowania



Rysunek 15 Projekt płytki elektrycznej

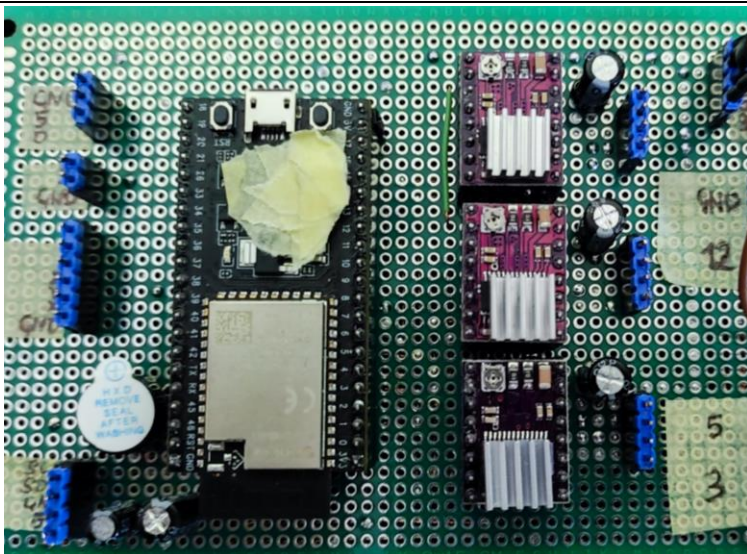


## 4. Wykonanie ramienia robota

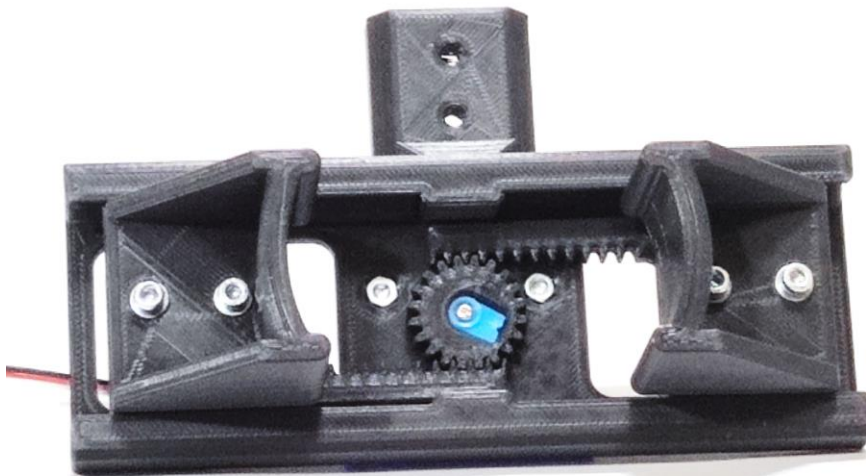
Po odpowiednim zaprojektowaniu ramienia robota możliwe było jego wykonanie. Wydrukowano wszystkie niezbędne elementy ramienia robota oraz dokonano ich złożenia. Złożony robot przedstawiono na rysunku 16. Wcześniej zaprojektowany układ sterowania ramienia robota przedstawiono na rysunku 17. Na rysunku 18 pokazano wydrukowany oraz złożony chwytak robota.



Rysunek 16 Wykonane oraz złożone ramie robota



Rysunek 17 Wykonany układ sterowania robota



Rysunek 18 Wykonany oraz złożony chwytak robota

## 5. Oprogramowanie

Aby mieć możliwość sterowania położeniem ramienia robota konieczne jest wykonanie odpowiedniego oprogramowania. Program miał spełniać założenia, że będą do niego wysyłane komendy przy pomocy portu szeregowego. Te komendy mają sterować położeniem ramion oraz sterować peryferiami i zwracać informacje o stanie robota. Dodatkowo konieczne jest wykonanie odpowiednich obliczeń kinematyki odwrotnej ramienia robota w celu łatwego sterowania położeniem kiści. Pole robocze robota często nie pokrywa się z globalnym układem odniesienia, dlatego w tym celu konieczne było zaimplementowanie odpowiedniego algorytmu definiowania oraz transformacji lokalnego układu współrzędnych do globalnego.

