

Politechnika Warszawska

W Y D Z I A Ł S A M O C H O D Ó W
I M A S Z Y N R O B O C Z Y C H



Praca dyplomowa inżynierska

na kierunku Mechatronika Pojazdów i Maszyn Roboczych
w specjalności Mechatronika Maszyn Roboczych

Projekt i wykonanie manipulatora przegubowego z wykorzystaniem
koncepcji Internetu rzeczy

numer pracy według wydziałowej ewidencji prac: 115B-ISP-MR/282628/1191236

Wojciech Wysokiński

numer albumu 282628

opiekun naukowy
dr hab. inż. Piotr Przybyłowicz, prof. PW

promotor
dr inż. Sebastian Korczak

konsultacje

—

WARSZAWA 2019

Streszczenie

W niniejszej pracy przedstawiono projekt oraz badania manipulatora przegubowego, w którym wykorzystano koncepcję Internetu rzeczy. Idea ta nawiązywała do prężnie rozwijającego się Przemysłu 4.0, którego założeniem była integracja ludzi oraz maszyn z Internetem i technologiami informatycznymi. W projekcie została wykorzystana płytki NodeMCU V2 z modułem ESP 8266, którego zadaniem była wymiana danych z chmurą oraz sterowanie silnikami. Oddziaływanie operatora na urządzenie odbywało się za pomocą interfejsu graficznego w chmurze danych. Wartości wprowadzone tam były konwertowane i przesyłane do modułu ESP 8266, a następnie przeliczone na pozycje kątowe 5 silników. Sterowanie nimi było możliwe za pomocą modulacji PWM. Cała konstrukcja nośna została zaprojektowana w programie 3D CAD SolidWorks wraz z obliczeniami wytrzymałościowymi wykorzystując metodę elementów skończonych. Wszystkie elementy mechaniczne zostały wykonane metodą FDM na drukarce 3d.

Praca dyplomowa została podzielona na trzy części. W pierwszej części pracy przedstawiono zagadnienie związane z manipulatorami oraz ich podziałem. Ponadto scharakteryzowano pojęcie Internetu rzeczy oraz praktyczne zastosowania w przemyśle. W następnej części szczegółowo została opisana konstrukcja nośna manipulatora oraz symulacje wytrzymałościowe z wykorzystaniem metody elementów skończonych. Dodatkowo przedstawiono szkic napisany w Arduino IDE, który był odpowiedzialny za komunikację z chmurą danych oraz za sterowanie silnikami. W ostatniej części pracy przetestowane zostało całe urządzenie pod kątem funkcjonalności oraz wrażliwości na błędy podczas komunikacji z chmurą. Dzięki temu było możliwe wyciągnięcie odpowiednich wniosków podsumowujących efekt pracy autora.

Słowa kluczowe:

Manipulator przegubowy, Internet rzeczy, chmura danych, NodeMCU

Design and implementation of an articulated manipulator using the concept of Internet of Things

Abstract

In this work a design and research on an articulated manipulator that uses the concept of the Internet of Things to control were presented. This idea referred to the dynamically developing Industry 4.0, which assumption was the integration of people and machines with the Internet and information technologies. This project used a NodeMCU V2 board with the ESP 8266 module, which task was to exchange data with the cloud and control electrical motors. The operator's influence on the device took place via a graphical interface in the data cloud. The values entered there were converted and sent to the ESP 8266 module, and then converted to the angular position of 5 motors. These servomotors were controlled with PWM signal. The entire supporting structure was designed in 3D CAD SolidWorks along with strength calculations using the finite element method. All mechanical components were made by the FDM method on a 3d printer.

The thesis has been divided into three parts. In the first part general informations and classification of manipulators were presented. Additionally the concept of the Internet of Things and it's practical applications in industry were described. In the next part, the support structure of the manipulator was described in detail, as well as strength simulations using the finite element method. In addition, a sketch written in the Arduino IDE was presented, which was responsible for communication with the cloud of data and for controlling motors. In the last part of the work, the entire device was tested for functionality and sensitivity to errors when communicating with the cloud. Thanks to this, it was possible to draw appropriate conclusions summarizing the effect of author's work.

Keywords:

Articulated manipulator, Internet of Things, data cloud, NodeMCU

Wojciech Wysokiński

/imię i nazwisko/

OŚWIADCZENIE

Jako autor pracy dyplomowej pt.: „Projekt i wykonanie manipulatora przegubowego z wykorzystaniem koncepcji Internetu rzeczy”, którą wykonałem samodzielnie przestrzegając zasad ochrony własności intelektualnej zezwalam na publiczne udostępnienie pracy i wyrażam zgodę na jej udostępnienie w Bibliotece Wydziału Samochodów i Maszyn Roboczych Politechniki Warszawskiej w ramach realizacji zadań statutowych biblioteki.

Warszawa, dnia 09.12.2019r.

.....

/podpis/

.....

Słowa kluczowe:

Manipulator przegubowy, Internet rzeczy, chmura danych, NodeMCU

Spis treści

1 Wprowadzenie	13
1.1 Przedmowa.....	13
1.2 Cel i zakres pracy	13
2 Manipulatory	15
2.1 Rys historyczny.....	15
2.2 Budowa i podział manipulatorów	15
3 Internet rzeczy	21
3.1 Pojęcie Internetu rzeczy	21
3.2 Technologie bezprzewodowe	21
3.3 Technologia WiFi	24
4 Koncepcja projektu manipulatora	27
4.1 Moduł Wi-Fi NodeMCU V2	27
4.2 Serwomechanizm SG-92R i MG996R	28
4.3 Chmura danych Thinger.io	32
4.4 Konstrukcja mechaniczna.....	33
4.4.1 Założenia	33
4.4.2 Realizacja.....	34
4.5 Kinematyka prosta i odwrotna manipulatora	49
4.5.1 Kinematyka prosta	49
4.5.2 Kinematyka odwrotna manipulatora	55
4.6 Sterowanie	57
4.7 Gotowe urządzenie	60
5 Badania funkcjonalne manipulatora	61
6 Zakończenie	65
6.1 Wnioski.....	65
6.2 Posumowanie pracy	65
7 Bibliografia.....	67
8 Spis symboli.....	69
9 Spis ilustracji.....	71
10 Spis tabel.....	73

1 Wprowadzenie

1.1 Przedmowa

Tematem pracy zainteresowałem się poprzez zajęcia poświęcone problemom automatyzacji maszyn roboczych. Podczas nich zapoznałem się z rozwiązaniami firmy Komatsu, które wykorzystywały łączność internetową do sterowania ruchem osprzętu roboczego oraz jazdy. Szukając informacji w sieci, natknąłem się na istniejące już chmury danych, które można zaimplementować do własnych zastosowań. Skłoniło to mnie do opracowania własnego urządzenia wykorzystującego Internet do sterowania.

1.2 Cel i zakres pracy

Celem pracy było opracowanie konstrukcji mechanicznej i elektrycznej oraz oprogramowanie manipulatora przegubowego, który wykorzystywał koncepcję Internetu rzeczy do sterowania osprzętem roboczym. Założeniem pracy było zbudowanie urządzenia wykorzystując płytkę NodeMCU v2 z modułem ESP 8266 do bezprzewodowej komunikacji z chmurą danych.

Zakres pracy obejmował projekt i budowę manipulatora przegubowego, którym można sterować poprzez łączność internetową. Została przedstawiona konstrukcja mechaniczna oraz wyniki analizy wytrzymałościowej, wykorzystującą metodę elementów skończonych. W dalszej części pracy został zaprezentowany szkic programu odpowiedzialny za łączność z chmurą danych oraz za sterowanie serwomechanizmami. Zakres pracy obejmował także testy funkcjonalne manipulatora wraz z zachowaniem się na przerwanie komunikacji.

2 Manipulatory

2.1 Rys historyczny

Manipulator jest to urządzenie, którego przeznaczeniem jest realizacja niektórych funkcji górnej kończyny człowieka. Powszechnie przyjmuje się, że rozwój techniki robotyzacyjnej zaczął się około 1954 roku. Wtedy to rozpoczęto prace naukowo-badawcze nad konstrukcjami maszyn manipulacyjnych dla poszukiwań oceanicznych oraz przemysłu nuklearnego. W tym roku George Devol konstruuje pierwszego programowalnego robota do zastosowań przemysłowych [2]. Siedem lat później firma Unimation zbudowała pierwszy manipulator przemysłowy, który zainstalowano później w Trenton na linii produkcyjnej General Motors [2].

Obecnie rynek robotów przemysłowych jest bardzo prężnie rozwijającą się dziedziną przemysłu. Według danych z raportu World Robotics Report 2018, w 2017r. liczba sprzedanych robotów przemysłowych wyniosła 381335 sztuk, zaś ogólna ich liczba przekroczyła liczbę 2 mln [7]. Od kilku lat tendencja ta jest wzrostowa. Największymi producentami tych urządzeń są: Chiny, Japonia, Korea Płd., Stany Zjednoczone oraz Niemcy. Najbardziej znane firmy produkujące manipulatory to: ABB, Fanuc, Kuka, Omron, Mitsubishi. Obecnie roboty przemysłowe są najczęściej wykorzystywane w branży automotive, gdzie trafia 1/3 światowej produkcji urządzeń, ponadto są obecne w przemyśle elektronicznym oraz metalowym.

2.2 Budowa i podział manipulatorów

W skład schematu funkcjonalnego robota wchodzi:

1. Podstawa – nieruchoma konstrukcja lub płyta, będąca pierwszym członem.
2. Korpus – obudowa elementów zespołu ruchów ramienia.
3. Ramię dolne i górne.
4. Przegub.
5. Element roboczy.
6. Sterowanie.
7. Napędy.

Manipulatory można podzielić na podstawie ich budowy, rodzaju sterowania, przeznaczenia.

Podział manipulatorów ze względu na budowę jednostki kinematycznej:

1. Jednostki monolityczne - jednostki kinematyczne, które posiadają niezmienną konstrukcję, która użytkownik może dodać z grupy komponentów dostarczonych przez producenta. Z tak przyjętej koncepcji wynika główna

wada, którą jest brak elastyczności często wymaga w dzisiejszych systemach produkcyjnych.

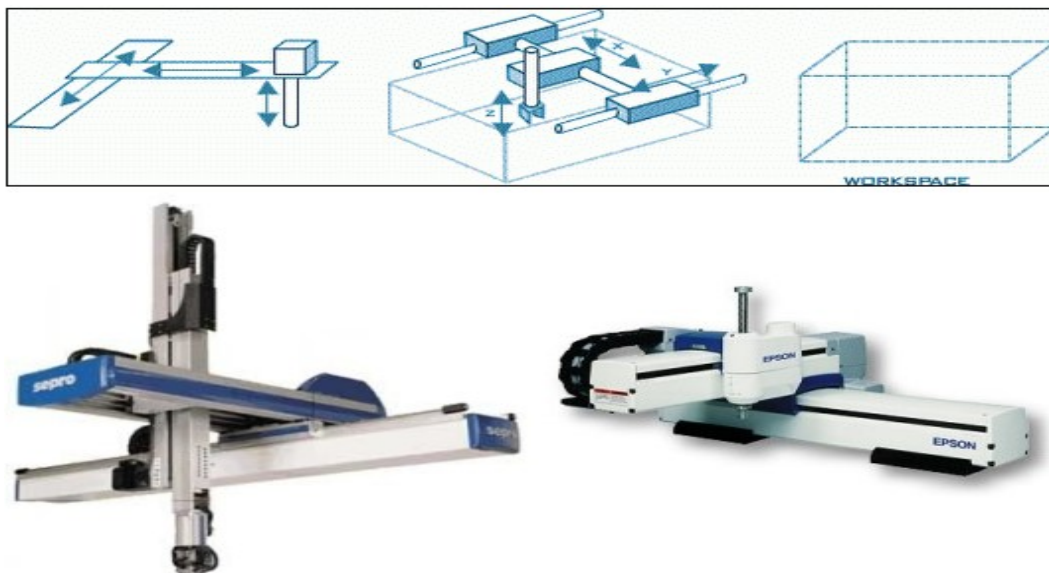
2. Jednostki modułowe – jednostki kinematyczne złożone z dostarczonych przez producenta zespołami ruchu. Zazwyczaj nie ma ograniczenia co do konfiguracji zestawienia modułów, jednak są ograniczone przez właściwości mechaniczne i dynamiczne.
3. Jednostki pseudomodułowe – jednostki kinematyczne o stałej strukturze kinematycznej. Charakteryzują się one możliwością wymiany niektórych elementów robota. Zazwyczaj są to ostatnie w łańcuchu kinematycznym zespoły ruchu.