### 5.1. Analizator komend

Podczas pracy robota konieczna jest możliwość sprawnej komunikacji z innymi urządzeniami sterującymi pracą ramienia. W tym celu wykonano nieblokującą klasę analizującą przychodzące komendy z portu szeregowego. Konieczność wykonania nieblokującej komendy była podyktowana koniecznością ciągłego, nieprzerwanego wysyłania impulsów do sterowników silników krokowych, ponieważ typowy sposób odczytu danych z portu szeregowego zatrzymuje program do czasu odebrania wszystkich znaków. Poniżej na rysunku 19 przedstawiono fragment pseudokodu funkcji *Tick()* odpowiedzialnej za zbieranie znaków z portu szeregowego.

```

void Tick()
{
    if(Serial.available() > 0) // czy w buforze komunikacji szeregowej jest więcej niż 0 znaków
    {
        char data = Serial.read(); // odczytaj znak z komunikacji szeregowej

        if(data == '\n') // jeżeli znak jest specjalnym znakiem następnej linii
        {
            ParseCommand(buffer); // to analizuj komendę
            buffer = ""; // to wyczyść bufor
        }
        else // w innym wypadku
        {
            buffer += data; // dodaj odczytany znak do końca buforu
        }
    }
}

```

Rysunek 19 Fragment pseudokodu nieblokującej funkcji zbierającej znaki z portu szeregowego

Po wcześniejszym zebraniu wszystkich znaków komendy z portu szeregowego możliwe jest odpowiednia analiza linii komendy. W tym celu napisano funkcję *ParseCommand()*, która przyjmuje jako argument ciąg znaków z portu szeregowego oraz wyszukuje odpowiedniego ciągu znaków przypisanego do konkretnej predefiniowanej komendy. Pseudo kod tej funkcji przedstawiono na rysunku 20.

```

bool ParseCommand(String command)
{
    command.trim(); // usunięcie białych znaków z przodu i końca ciągu znaków
    int separationNameIndex = command.indexOf(" "); // wyszukanie pierwszego indeksu spacji w ciągu
    String commandName = command.substring(0, separationNameIndex); // podzielenie ciągu do pierwszej spacji
    String commandArgs = command.substring(separationNameIndex + 1); // podzielenie ciągu od pierwszej spacji do końca ciągu

    for(Command cmd in commands) // iterowanie poprzez wszystkie komendy
    {
        if(cmd.commandName == commandName) // jeżeli nazwa iterowanej komendy jest taka sama jak komenda z konsoli
        {
            return cmd.Invoke(commandArgs); // to wykonaj komendę z argumentami z konsoli
        }
    }
    Serial.println("Didn't found command"); // wypisz w terminalu informację, że nie znaleziono komendy
    return false; // zwróć fałsz
}

```

Rysunek 20 Fragment pseudokodu funkcji analizującej komendę

## 5.2. Kinematyka prosta i odwrotna

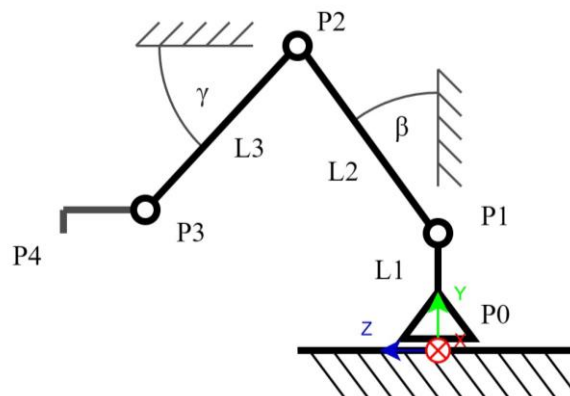
Sterowanie robotem odbywa się poprzez określenie pozycji, do której silniki mają obrócić ramiona robota. Jednak ten sposób jest mało praktyczny, ze względu na brak łatwej możliwości ustawienia dokładnej pozycji chwytaka lub innej końcówki robota. Przeprowadzane obliczenia kinematyki prostej i odwrotnej muszą dodatkowo spełniać

możliwość zmiany przesunięcia końca narzędzia, które dla każdego narzędzia może być inne. Obliczenia w tym rozdziale powstały w oparciu o [13].

Do obliczania kinematyki prostej posłużono się uproszczonym modelem kinematycznym robota pokazanym na rysunku 21. Kąty ramion L2 oraz L3, czyli  $\beta$  oraz  $\gamma$  odnoszą się do podłoża odpowiednio pionowego oraz poziomego. Kąt  $\alpha$  odnosi się do obrotu ramienia L1 wokół własnej pionowej osi. Przesunięcie punktu P3 do P4 jest indywidualną wartością ustawianą podczas pracy ramienia robota i oznacza ona przesunięcie punktu końcowego zamontowanego narzędzia. W równaniach, jeżeli chciano użyć wartości z konkretnej komórki wektora, to użyto oznaczenia  $\bar{v}(n)$ , gdzie  $n$  oznacza numer komórki wektora  $v$ .

W przypadku kinematyki prostej oraz odwrotnej konieczne jest odpowiednie obliczenie przemieszczenia zależnie od narzędzia zamontowanego do kiści robota. Wektor tego przemieszczenia w obliczeniach oznaczany jest za pomocą  $\bar{t}$ .

W obliczaniu kinematyki prostej pierwszym krokiem jest utworzenie wektora punktu P1, co opisano w równaniu (4) następnie liczone jest przemieszczenie z punktu P1 do P2 w równaniu (5) oraz położenie punktu P2 w równaniu (6). W następnym kroku obliczany jest wektor przemieszczenia z punktu P2 do P3 (równanie (7)) oraz położenie punktu P3 (równanie (8)). Kolejnym etapem jest obrót punktu wokół osi Y o kąt  $\alpha$ , obliczenia przedstawiono w równaniu (9). Równanie (10) przedstawia sposób obliczania przemieszczenia z punktu P3 do P4, które jest przesunięciem kątowym narzędzia wektora  $\bar{t}$  przy pomocy algorytmu kąta obrotu układu współrzędnych. Ostatnim krokiem jest obliczenie pozycji punktu P4, które pokazano w równaniu (11).



Rysunek 21 Model kinematyczny robota

$$\bar{P}_1 = \begin{bmatrix} 0 \\ L1 \\ 0 \end{bmatrix} \quad (4)$$

$$\bar{P}_{21} = \begin{bmatrix} 0 \\ \cos(\beta) \cdot L2 \\ 0 \end{bmatrix} \quad (5)$$

$$\bar{P}_2 = \bar{P}_1 + \bar{P}_{21} \quad (6)$$

$$\bar{P}_{32} = \begin{bmatrix} 0 \\ -\sin(\gamma) \cdot L3 \\ \cos(\gamma) \cdot L3 \end{bmatrix} \quad (7)$$

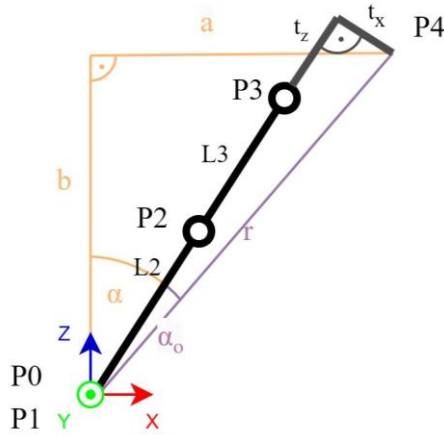
$$\bar{P}_3 = \bar{P}_2 + \bar{P}_{32} \quad (8)$$

$$\bar{P}_{3R} = \begin{bmatrix} \sin(\alpha) \cdot \bar{P}_3(1) \\ \bar{P}_3(2) \\ \cos(\alpha) \cdot \bar{P}_3(3) \end{bmatrix} \quad (9)$$

$$\bar{O} = \begin{bmatrix} \cos(\alpha) \cdot \bar{t}(1) + \sin(\alpha) \cdot \bar{t}(3) \\ \bar{t}(2) \\ \cos(\alpha) \cdot \bar{t}(3) + \sin(\alpha) \cdot \bar{t}(1) \end{bmatrix} \quad (10)$$

$$\bar{P}_4 = \bar{P}_{3R} + \bar{O} \quad (11)$$

Przy obliczaniu kinematyki odwrotnej możliwe jest podzielenie problemu na dwa mniejsze. Pierwszym jest rozpatrzenie układu rzutowanego na płaszczyźnie XY. Schemat tego rzutowania przedstawiono na rysunku 22. W pierwszym kroku liczona jest długość odcinka od punktu P1 do punktu P4 oznaczona literą  $r$ , a wzór pokazano w równaniu (12). Kolejnym krokiem jest obliczenie kompensacji przesunięcia końcówki narzędzia o wartość  $t_x$ , która jest pierwszą składową wektora  $\bar{t}$ , (13). Ostatnim krokiem w tym układzie jest obliczenie kąta  $\alpha$  (14).



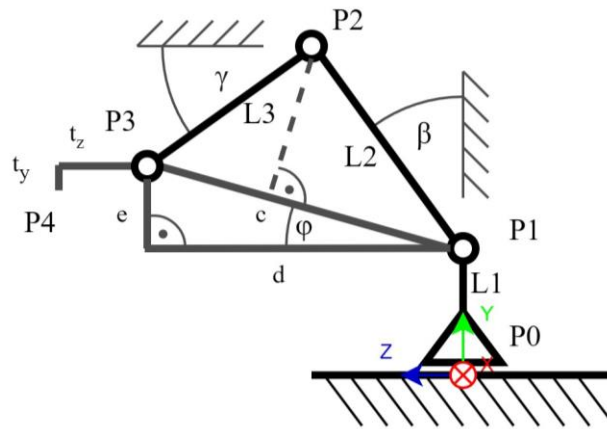
Rysunek 22 Schemat do obliczeń kąta obrotu pierwszej osi