Podział manipulatorów ze względu na strukturę kinematyczną:

1. Roboty stacjonarne

1.1 Roboty z szeregową strukturą kinematyczną:

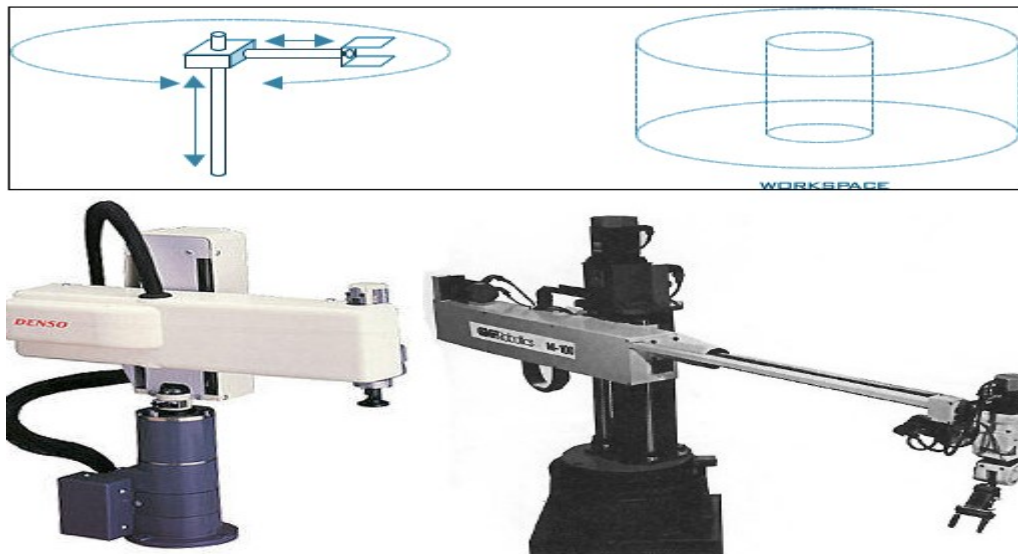
- 1.1.1 Kartezjańskie - roboty o takiej strukturze posiadają prostokątny układ współrzędnych i przestrzeń ruchu o kształcie prostopadłościanu. Taka budowa manipulatora jest stosowana do transportu materiału, ponadto łatwo montowalna na blacie stołu. (Rysunek 1)



Rysunek 1. Roboty kartezjańskie.

Źródło: <http://www.asimo.pl/teoria/roboty-przemyslowe.php> [09.12.2019]

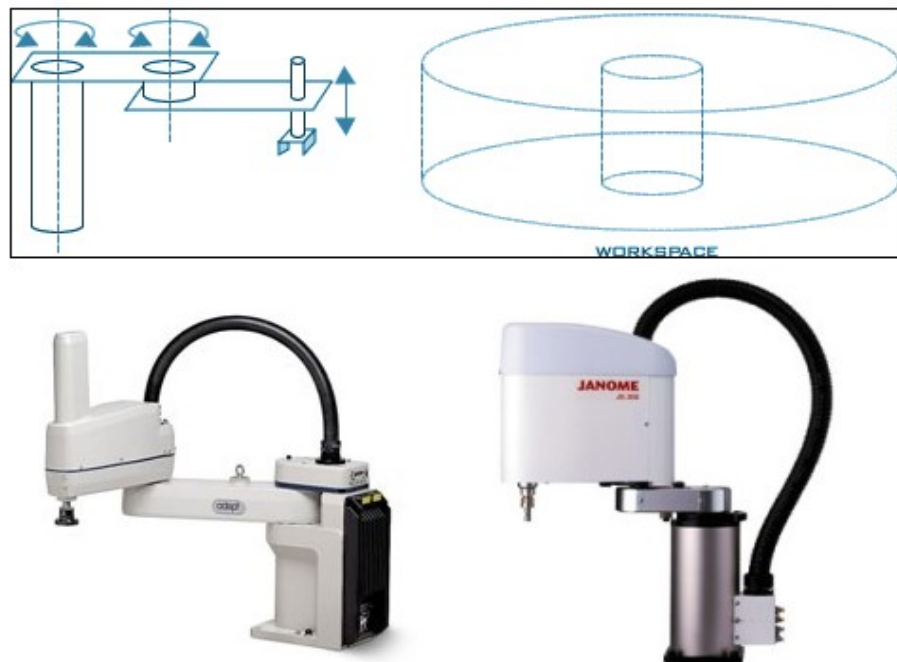
- 1.1.2 Cylindryczne – roboty posiadające dwa liniowe zespoły ruchu oraz jeden obrotowy. Cechą charakterystyczną jest walcowy układ współrzędnych oraz cylindryczna przestrzeń robocza. (Rysunek 2)



Rysunek 2. Roboty cylindryczne.

Źródło: <http://www.asimo.pl/teoria/roboty-przemyslowe.php> [09.12.2019]

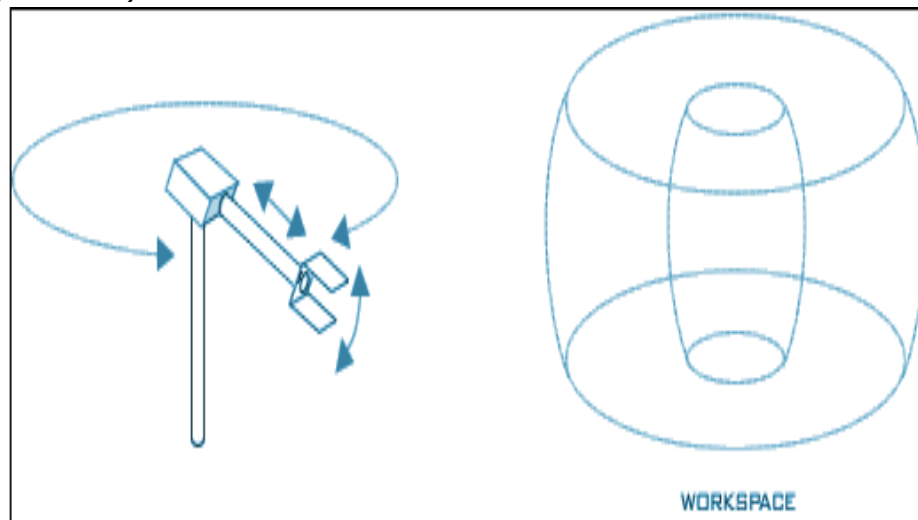
- 1.1.3 SCARA – roboty charakteryzujące się posiadaniem trzech osi równoległych – dwoma o ruchu obrotowym oraz jedną o ruchu postępowym. Głównym zastosowaniem manipulatorów tej klasy jest montaż elementów i podzespołów oraz podnoszenie detali i ich sortowanie. (Rysunek 3)



Rysunek 3. Roboty SCARA.

Źródło: <http://www.asimo.pl/teoria/roboty-przemyslowe.php> [09.12.2019]

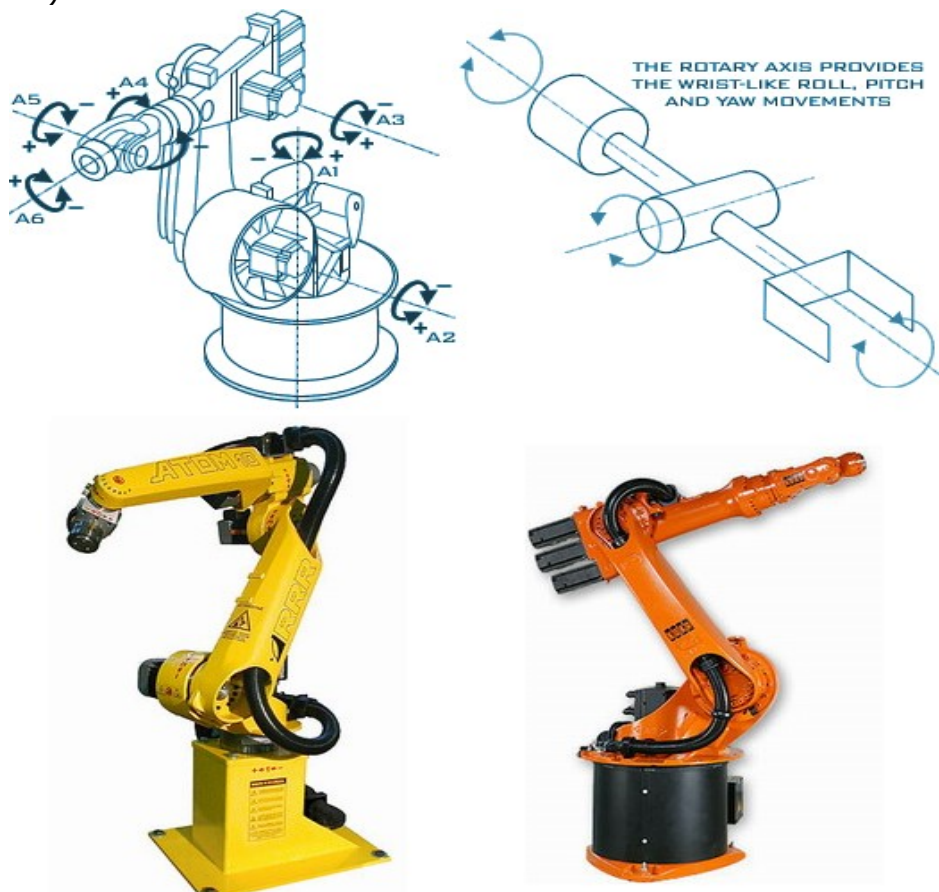
1.1.4 Sferyczne – roboty posiadające dwa obrotowe oraz jeden liniowy zespół ruchu. (Rysunek 4)



Rysunek 4. Roboty cylindryczne.

Źródło: <http://www.asimo.pl/teoria/roboty-przemyslowe.php> [09.12.2019]

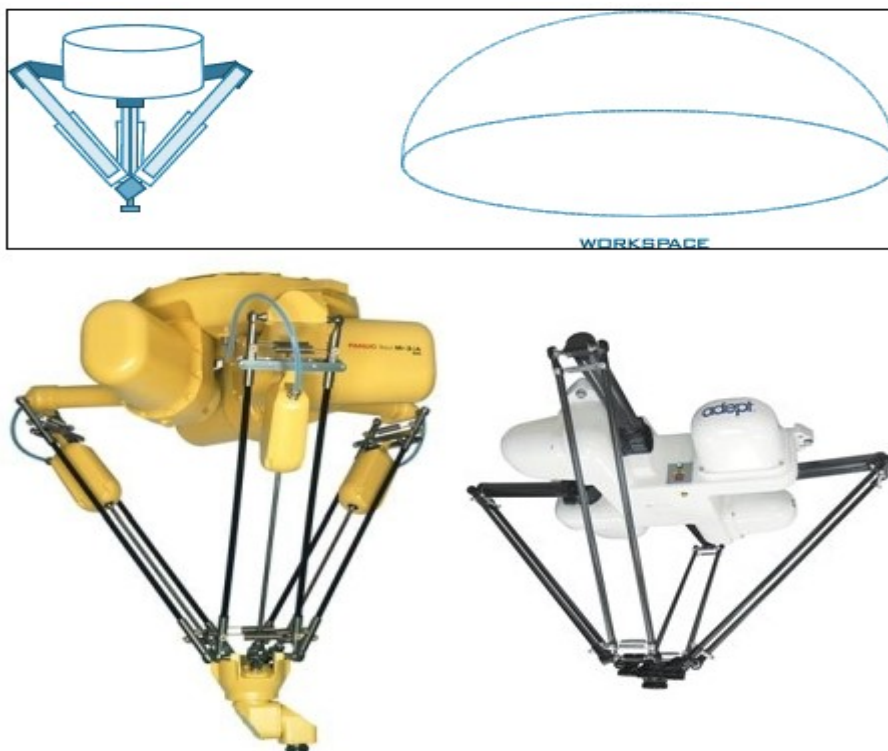
1.1.5 Przegubowe – roboty posiadające 3 osie obrotowe. Posiadają wiele zastosowań np. przy operacjach polerowania, montażu lub lakierowania. (Rysunek 5)



Rysunek 5. Roboty przegubowe.

Źródło: <http://www.asimo.pl/teoria/roboty-przemyslowe.php> [09.12.2019]

- 1.2 Roboty z równoległą strukturą kinematyczną – manipulator, którego ramiona tworzą zamknięty łańcuch kinematyczny. Często to są konstrukcje trój- lub sześciocłonowe. Dzięki takiemu rozwiązaniu robot posiada większą sztywność przy jednoczesnym zmniejszeniu masy. Charakteryzują się lepszą dokładnością pozycjonowania niż manipulatory o innych strukturach kinematycznych. (Rysunek 6)



Rysunek 6. Roboty z równoległą strukturą kinematyczną.

Źródło: <http://www.asimo.pl/teoria/roboty-przemyslowe.php> [09.12.2019]

2. Roboty mobilne – w grupie tej wyróżnia się manipulatory poruszające się po stałym torze oraz autonomiczne.

Podział manipulatorów ze względu na sterowanie:

1. Roboty sekwencyjne – roboty posiadające sekwencyjny system sterowania.
2. Roboty realizujące zadane trajektorie.
3. Roboty adaptacyjne – roboty posiadające zaawansowany system sterowania, który pozwala na dostosowanie ruchów do aktualnej sytuacji. Przykładem może być robot wyposażony w system wizyjny, dzięki któremu możliwa jest korekta trajektorii ruchu.
4. Teleoperator – robot ze sterowaniem zdalnym, często realizowanym przez komputer lub operatora.

Podział manipulatorów ze względu na rodzaj napędu:

1. Pneumatyczne
2. Elektryczne
3. Hydrauliczne

Klasyfikacja robotów ze względu na rodzaj wykonywanej pracy:

1. spawalnicze
2. montażowe
3. paletyzujące, pakujące i depaletyzujące
4. malarskie
5. do obróbki materiałów i manipulowania obiektami
6. transportujące i do załadunku palet (AGV, LGV)
7. roboty do owijania palet (owijarki)
8. roboty do utylizacji i zabezpieczania odpadów przemysłowych
9. napełniające i dozujące
10. inspekcyjne i testujące (Rysunek 7)



Rysunek 7. Przykłady robotów przemysłowych.

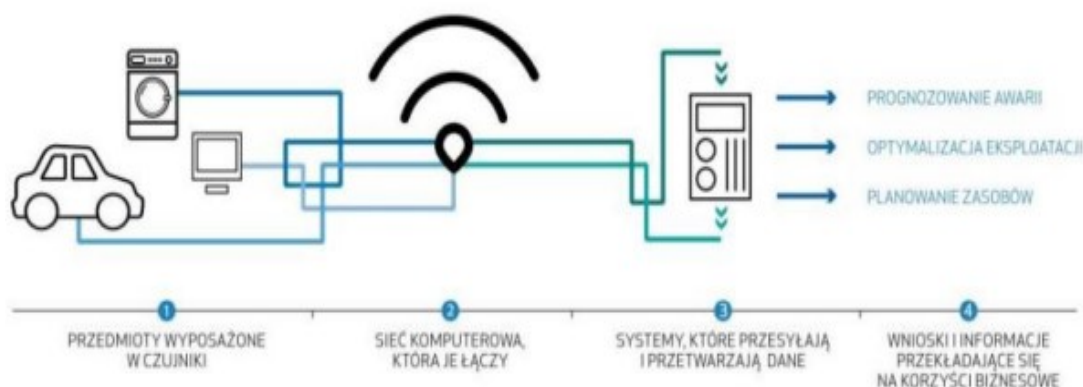
Źródło: <http://www.asimo.pl/teoria/roboty-przemyslowe.php> [09.12.2019]

3 Internet rzeczy

3.1 Pojęcie Internetu rzeczy

Pod pojęciem Internetu rzeczy należy rozumieć koncepcję, według której istnieje zbiór urządzeń włączonych do globalnej sieci. Dzięki temu poszczególne elementy systemu gromadzą, przetwarzają oraz wysyłają dane do innych urządzeń za pomocą dostępnej infrastruktury sieciowej. Można wyróżnić cztery elementy niezbędne do prawidłowego funkcjonowania Internetu rzeczy:

1. Urządzenia, których zadaniem jest gromadzenie oraz przesyłanie danych.
2. Sieci komunikacyjne łączące poszczególne urządzenia.
3. System teleinformatyczny, których zadaniem jest gromadzenie napływających danych.
4. Algorytmy analizujące dane. (Rysunek 8)



Rysunek 8. Koncepcja Internetu rzeczy.

Źródło: www.mechatronika-simr.home.pl [09.12.2019]

Termin ten został użyty po raz pierwszy w 1999 przez brytyjskiego przedsiębiorcę Kevina Ashtona, chcąc określić sieć połączonych ze sobą przedmiotów. Systemy wykorzystujące koncepcję Internetu rzeczy charakteryzują się nieograniczonym zasięgiem działania, dzięki temu znalazły zastosowania m. in. w inteligentnych budynkach oraz miastach, inteligentnych sieciach energetycznych, przemyśle. Idea ta jest ściśle związana z prężnie rozwijającym się w obecnych czasach Przemysłem 4.0, którego głównym założeniem jest integracja ludzi oraz maszyn sterowanych cyfrowo z Internetem oraz technologiami informatycznymi.

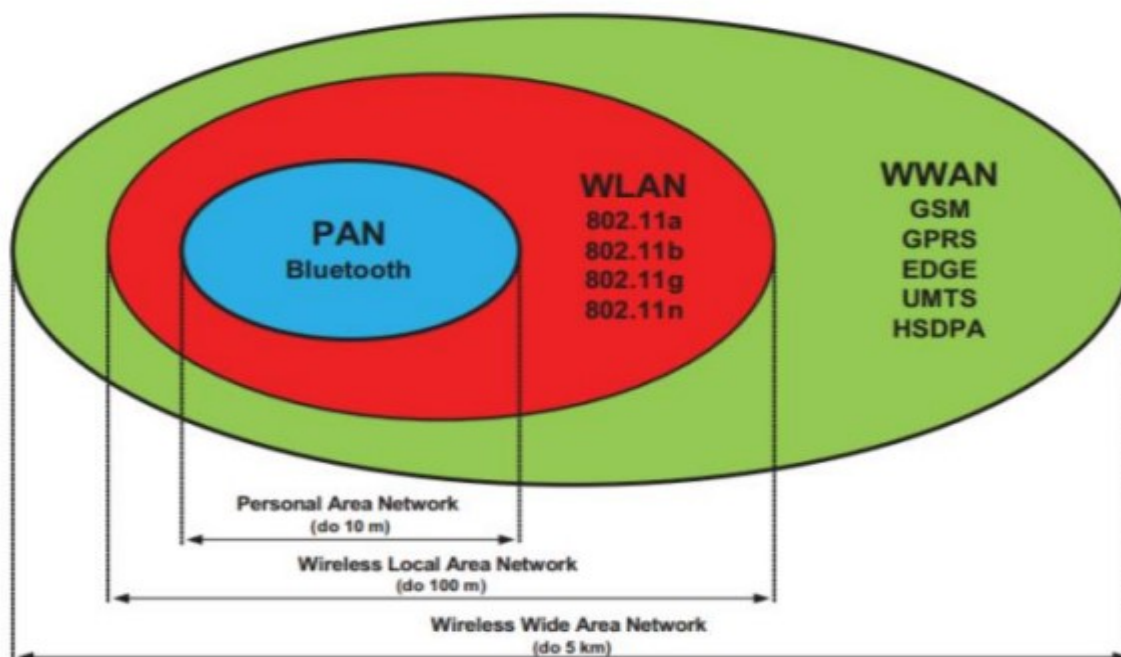
3.2 Technologie bezprzewodowe

Sieci bezprzewodowe są alternatywą dla tradycyjnych połączeń kablowych. Zaletą takich technologii jest brak konieczności stosowania przewodów, mobilność oraz szeroka dostępność dla urządzeń takich jak: komputery, smartfony, tablety.

Ponadto dostęp do sieci i ich zasobów jest stosunkowo prosty. Jednak poważnych ich ograniczeniem jest dość krótki zasięg, który w zależności od użytej technologii może wynosić od kilku metrów do kilku kilometrów. Ponadto sieci takie charakteryzują się niską przepustowością oraz stosunkowo dużym rozpraszaniem energii i wysokim poziomem zakłóceń pochodzących z zewnątrz.

Ze względu na zasięg działania można wyróżnić następujące typy sieci bezprzewodowej:

1. WPAN (Bezprzewodowa Sieć Osobista)- jest to sieć o najmniejszym zasięgu działania. Urządzenia działające w tej sieci mogą pracować w maksymalnej odległości kilkunastu metrów od siebie. Obecnie najważniejsze technologie sieci WPAN to Bluetooth oraz technologia podczerwieni.
2. WLAN (Bezprzewodowa Sieć Lokalna)- jest to sieć, której zasięg wynosi do 100 metrów. Może w niej pracować od kilku do kilkunastu urządzeń. Projektowana jest z wykorzystaniem standardu IEEE 802.11, która wykorzystuje fale radiowe o częstotliwości 2,4 i 5 GHz.
3. WWAN (Bezprzewodowe Sieci Rozległe)- jest to sieć o największym zasięgu działania, w której może występować wiele urządzeń. Przykładem technologii, o którą jest oparte WWAN jest GSM. (Rysunek 9)

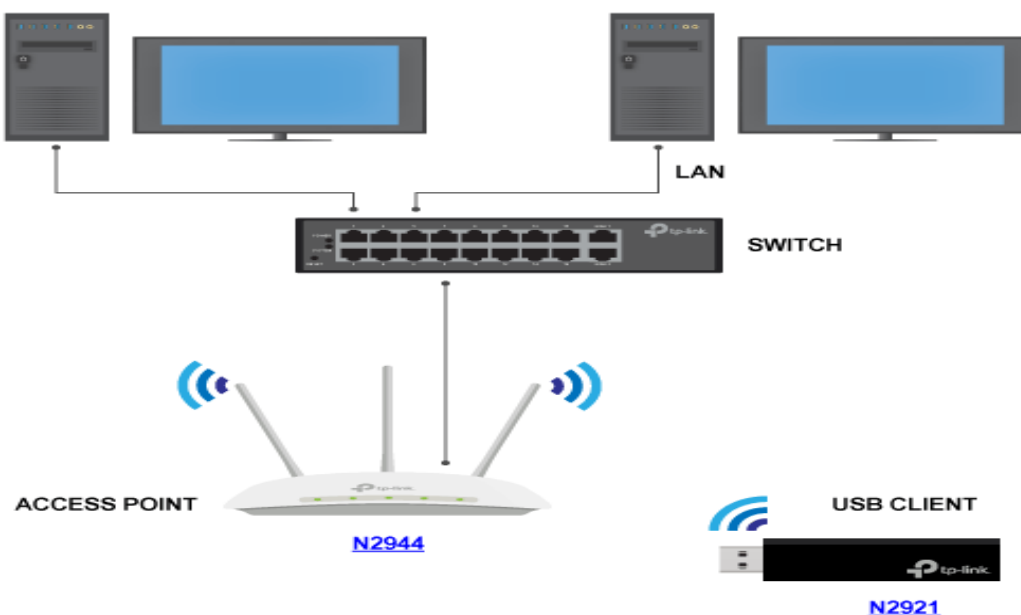


Rysunek 9. Rodzaje sieci bezprzewodowych.

Źródło: www.mechatronika-simr.home.pl [09.12.2019]

Istniejące już sieci bezprzewodowe wymagają urządzeń, które będą pełniły funkcje łączników. Rolę taką pełnią punkty dostępu. Można wyróżnić kilka trybów pracy punktów dostępu:

1. Access Point – zapewnia połączenie sieci bezprzewodowej sieci lokalnej z siecią lokalną dzięki bezprzewodowym kartom sieciowym. (Rysunek 10)



Rysunek 10. Tryb pracy Access Point.

Źródło: https://www.dipol.com.pl/tryby_pracy_access_pointa_na_przykladzie_urzadzenia_tp-link_tl-wa901nd_bib505.htm [09.12.2019]

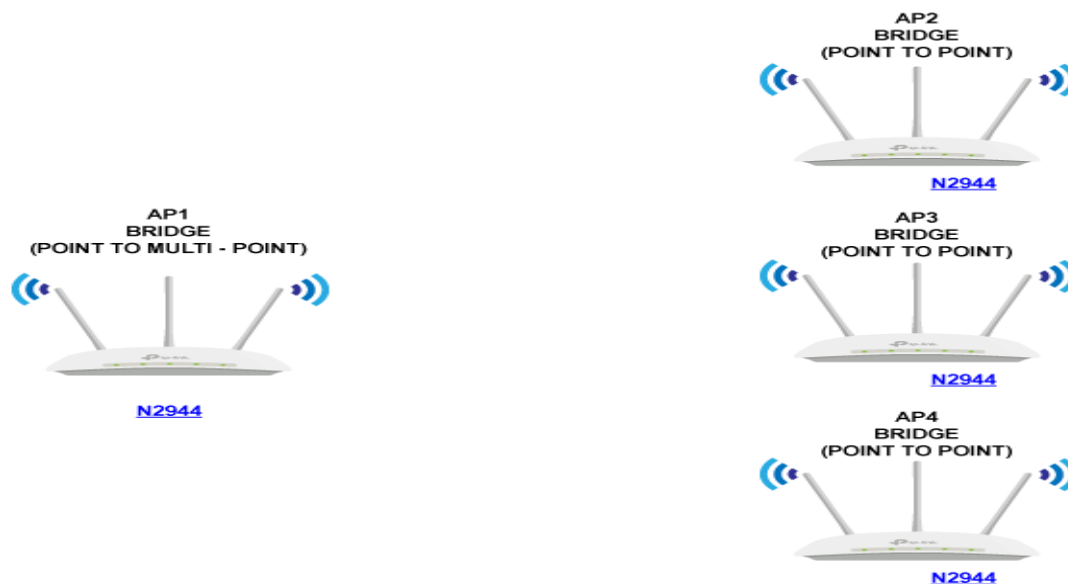
2. Ad Hoc – urządzenie podłączone może pełnić funkcję zarówno klienta jak i punktu dostępu. Jest to rodzaj sieci o zdecentralizowanej strukturze.
3. Bridge Point to Point – tryb taki pozwala na połączenie istniejących już sieci LAN. Działanie sieci opiera się na routingu statycznym, który bazuje na adresach MAC łączonych ze sobą punktów dostępu. (Rysunek 11)



Rysunek 11. Tryb pracy BPTP.