$$r = \sqrt{\overline{P_{41}}(1)^2 + \overline{P_{41}}(3)^2} \quad (12)$$

$$\alpha_o = \text{asin}\left(\frac{t_x}{\overline{P_{41}}}\right) \quad (13)$$

$$\alpha = \text{atan}\left(\frac{a}{b}\right) - \alpha_o \quad (14)$$

Drugim krokiem jest rozpatrzenie układu w osi wyznaczonej przez ramie z układu pierwszego, taki układ można uprościć do płaszczyzny ZY, schemat przedstawiono na rysunku 23. Pierwszym krokiem w obliczaniu tego układu jest rzutowanie wektora przesunięcia narzędzia  $\bar{t}$  na lokalny układ (15). W następnym kroku obliczono wektor przemieszczenia z punktu P1 do P3 (16). Następnie obliczono d (17) oraz e (18), będące ramionami trójkąta prostokątnego, co pozwoliło na obliczenie kąta  $\varphi$  w równaniu (19). Później dokonano obliczenia długości prostej c (20), co pozwoliło na obliczenie kąta  $\beta$ . Ze względu na to, że długości ramion L2 oraz L3 są takie same, możliwe było zastosowanie uproszczonego wzoru do obliczenia kąta. Ostatnim krokiem jest obliczenie kąta  $\gamma$ , co pokazano w równaniu (22) [14].



Rysunek 23 Schemat do obliczeń kąta obrotu drugiej oraz trzeciej osi

$$\bar{O} = \begin{bmatrix} \sin(\alpha) \cdot t_z \\ t_y \\ \cos(\alpha) \cdot t_z \end{bmatrix} \quad (15)$$

$$\overline{P_{31}} = \overline{P_{41}} - \bar{O} \quad (16)$$

$$d = \sqrt{\overline{P_{31}}(1)^2 + \overline{P_{31}}(3)^2} \quad (17)$$

$$e = \overline{P_{31}}(2) \quad (18)$$

$$\varphi = \text{atan}\left(\frac{e}{d}\right) \quad (19)$$

$$c = \sqrt{d^2 + e^2} \quad (20)$$

$$\beta = 90^\circ - \text{acos}\left(\frac{c/2}{L_2}\right) - \varphi \quad (21)$$

$$\gamma = 180^\circ - \beta - \text{acos}\left(\frac{L_2^2 + L_3^2 - c^2}{2 \cdot L_2 \cdot L_3}\right) - \varphi \quad (22)$$

### 5.3. Implementacja interpolacji liniowej

Ze względu na konstrukcję robota składającą się z ramion poruszających się ruchem obrotowym interpolacja liniowa nie polega jedynie na obrocie silników krokowych z jednego punktu do drugiego. W takim wypadku końcówka ramienia będzie się poruszać po bliżej nieokreślonym łuku. Uproszczony schemat takiego ruchu pokazano na rysunku 24, gdzie pomarańczowa linia przedstawia tor ruchu z punktu A do B. Rozwiązaniem jest



podzielenie ruchu z jednego punktu do drugiego na dużo mniejszych ruchów pomiędzy punktami A i B, które dalej będą się odbywać po łuku, jednak poprzez zastosowanie odpowiednio małych przyrostów w ruchu, taki tor będzie nie rozróżnialny od liniowego.



Rysunek 24 Schemat ruchu przed zastosowaniem algorytmu interpolacji liniowej



Rysunek 25 Schemat ruchu po zastosowaniu interpolacji liniowej

Implementację tego algorytmu pokazano pseudokodem na rysunku 26. Aby ten algorytm działał prawidłowo konieczne jest wywoływanie tego algorytmu w głównej pętli programu tak często jak to tylko jest możliwe.

```

accumulatedDT += Time.deltaTime; // Dodaj do zakumulowanej zmiany czasu aktualną zmianę czasu
if(accumulatedDT < requiredDT) // Czy zakumulowana zmiana czasu jest mniejsza od wymaganej zmiany czasu
    goto EXIT_LINEAR; // To opóść interpolację liniową

Vector3 currentPos = CurrentPosition(); // Pobierz aktualną pozycję
double velocityMag = GetVelocityMagnitude(); // Pobierz wartość prędkości

if(IsCloseEnough(targetPos,currentPos, 0.5)) // Czy aktualne położenie jest wystarczająco blisko celu
{
    rot = InverseKinematics(targetPos); // Oblicz odwrotną kinematykę dla pozycji celu
}
else
{
    Vector3 currentToTarget = targetPos - currentPos; // Oblicz wektor do pozycji celu
    Vector3 velocityDir = currentToTarget.Normalized(); // Normalizuj wektor do celu
    Vector3 velocity = velocityDir * velocityMag; // Oblicz prędkość
    Vector3 deltaPos = velocity * accumulatedDT; // Oblicz zmianę położenia
    Vector3 nextPos = OverShootSafe(targetPos, currentPos, deltaPos); // Zabezpiecz przed przestrzałem pozycji
    rot = InverseKinematics(nextPos); // Oblicz odwrotną kinematykę dla następnej pozycji
}

Vector3 currentSpeed;
Vector3 currentAccel;
currentSpeed, currentAccel = SimpleSpeed(rot); // Oblicz przyspieszenie i prędkość dla obrotu
SetSteppersMaxSpeed(currentSpeed); // Ustaw obliczoną prędkość
SetSteppersMaxAcceleration(currentAccel); // Ustaw obliczone przyspieszenie

steppers.MoveTo(rot[i]); // Ustaw pozycję ruchu dla silników krokowych

accumDT -= requiredDT; // Odejmij wymaganą zmianę czasu

```

Rysunek 26 Algorytm interpolacji liniowej

## 5.4. Transformacja układu współrzędnych

Pomimo że możliwe jest wykonywanie ruchów w układzie współrzędnych robota, jest to jednak trudne i niepraktyczne. Dlatego konieczne było zaimplementowanie odpowiednich transformacji układu współrzędnych lokalnych, definiowanych przez operatora robota,

na globalny układ współrzędnych robota. Procedura definiowania lokalnego układu polega na podjechaniu końcówką robota w trzy specyficzne punkty. Pierwszym jest początek układu współrzędnych. Drugim jest zdefiniowanie pozycji kierunku osi Z, a następnie zdefiniowanie kierunku osi X. Posiadając te dane może zostać zbudowana macierz przekształcenia lokalnego układu współrzędnych na globalny układ współrzędnych robota.

Obliczenia rozpoczynane są od wyznaczenia kierunku osi Z. Obliczenia przedstawiono w równaniu (23), gdzie  $\bar{O}$  jest początkiem lokalnego układu a  $\bar{Z}$  jest pozycją kierunku osi Z. Analogicznie postępujemy z osią X (24). Kolejnym krokiem jest wyznaczenie wersora  $\bar{i}$ , gdzie obliczenia przedstawiono w równaniu (25). Analogicznie postępujemy przy wyznaczaniu wersora  $\bar{k}$  (26). Sposób obliczania wersora  $\bar{j}$  pokazano w równaniu (27). Posiadając trzy wersory i przesunięcie początkowe możliwe jest utworzenie macierzy transformacji z układu lokalnego do układu globalnego robota. Sposób ustawienia danych w macierzy przedstawia wzór (28). Posiadając obliczoną macierz transformacji  $\mathbb{T}$  możemy obliczyć współrzędne w układzie robota  $\bar{R}$  podając współrzędne w układzie lokalnym  $\bar{L}$ , co opisano równaniem (29).

$$\overline{Dir_Z} = \bar{Z} - \bar{O} \quad (23)$$

$$\overline{Dir_X} = \bar{X} - \bar{O} \quad (24)$$

$$\bar{i} = \overline{Dir_X} \cdot \frac{1}{\|\overline{Dir_X}\|} \quad (25)$$

$$\bar{k} = \overline{Dir_Z} \cdot \frac{1}{\|\overline{Dir_Z}\|} \quad (26)$$

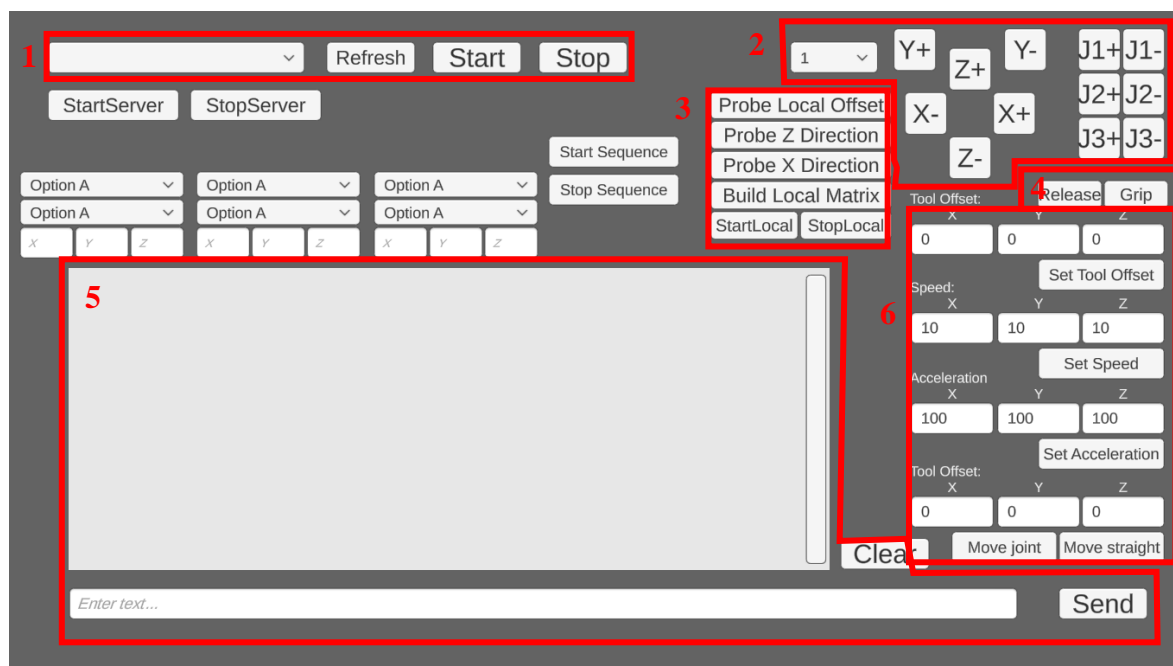
$$\bar{j} = \bar{i} \times \bar{k} \quad (27)$$