Źródło: https://www.dipol.com.pl/tryby_pracy_access_pointa_na_przykladzie_urzadzenia_tp-link_tl-wa901nd_bib505.htm [09.12.2019]

4. Bridge Point to Multipoint – tryb taki pozwala na połączenie odległych segmentów sieci z kilku lokalizacji. (Rysunek 12)



Rysunek 12. Tryb pracy BPTM.

Źródło: https://www.dipol.com.pl/tryby_pracy_access_pointa_na_przykladzie_urzadzenia_tp-link_tl-wa901nd_bib505.htm [09.12.2019]

3.3 Technologia WiFi

Jest to technologia wykorzystywana podczas tworzenia sieci WLAN. Podstawą działania jest komunikacja radiowa, która wykorzystuje pasmo o częstotliwości 2,4 i 5 GHz. W celu usystematyzowania wprowadzono standardy IEEE 802.11:

1. 802.11 – wprowadzony w 1997 roku. Standard określał dwie częstotliwości transmisji 2,4 i 5 GHz.
2. 802.11b – pozwalał osiągnąć zasięg ok 100 metrów w otwartej przestrzeni oraz około 50 metrów w pomieszczeniu. Urządzenia pracujące w tym standardzie mogą przysyłać dane z prędkością do 11 Mb/s. Częstotliwość pracy wynosi 2,4 GHz, obejmując 13 kanałów o szerokości 22 MHz.
3. 802.11a – Urządzenia pracujące w tym standardzie pracują z częstotliwością 5 GHz. Podstawowa prędkość transmisji wynosi 54 Mb/s.
4. 802.11g – wprowadzony w czerwcu 2003 roku. Urządzenia pracujące zgodnie z tym standardem wykorzystują częstotliwość 2,4 GHz podobnie jak w standardzie 802.11b. W tym przypadku prędkość transmisji wynosi do 54 Mb/s.
5. 802.11n – został zatwierdzony w 2009 roku. Wykorzystuje dwie częstotliwości: 2,4 i 5 GHz. Prędkość transmisji wynosi 300 Mb/s dla 5 GHz oraz 150 Mb/s dla 2,4 GHz.
6. 802.11ac – zaprezentowany w 2012 roku. Prędkość do 1 Gb/s.
7. 802.11ax – zaprezentowany w 2017 roku. Prędkość transmisji wynosi do 10 Gb/s.

Wi-fi bazuje na protokołach warstwy fizycznej takich jak:

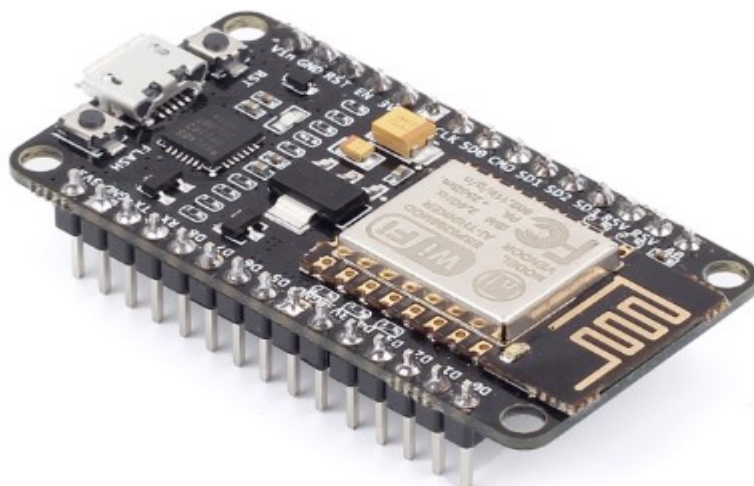
1. DSSS – częstotliwość fali nośnej jest stała. Każdy bit informacji jest zastąpiony przez ciąg bitów. Ułatwia to detekcję oraz korekcję błędów, ponadto minimalizuje liczbę retransmisji.
2. FHSS – w systemach tych częstotliwość fali nośnej zmienia się skokowo i periodycznie w takich samych odstępach czasu.
3. OFDM – pozwala na transmisję wielu strumieni danych na wzajemnie prostopadłych częstotliwościach nośnych.

4 Koncepcja projektu manipulatora

4.1 Moduł Wi-Fi NodeMCU V2

ESP8266 jest obecnie jednym z najbardziej popularnych kontrolerów z wbudowanym Wi-Fi, jednakże wymaga stabilnego zasilania, zewnętrznego programatora. Wymienione powyżej problemy rozwiązuje zastosowanie płytek posiadające już wlutowany moduł Wi-Fi.

NodeMCU V2 jest jedną z wielu płytek, które posiadają wbudowany moduł ESP8266. Ponadto posiada wbudowany konwerter USB-UART, który pozwala zaprogramować układ przez USB bez dodatkowych elementów. Do tego celu można wykorzystać Arduino IDE lub język skryptowy Lua. Płytkę posiada wewnętrzny stabilizator napięcia, dzięki temu może być zasilana poprzez port USB lub zewnętrzne źródło zasilania 7-12 woltów. Sam mikrokontroler i moduł Wi-Fi wykorzystuje logikę 3,3 V. (Rysunek 13)



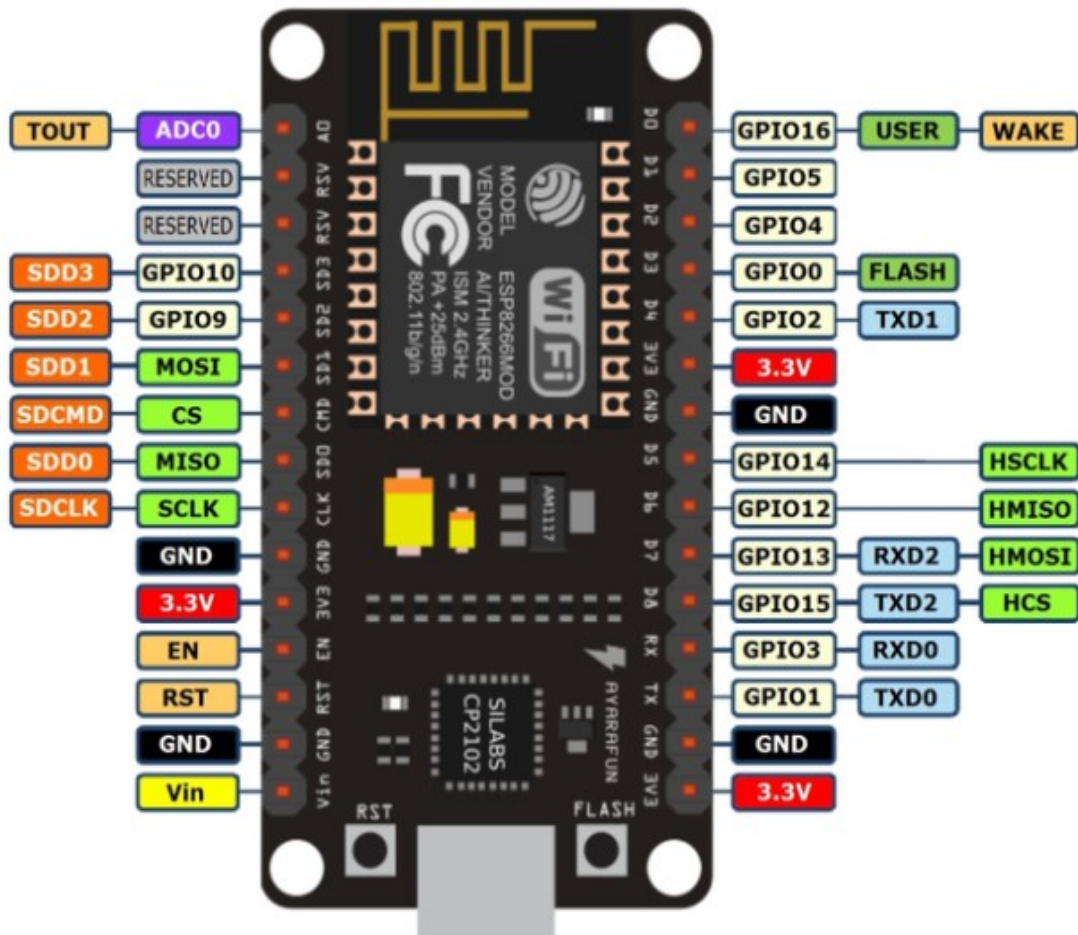
Rysunek 13. NodeMCU V2.

Źródło: <https://nettigo.pl/products/modul-wifi-nodemcu-v2-bezprzewodowy-modul-oparty-na-esp8266-12e> [09.12.2019]

NodeMCU V2 posiada:

- Moduł ESP8266-12E z anteną PCB.
- Łączność Wi-Fi w standardzie 802.11 b/g/n.
- Może pracować w trybach Access Point, Standalone (STA), Access Point+STA.
- Może obsługiwać: WPA/WPA2, WPS.
- Zasilanie: 3,3 V (wewnętrzne), 5 V (przez USB), 7-12 V (zewnętrzne).
- Procesor RISC o taktowaniu 80MHz.
- Pamięć flash 4Mb.
- 16 portów GPIO, które mogą obsługiwać PWM, I2C, SPI. (Rysunek 14)
- Maksymalne natężenie prądu w pinach GPIO może wynosić do 12mA.
- Konwerter USB-UART CP2102.

- 10 bitowy przetwornik ADC.
- 30 pinów w rastrze 2,54 mm, bez problemów pasuje do płytek stykowych.
- Złącze micro USB.
- Wymiary 49x25 mm.



Rysunek 14. Rozmieszczenie pinów na płytce NodeMCU V2.

Źródło: <https://iotbytes.wordpress.com/nodemcu-pinout/> [09.12.2019]

4.2 Serwomechanizm SG-92R i MG996R

Serwomechanizm jest urządzeniem wykonawczym, którego zadaniem jest precyzyjne położenie kątowe. W konstrukcji można wyróżnić kilka składowych elementów:

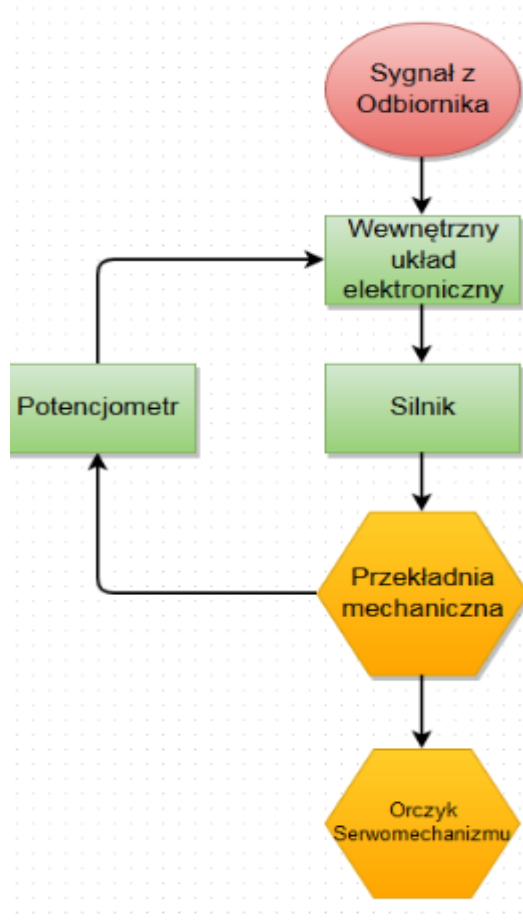
1. Silnik napędowy prądu stałego
2. Potencjometr
3. Elektroniczny układ sterowania
4. Przekładnia zębata
5. Pokrywa (Rysunek 15)



Rysunek 15. Podstawowe elementy serwomechanizmu.