$$\mathbb{T} = \begin{bmatrix} \bar{i}(1) & \bar{j}(1) & \bar{k}(1) & \bar{O}(1) \\ \bar{i}(2) & \bar{j}(2) & \bar{k}(2) & \bar{O}(2) \\ \bar{i}(3) & \bar{j}(3) & \bar{k}(3) & \bar{O}(3) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (28)$$

$$\bar{R} = \mathbb{T} \times \bar{L} \quad (29)$$

## 5.5. Dedykowane oprogramowanie do sterowania robotem

Sterowanie oraz konfiguracja robota odbywają się poprzez wysyłanie komend przy użyciu portu szeregowego. Dlatego z założenia nie jest wymagane posiadanie dedykowanego oprogramowania do sterowania robotem. Jednak wpisywanie komend często wiąże się z błędami we wpisywaniu do terminala oraz ponownego ich wpisania. Dodatkowo część operacji wymaga częstego wpisywania komend. Dlatego w celu rozwiązania tej niedogodności wykonano dedykowane oprogramowanie na komputery osobiste. Miało ono pozwalać na przeprowadzanie operacji ruchu i prostych powtarzalnych czynności konfiguracyjnych przy pomocy interfejsu graficznego. Dodatkowo konieczne było posiadanie terminala do wpisywania komend rzadko używanych lub takich, które mają znaczący wpływ na pracę robota, i nie powinny być naciśnięte/wybrane przypadkiem. Oprogramowanie zostało wykonane przy użyciu silnika Unity z wykorzystaniem języka programowania C#. Na rysunku 27 pokazano widok okna dedykowanego oprogramowania do sterowania robotem. Numerem 1 oznaczono część interfejsu odpowiedzialną za wybór odpowiedniego portu szeregowego. Numerem 2 oznaczono część odpowiedzialną za poruszanie ramieniem w osiach X,Y,Z oraz w przegubach obrotowych. W obszarze 3 znajdują się przyciski odpowiedzialne za wyznaczanie oraz włączanie i wyłączanie układu lokalnego. Numer 4 przedstawia przyciski odpowiedzialne za otwarcie i zamknięcie chwytaka. Terminal został oznaczony obszarem z numerem 5. Ustawianie prędkości przyspieszeń czy wektora przesunięcia narzędzia oznaczono obszarem z numerem 6.

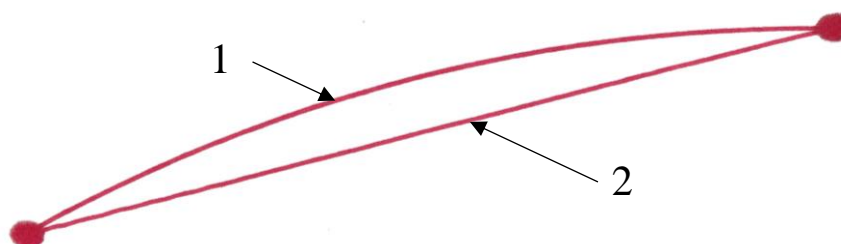


Rysunek 27 Widok okna dedykowanego oprogramowania do serowania robotem

## 6. Wyniki przeprowadzonych testów

Przetestowanie sprawności robota polegało na pokazaniu wpływu zastosowanego algorytmu interpolacji liniowej oraz przeprowadzeniu testów możliwości udźwigu robota.

Na rysunku 28 przedstawiono efekt zastosowania algorytmu interpolacji liniowej. Rysunek ten przedstawia skan kartki, po której rysowało ramię robota przy użyciu zamontowanego do jego kiści pisaka. Numerem 1 oznaczono tor ruchu bez interpolacji liniowej, numerem 2 oznaczono tor ruchu po zastosowaniu interpolacji liniowej. Tor ruchu oznaczony numerem 2 jest zbliżony do linii.