Źródło: <http://abc-modele.pl/serwomechanizmy/> [09.12.2019]

Zasadę działania przedstawiono na Rysunku 16



Rysunek 16. Schemat działania serwomechanizmu.

Źródło: <http://abc-modele.pl/serwomechanizmy/> [09.12.2019]

W zależności od różnicy pomiędzy położeniem orczyka a zadaniem, układ elektroniczny wysyła do silnika sygnał sterujący o większym lub mniejszym wypełnieniu. Motor z kolei napędza przekładnię mechaniczną, która jest połączona z potencjometrem. Zmiana położenia potencjometru powoduje zmianę rezystancji odczytaną przez układ sterujący. Gdy serwomechanizm osiągnie zadaną pozycję, silnik przestaje działać.

Ze względu na wielkość serwomechanizmy można sklasyfikować do poszczególnych grup:

- Micro – o wadze do 10 gramów
- Mini – o wadze do 25 gramów
- Standard – o wadze około 50 gramów

Ze względu na sposób sterowania:

- Analogowe – podczas ruchu przekładni obraca się również potencjometr, dzięki temu zmienia się rezystancja a więc i napięcie odłożone na nim, które jest odczytywane przez układ elektroniczny i porównywane ze skonwertowanym sygnałem wejściowym. Wypełnienie sygnału PWM sterującego silnikiem jest zależne od różnicy tychże napięć.
- Cyfrowe – wewnętrzny układ elektroniczny wyposażony jest w mikrokontroler, który dokonuje porównania w dziedzinie cyfrowej aktualnej pozycji serwomechanizmu z zadaną.

Serwomechanizmy cyfrowe posiadają szybszą reakcję na sygnał oraz są bardziej precyzyjne.

Serwomechanizm SG-92R jest jednym z urządzeń, które posiadają największy moment obrotowy wśród serw klasy micro. Posiada koła zębate, które są wykonane z włókna węglowego. (Rysunek 17)

Parametry pracy:

- Napięcie zasilania: 4,8-7,2 voltów
- Moment obrotowy: 0,24 Nm (dla 4,8 V)
- Prędkość obrotowa: 0,1s/60 deg
- Pobór prądu: 0,5 A



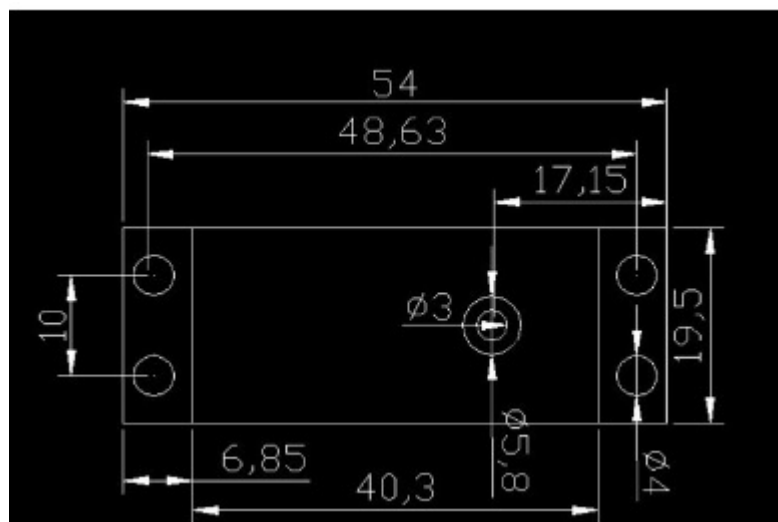
Rysunek 17. Serwomechanizm SG-92.

Źródło: www.abc-rc.pl [09.12.2019]

Parametry pracy:

-

Źródło: www.abc-rc.pl [09.12.2019]

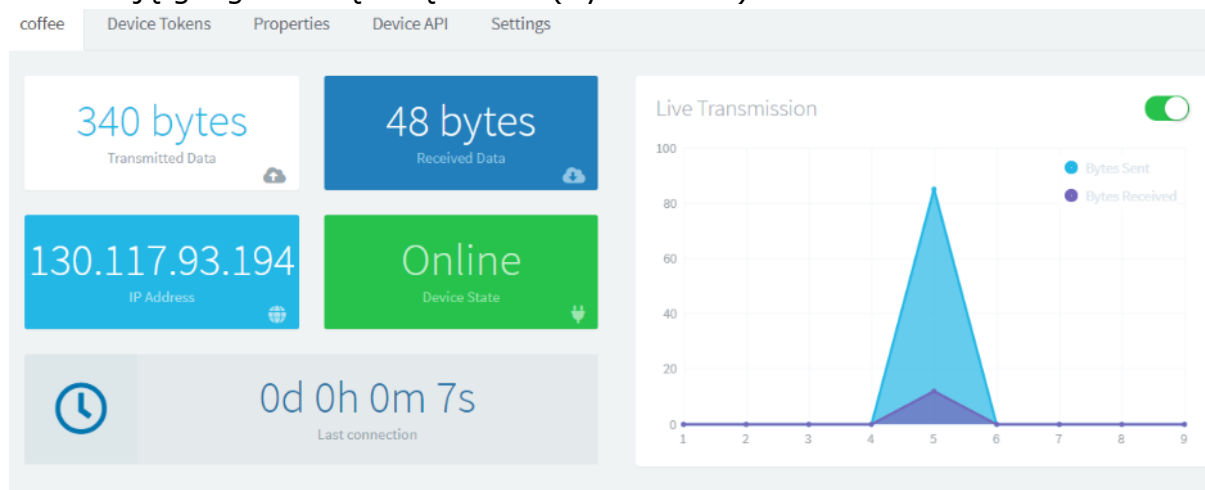


Źródło: www.abc-rc.pl [09.12.2019]

4.3 Chmura danych Thinger.io

Do poprawnego działania systemu wykorzystującego koncepcję Internetu rzeczy poza modułami komunikacyjnymi wymagany jest „łącznik” pomiędzy interfejsem operatora a manipulatorem. Funkcję taką pełni chmura danych. Jest to model przetwarzania danych, w którym dane są przechowywane na serwerze usługodawcy, zaś klient ma stały dostęp do jego zawartości. Na chwilę obecną jest to jedna z najbardziej rozwijanych technologii. Najbardziej znane chmury danych to: Microsoft Azure IoT Suite, Google Cloud’s IoT Platform. Zaletą ich jest przesyłanie danych w czasie rzeczywistym, wizualizacja ich oraz wsparcie sprzętowe. Do największych wad należą: wysoka cena lub poważne ograniczenia dla darmowych kont.

Thingier.io jest chmurą danych Open Source dla urządzeń Internetu rzeczy. Darmowe konto posiada tylko kilka ograniczeń np. ilość danych, które można przechować w tzw. wiadrze danych. Zaletą tej platformy jest ogromna ilość bibliotek i przykładów programów dla urządzeń takich jak Raspberry PI lub Arduino, co pozwala w stosunkowo prosty sposób stworzyć własny system Internetu rzeczy. Ponadto użytkownik może stworzyć własny interfejs, który będzie mógł wyświetlać dane w czasie rzeczywistym lub sterować urządzeniami. Chmura danych Thinger.io pozwala na podgląd takich danych jak: adres IP urządzenia z którym się komunikuje, ilość bajtów odebranych oraz wysłanych oraz lokalizację geograficzną urządzenia. (Rysunek 20)



Rysunek 20. Okno interfejsu chmury Thinger.io.

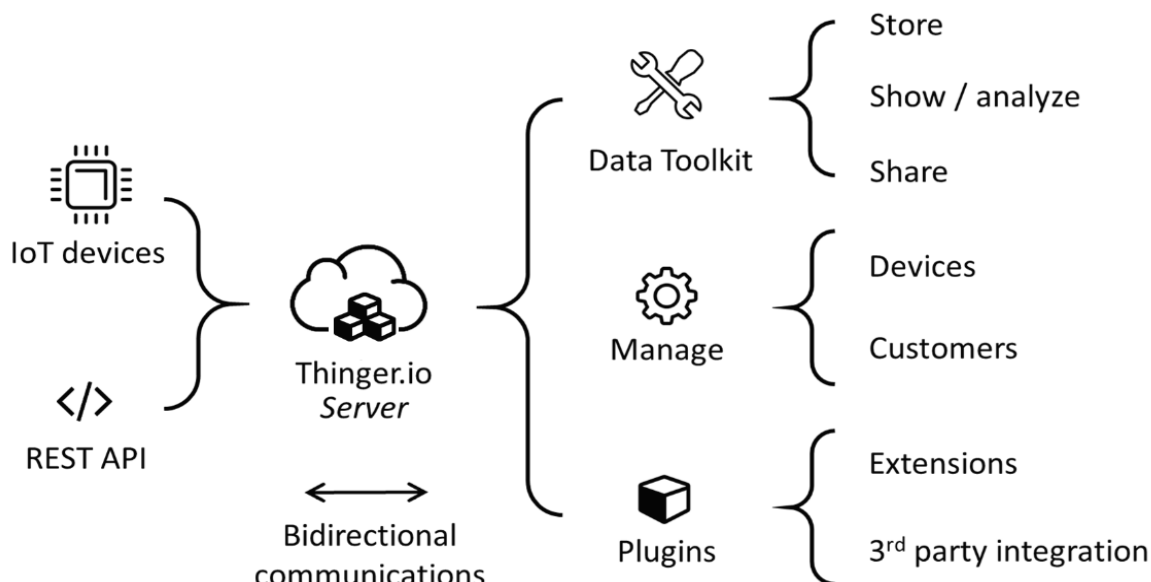
Źródło: www.docs.thinger.io [09.12.2019]

Komunikacja pomiędzy urządzeniami Internetu rzeczy a chmurą Thinger.io odbywa się z wykorzystaniem protokołu sieciowego WebSocket. Jest to technologia dzięki której istnieje możliwość komunikacji dwukierunkowej pomiędzy serwerem a klientem. Rozwiązanie takie znacząco zmniejsza obciążenie sieci oraz zapewnia przesył danych w czasie rzeczywistym. Nawiązanie połączenia odbywa się w następujący sposób:

- Wysłanie żądania HTTP przez klienta do serwera

- Nawiązanie połączenia przez HTTP
- Aktualizacja połączenia, jeśli serwer obsługuje WebSocket
- Zastąpienie połączenia HTTP przez WebSocket, korzystając z tego samego protokołu TCP/IP

Od tej chwili jest możliwa komunikacja dwustronna pomiędzy chmurą a urządzeniem. (Rysunek 21)



Rysunek 21. Schemat działania chmury danych.
Źródło: www.docs.thingier.io [09.12.2019]

4.4 Konstrukcja mechaniczna

4.4.1 Założenia

Głównym założeniem podczas realizacji projektu konstrukcji mechanicznej manipulatora było wykorzystanie technologii FDM do wykonania poszczególnych elementów urządzenia. Pierwszym problemem był odpowiedni dobór materiału, z którego zostaną wykonane komponenty manipulatora. Głównymi materiałami wykorzystywanymi w technologii druku 3d są PLA, ABS, CPE, Nylon. Właściwości mechaniczne przedstawiono w Tabeli 1.