Rysunek 28 Test interpolacji liniowej

Kolejnym przeprowadzonym testem był sprawdzenie możliwości udźwigu robota. Test polegał na zamontowaniu do kiści robota końcówki ze znanymi wymiarami i przesunięcie tej końcówki tak, aby znajdowała się ona 400 mm od osi robota. Następnie na wyzerowaną wagę zaczęto powoli napierać ramieniem robota do momentu zgubienia kroku. Aby możliwe było odczytanie wyniku z wagi cały proces był nagrywany. Podczas testów maksymalna zmierzona siła wynosiła 2175 g. Należy jednak pamiętać, że jest to tylko siła statyczna. Dodatkowym testem było zamontowanie ciężaru o masie 1 kg i poruszanie kiścią robota w górę i w dół, na różnych odległościach od podstawy. Prędkość tego ruchu wynosiła 30 mm/s. Odległość, na której ramię robota zaczęło gubić kroki wynosiło 460 mm, gdzie testy przeprowadzano przy przesunięciach co 10 mm.

## 7. Podsumowanie

W pracy dokonano wyboru odpowiedniej konstrukcji ramienia robota. Zaprojektowano jego konstrukcję z myślą o wytwarzaniu przyrostowym oraz zaprojektowano odpowiedni układ elektroniczny. Wykonano odpowiednie oprogramowanie na mikrokontroler, który steruje ruchem ramienia robota. Oprogramowanie to pozwala na obliczanie kinematyki odwrotnej ruchu robota, ruch liniowy oraz transformację układu współrzędnych. Dodatkowo w celu łatwej komunikacji z robotem wykonane zostało dedykowane oprogramowanie.

Możliwości robota mogą zostać poszerzone o dodanie enkoderów kontrolujących pracę silników krokowych. Możliwe jest też dodanie dodatkowej czwartej oraz ewentualnej piątej osi na kiści robota, co pozwoliłoby na przeprowadzanie bardziej skomplikowanych operacji. Od strony oprogramowania robota możliwe jest dodanie możliwości poruszania się kiści po łukach i krzywych. W celu poprawy płynności ruchu można też dodać możliwość ustawienia zrywu w ruchu kiści robota.

Ramię robota spełnia założenia projektu o przenoszeniu minimalnej masy oraz zakresie ruchu ramienia. Potrafi również sprawnie wykonywać przekazywane mu instrukcje wysyłane portem szeregowym.

Wykonany projekt pozwolił wykorzystać wiedzę z projektowania konstrukcji maszyn, programowania mikrokontrolerów, programowania aplikacji oraz projektowania układów elektronicznych.

## Bibliografia

- [1] J. Kosmol, Napędy Mechatroniczne, Gliwice: Wydawnictwo Politechniki Śląskiej, 2013.
- [2] „Silnik krokowy JK42HS48-1204 200 kroków/obr 3,6V/1,2A/0,47Nm,” BOTLAND, [Online]. Available: [www.botland.com.pl/silniki-krokowe/3613-silnik-krokowy-jk42hs48-1204-200-krokowobr-36v12a047nm-5904422332020.html](http://www.botland.com.pl/silniki-krokowe/3613-silnik-krokowy-jk42hs48-1204-200-krokowobr-36v12a047nm-5904422332020.html). [Data uzyskania dostępu: 26 12 2023].
- [3] „Sterownik silnika krokowego DRV8825 StepStick,” BOTLAND, [Online]. Available: [www.botland.com.pl/sterowniki-silnikow-krokowych/3739-sterownik-silnika-krokowego-drv8825-stepstick-5904422332174.html](http://www.botland.com.pl/sterowniki-silnikow-krokowych/3739-sterownik-silnika-krokowego-drv8825-stepstick-5904422332174.html). [Data uzyskania dostępu: 26 12 2023].
- [4] „Serwo MG-90S - micro - 180 stopni - metalowa przekładnia,” BOTLAND, [Online]. Available: [www.botland.com.pl/serwa-typu-micro/20435-serwo-mg-90s-micro-180-stopni-metalowa-przekladnia-5904422380915.html](http://www.botland.com.pl/serwa-typu-micro/20435-serwo-mg-90s-micro-180-stopni-metalowa-przekladnia-5904422380915.html). [Data uzyskania dostępu: 26 12 2023].
- [5] „ESP32-S2-Saola-1R + ESP32-S2-WROVER - zestaw rozwojowy,” BOTLAND, [Online]. Available: [www.botland.com.pl/moduly-wifi-i-bt-esp32/18400-esp32-s2-saola-1r-esp32-s2-wrover-zestaw-rozwojowy-5904422369446.html](http://www.botland.com.pl/moduly-wifi-i-bt-esp32/18400-esp32-s2-saola-1r-esp32-s2-wrover-zestaw-rozwojowy-5904422369446.html). [Data uzyskania dostępu: 26 12 2023].
- [6] D. Zieliński, „Technologia FDM/FFF,” BIBUS MENOS Sp. z o.o., [Online]. Available: [www.drukarki3d.pl/technologie/technologie-fdm-fff/](http://www.drukarki3d.pl/technologie/technologie-fdm-fff/). [Data uzyskania dostępu: 02 01 2024].
- [7] „PET-G Premium,” Spectrum Group Sp. z o.o., [Online]. Available: [www.spectrumfilaments.com/filament/pet-g-premium/](http://www.spectrumfilaments.com/filament/pet-g-premium/). [Data uzyskania dostępu: 02 01 2024].
- [8] „PLA Premium,” Spectrum Group Sp. z o.o., [Online]. Available: [www.spectrumfilaments.com/filament/pla-premium/](http://www.spectrumfilaments.com/filament/pla-premium/). [Data uzyskania dostępu: 01 02 2024].
- [9] D. Conradie, „3D PRINTING BORES WITHOUT SUPPORT,” 02 01 2024. [Online]. Available: <https://hackaday.com/2023/05/28/3d-printing-bores-without-support/>.