Materiał	ABS	PLA	CPE	Nylon
Gęstość[kg/m ³]	1060	1240	960	1150
Moduł Younga [MPa]	1618	2340	1537	579
Wytrzymałość na rozciąganie [MPa]	39	49	41	34
Wytrzymałość na zginanie [MPa]	70	103	80	24

Tabela 1. Właściwości mechaniczne materiałów wykorzystywanych w druku 3d.
Źródło: <https://ultimaker.com/en/resources/manuals/materials> [09.12.2019]

Spośród tych materiałów PLA posiada najlepsze właściwości wytrzymałościowe, jednakże z drugiej strony ma największą gęstość. PLA zostało użyte do

wytworzenia elementów manipulatora. Maksymalne naprężenie zredukowane według hipotezy Hubera-Misesa w każdym elemencie konstrukcji nie może przekroczyć 25 MPa. Wartość ta wynika z przyjętego współczynnika bezpieczeństwa $X=2$ (wzór 1).

$$R_z = \frac{R_e}{X} \quad (1)$$

Gdzie:

R_z – naprężenie zredukowane [MPa]

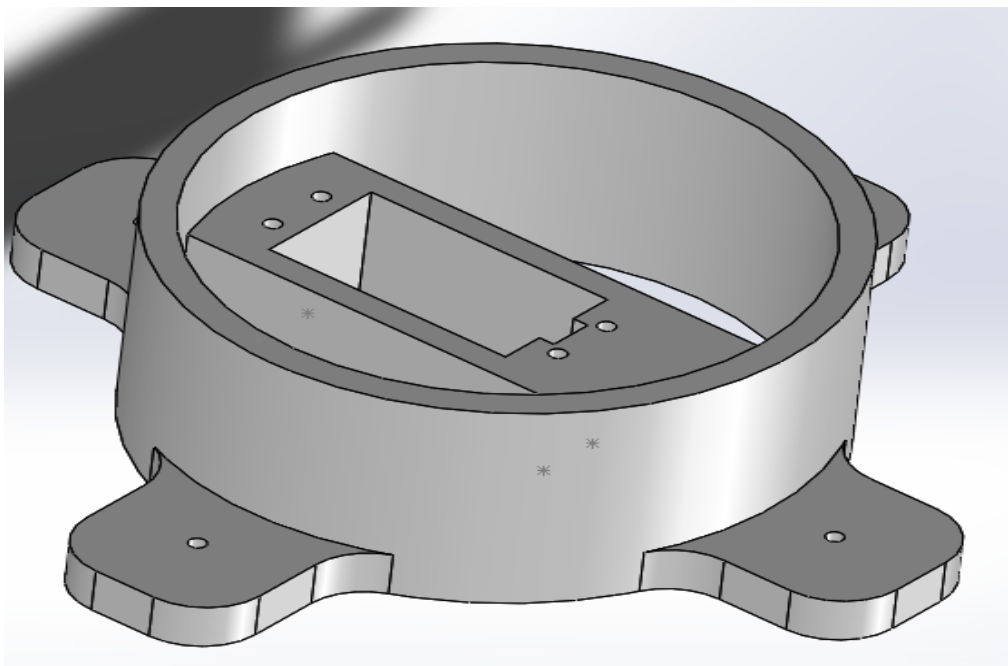
R_e – wytrzymałość na rozciąganie [MPa]

X – współczynnik bezpieczeństwa

4.4.2 Realizacja

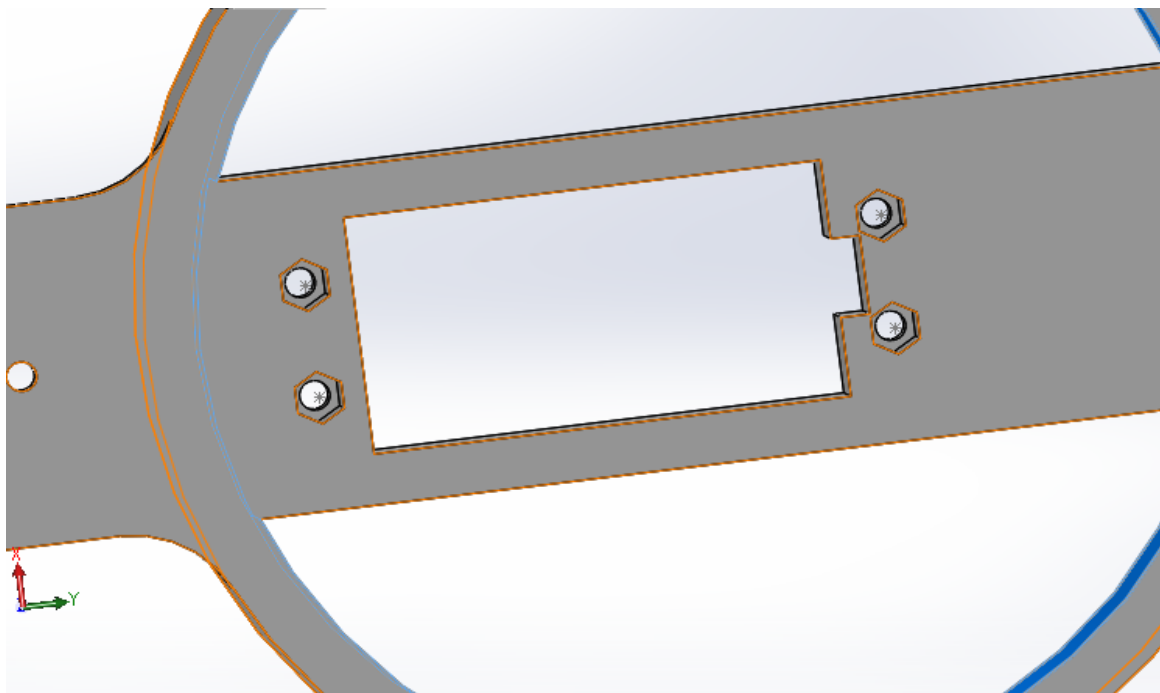
Wszystkie elementy konstrukcji manipulatora przegubowego zostały zaprojektowane w środowisku 3D CAD Solidworks. Głównymi komponentami urządzenia były podstawa, ramiona i osprzęt chwytający.

a) Podstawa



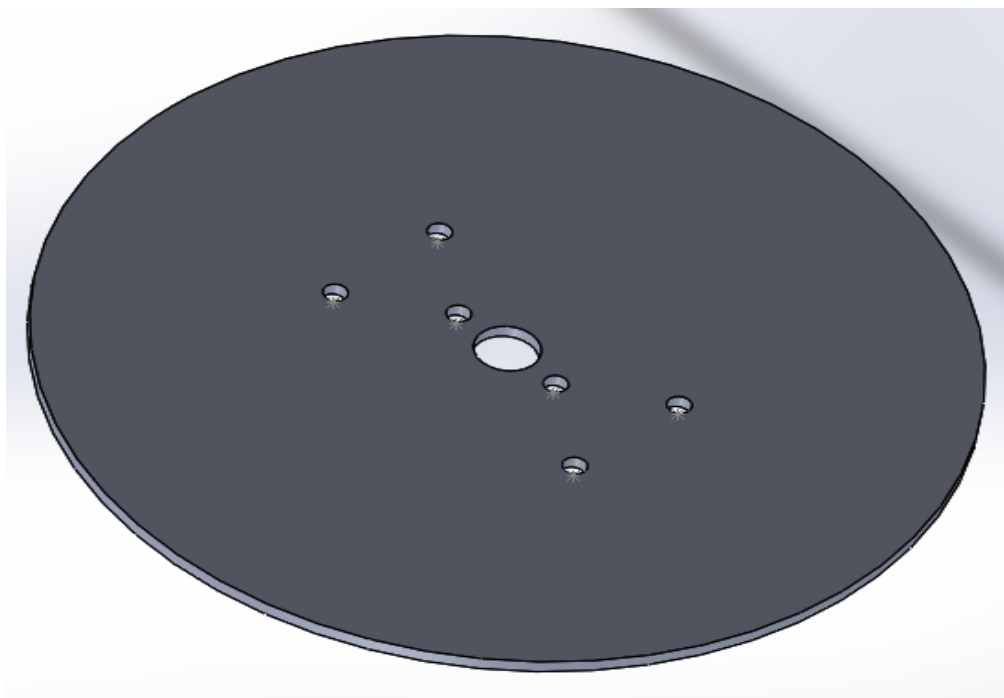
Rysunek 22. Model podstawy.

Rysunek 22 przedstawia model podstawy manipulatora. Jego wymiary gabarytowe wynoszą: 150x150x49 mm. Posiada gniazdo, w którym będzie osadzony serwomechanizm MG 996. Mocowanie aktuatora będzie możliwe poprzez zastosowanie czterech śrub M2 z nakrętkami. Obrót nakrętek zostanie zablokowany dzięki pogłębieniom sześciokątnym (Rysunek 23). Cztery otwory pod śruby M2 zlokalizowane w podporach podstawy zapewniają możliwość mocowania do stołu.



Rysunek 23. Koncepcja mocowania silnika.

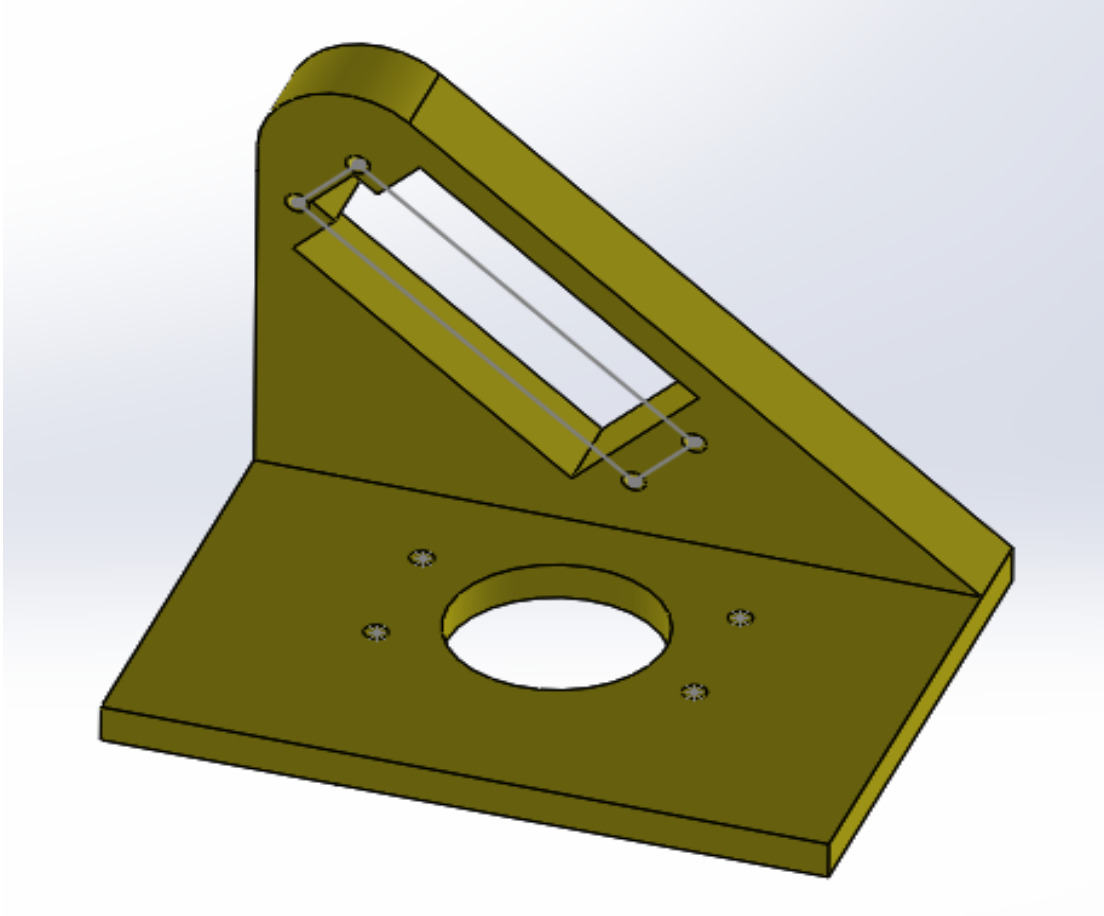
b) **Dekiel**



Rysunek 24. Model dekla.

Rysunek 24 przedstawia model dekla, który łączy orczyk serwomechanizmu umiejscowionego w podstawie a mocowaniem aktuatora odpowiedzialnego za obrót ramienia. Wymiary elementu wynoszą: $\Phi 98$ mm i 2 mm wysokości. Mocowanie dekla realizowane jest za pomocą śrub M2.

c) **Mocowanie mechanizmu obrotu ramienia**



Rysunek 25. Model mocowania mechanizmu obrotu ramienia.

Rysunek 25 przedstawia model mocowania mechanizmu obrotu ramienia. Wymiary gabarytowe wynoszą: 80x60x63 mm. Rysunek 26 przedstawia schemat obciążenia mocowania. Do elementu jest zamocowany serwomechanizm za pomocą 4 śrub M2, których napięcie wstępne wyznaczono ze wzoru na naciski powierzchniowe (wzór 2):

$$p = \frac{F}{S} \quad (2)$$

Gdzie:

p- nacisk powierzchniowy dla PLA $p=10\text{MPa}$

F-siła

S-pole powierzchni styku nakrętki (wzór 3):

$$S = \pi \frac{D^2 - d^2}{4} \quad (3)$$

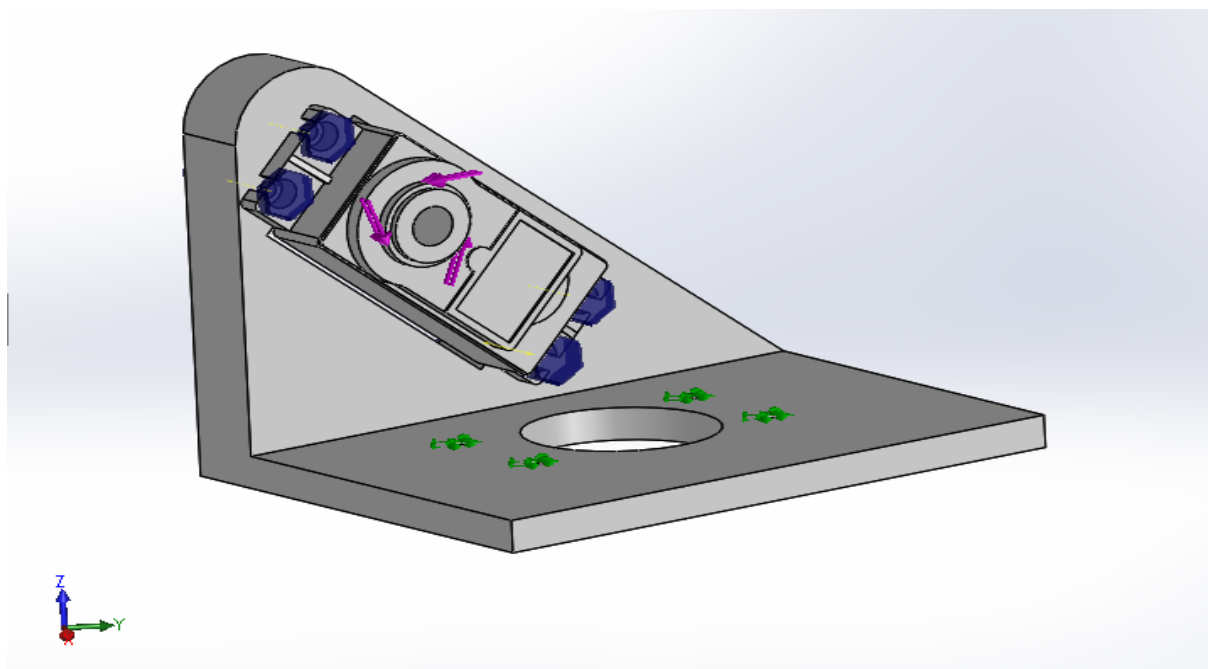
Przyjmując: $D=4\text{mm}$ oraz $d=2\text{ mm}$

Po przekształceniu otrzymujemy (wzór 4):

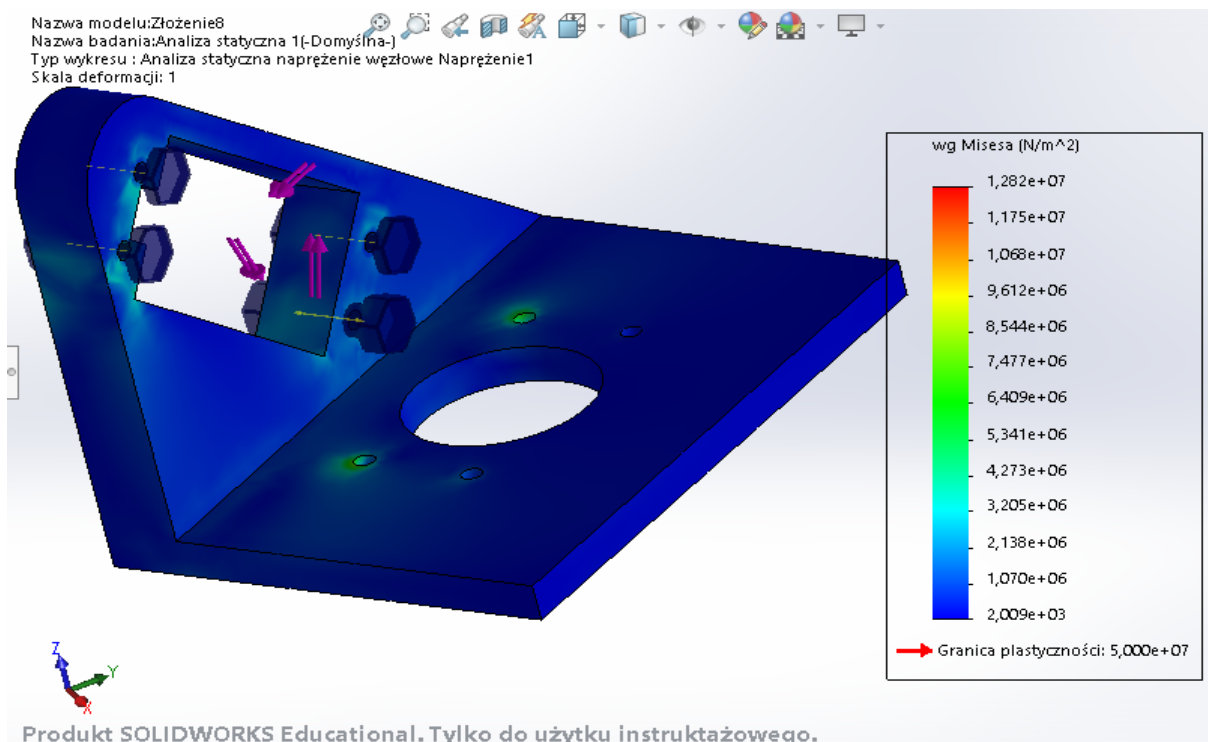
$$F = p \cdot \pi \cdot \frac{D^2 - d^2}{4} \quad (4)$$

Po podstawieniu: $F=100\text{N}$

Do korpusu serwomechanizmu został przyłożony moment obrotowy o wartości 1,5 Nm, czyli maksymalna wartość jaką może wytworzyć aktuator. Mocowanie zostało unieruchomione poprzez замуrowanie otworów przelotowych w podstawie.



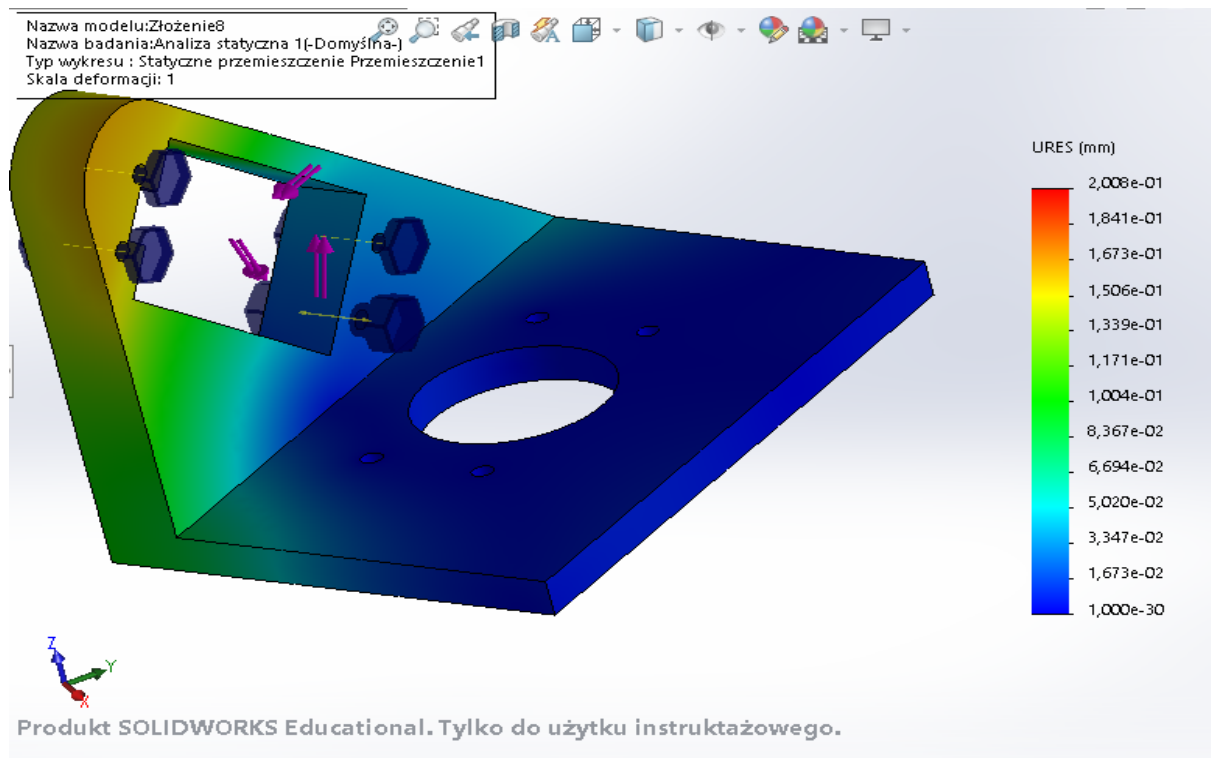
Rysunek 26. Schemat obciążenia mocowania.



Rysunek 27. Wykres rozkładu naprężeń zredukowanych w mocowaniu.

Rysunek 27 przedstawia rozkład naprężeń w konstrukcji mocowania mechanizmu obrotu ramienia. Maksymalna wartość naprężenia zredukowanego

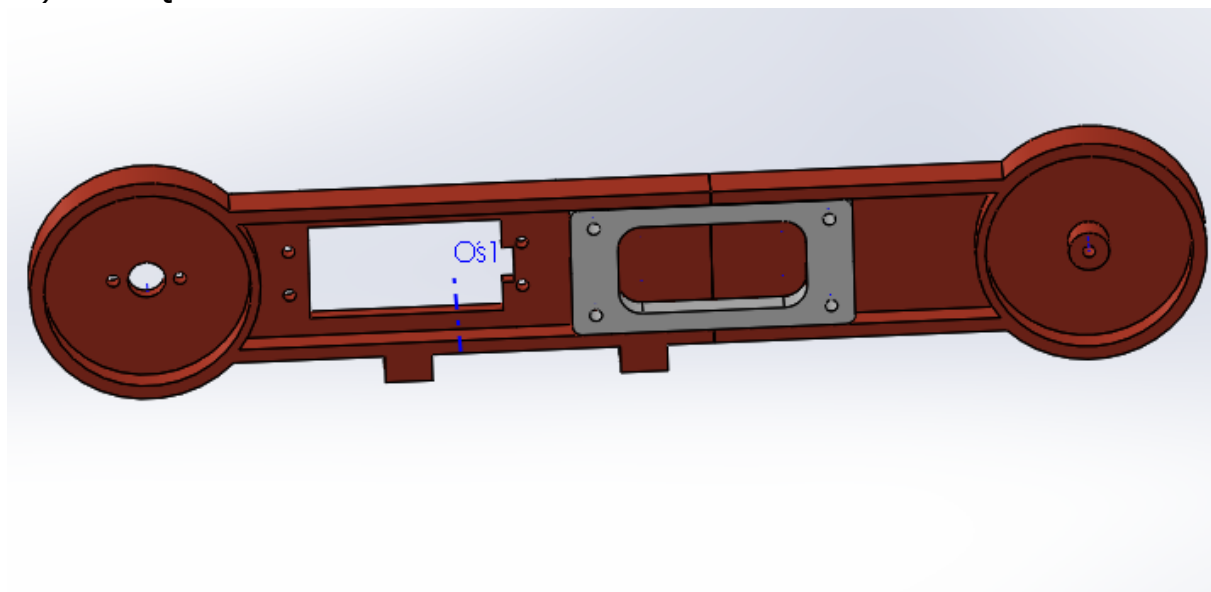
według hipotezy Hubera-Misesa wynosi 12,82 MPa, która została wyznaczona w punkcie zamurowania elementu.



Rysunek 28. Wykres rozkładu przemieszczeń mocowania.