- [10] „How 3D Printing direction/orientation affects strength?,” FacFox Docs, [Online]. Available: [www.facfox.com/docs/kb/how-3d-printing-direction-orientation-affects-strength](http://www.facfox.com/docs/kb/how-3d-printing-direction-orientation-affects-strength). [Data uzyskania dostępu: 02 01 2024].
- [11] „Embedding Nuts in 3D Printed Parts for Hidden Fastener Strength,” Markforged, Inc., [Online]. Available: [www.markforged.com/resources/blog/embedding-nuts-3d-printing](http://www.markforged.com/resources/blog/embedding-nuts-3d-printing). [Data uzyskania dostępu: 03 01 2024].
- [12] „DRV8825 Stepper Motor Driver Carrier, High Current,” Pololu Corporation, [Online]. Available: <https://www.pololu.com/product/2133/resources>. [Data uzyskania dostępu: 2024 01 18].
- [13] J. J. Craig, Wprowadzenie do robotyki, mechanika i sterowanie, Warszawa: Wydawnictwa Naukowo-Techniczne, 1995.
- [14] „Solving SSS Triangles,” Rod Pierce, [Online]. Available: <https://www.mathsisfun.com/algebra/trig-solving-sss-triangles.html>. [Data uzyskania dostępu: 22 01 2024].



# Spis rysunków

Rysunek 1 Model kinematyczny robota .....	9
Rysunek 2 Silnik krokowy JK42HS48-1204 [2] .....	9
Rysunek 3 Sterownik silnika krokowego DRV8825 [3] .....	10
Rysunek 4 Serwo MG-90S [4] .....	10
Rysunek 5 Mikrokontroler ESP32-S2-WROWER [5] .....	10
Rysunek 6 Geometria otworu drukowanego od spodu .....	12
Rysunek 7 Geometria wydrukowanego otworu od spodu .....	12
Rysunek 8 Ułożenie ramienia głównego robota w programie Cura .....	13
Rysunek 9 Przekrój kiści robota .....	13
Rysunek 10 Zatrzymany wydruk kiści robota z osadzonymi nakrętkami .....	13
Rysunek 11 Uproszczony model ramienia z obciążeniem .....	14
Rysunek 12 Widok robota: a) z przedniego lewego boku b) z tylnego prawego boku .....	15
Rysunek 13 Mechanizm chwytaka .....	16
Rysunek 14 Schemat elektryczny układu sterowania .....	17
Rysunek 15 Projekt płytki elektrycznej .....	18
Rysunek 16 Wykonane oraz złożone ramie robota .....	19
Rysunek 17 Wykonany układ sterowania robota .....	20
Rysunek 18 Wykonany oraz złożony chwytak robota .....	20
Rysunek 19 Fragment pseudokodu nieblokującej funkcji zbierającej znaki z portu szeregowego .....	22
Rysunek 20 Fragment pseudokodu funkcji analizującej komendę .....	22
Rysunek 21 Model kinematyczny robota .....	23
Rysunek 22 Schemat do obliczeń kąta obrotu pierwszej osi .....	25
Rysunek 23 Schemat do obliczeń kąta obrotu drugiej oraz trzeciej osi .....	26
Rysunek 24 Schemat ruchu przed zastosowaniem algorytmu interpolacji liniowej .....	27
Rysunek 25 Schemat ruchu po zastosowaniu interpolacji liniowej .....	27
Rysunek 26 Algorytm interpolacji liniowej .....	27
Rysunek 27 Widok okna dedykowanego oprogramowania do serowania robotem .....	30
Rysunek 28 Test interpolacji liniowej .....	31