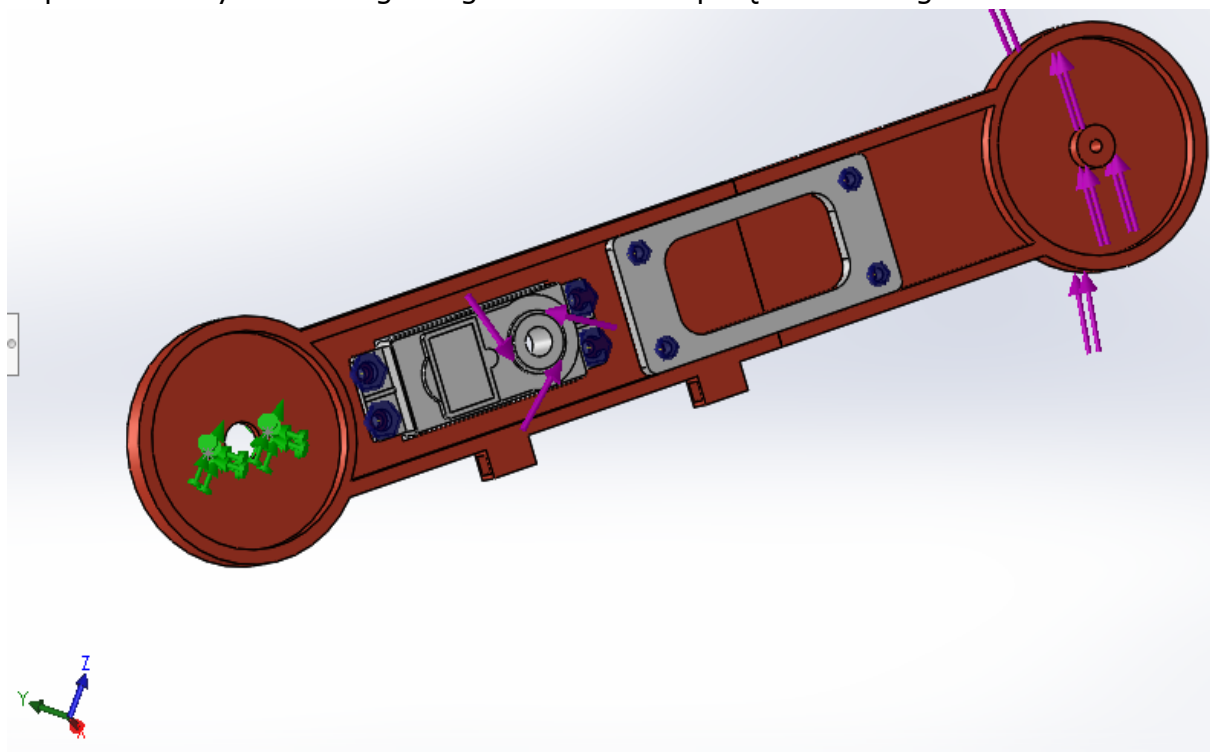
Rysunek 28 przedstawia wyniki symulacji przemieszczeń. Maksymalna wartość przemieszczenia wyznaczono dla górnego skrajnego położenia elementu, które wynosi 0,2 mm.

d) **Ramię dolne**



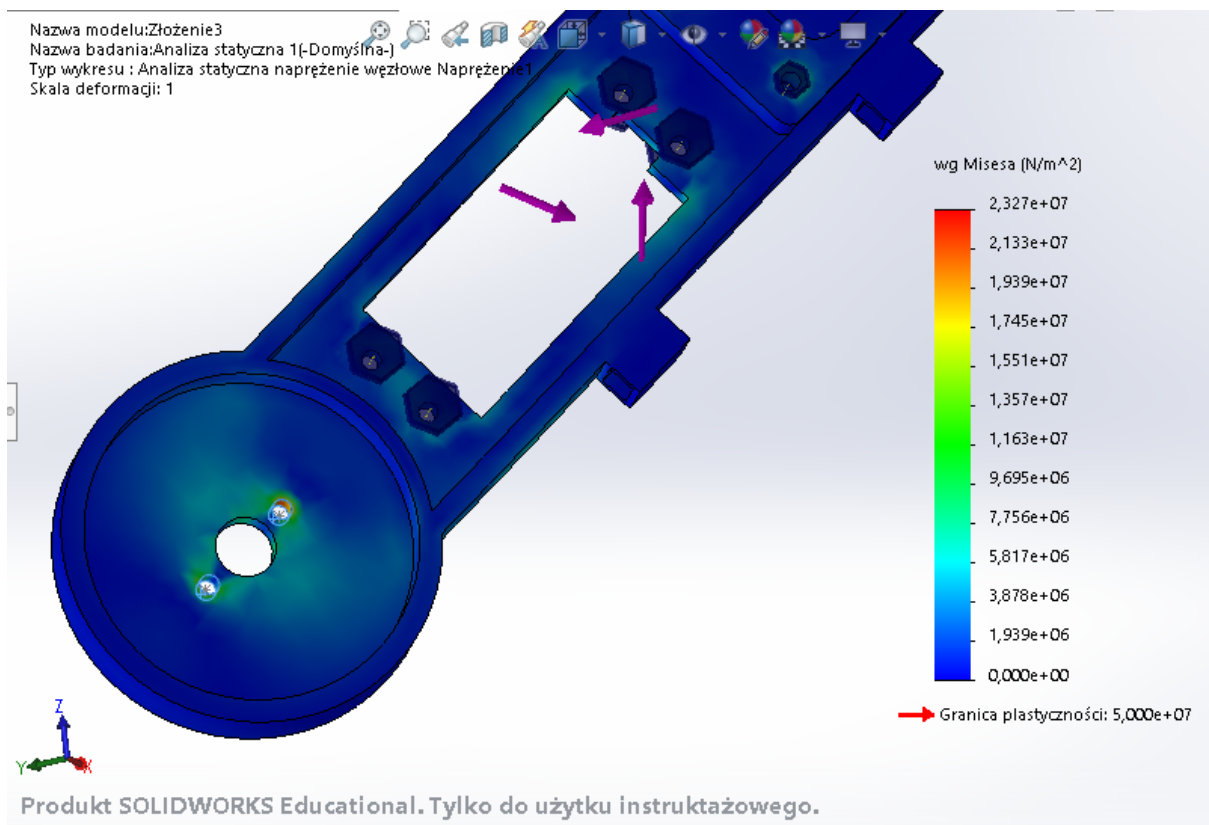
Rysunek 29. Model ramienia dolnego.

Rysunek 29 przedstawia model dolnego ramienia. Składa się z trzech komponentów: części dolnej, górnej oraz płytki która łączy obie części ramienia. Dolna część posiada wnękę na serwomechanizm MG 996, który jest odpowiedzialny za obrót górnego ramienia i osprzętu roboczego.

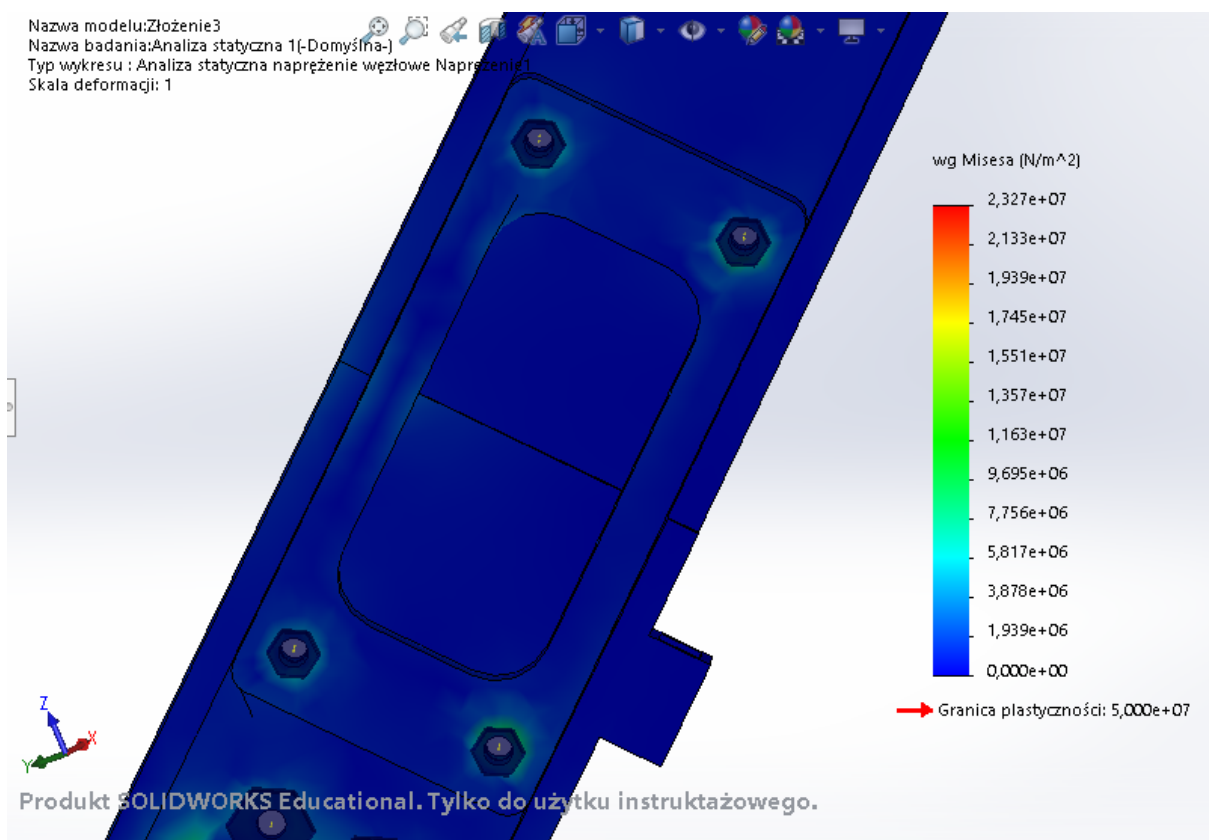


Rysunek 30. Schemat obciążenia ramienia.

Rysunek 30 przedstawia schemat obciążenia ramienia. Unieruchomione zostały dwa otwory przelotowe, służące do mocowania elementu do orczyka serwomechanizmu umiejscowionego w mocowaniu mechanizmu obrotu ramienia. Aktuator odpowiedzialny za obrót ramienia górnego został obciążony momentem o wartości 1,5 Nm względem osi obrotu orczyka oraz unieruchomiony za pomocą 4 śrub M2 o napięciu wstępnym 100 N. Górna część została poddana obciążeniu o wartości 1,5 Nm względem osi obrotu przechodzącej przez otwór przelotowy o średnicy 8 mm. Płytką łączącą część dolną i górną ramienia została przytwierdzona za pomocą 4 śrub M2 o napięciu wstępnym 100 N, oraz określono zestaw kontaktowy pomiędzy wewnętrznymi ścianami ramienia o zewnętrznymi płytki. Ponadto zdefiniowano kontakt pomiędzy dwoma częściami ramienia.



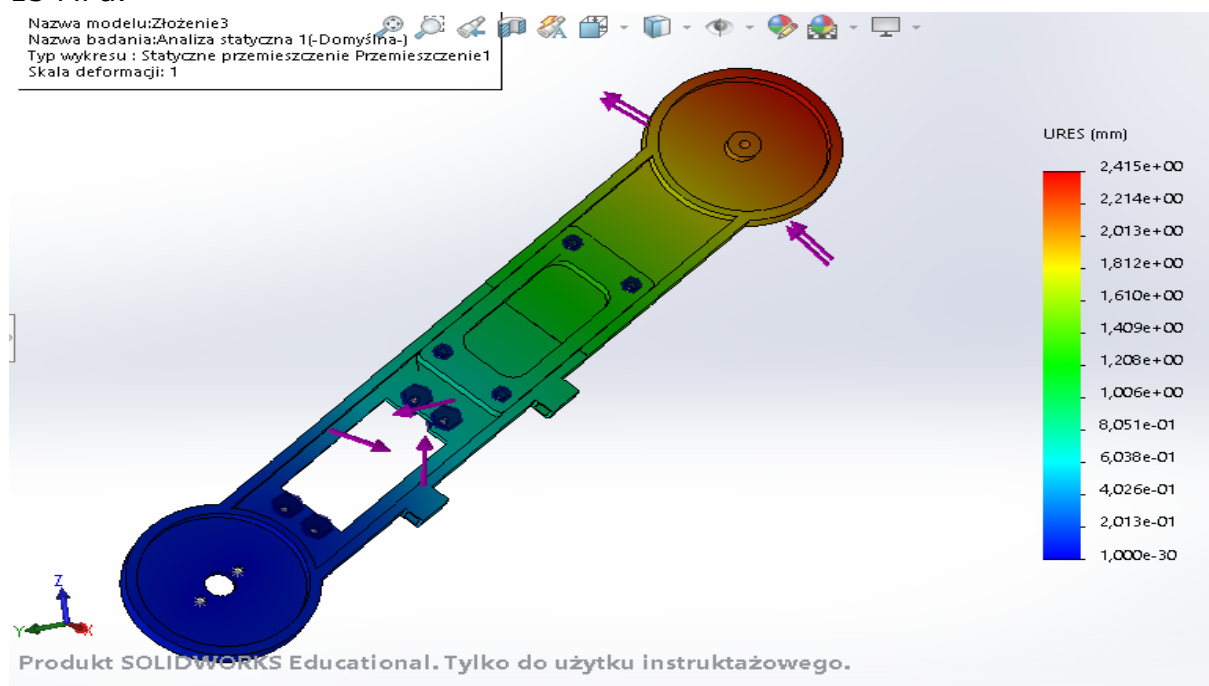
Rysunek 31. Wykres rozkładu naprężeń zredukowanych w ramieniu.



Rysunek 32. Wykres rozkładu naprężeń w płycie.

Rysunek 31 przedstawia rozkład naprężeń zredukowanych według hipotezy Hubera-Misesa. Maksymalna jego wartość wynosi 23,27 MPa, najbardziej obciążonym miejscem jest otwór mocujący ramię do orczyka serwomechanizmu MG 996.

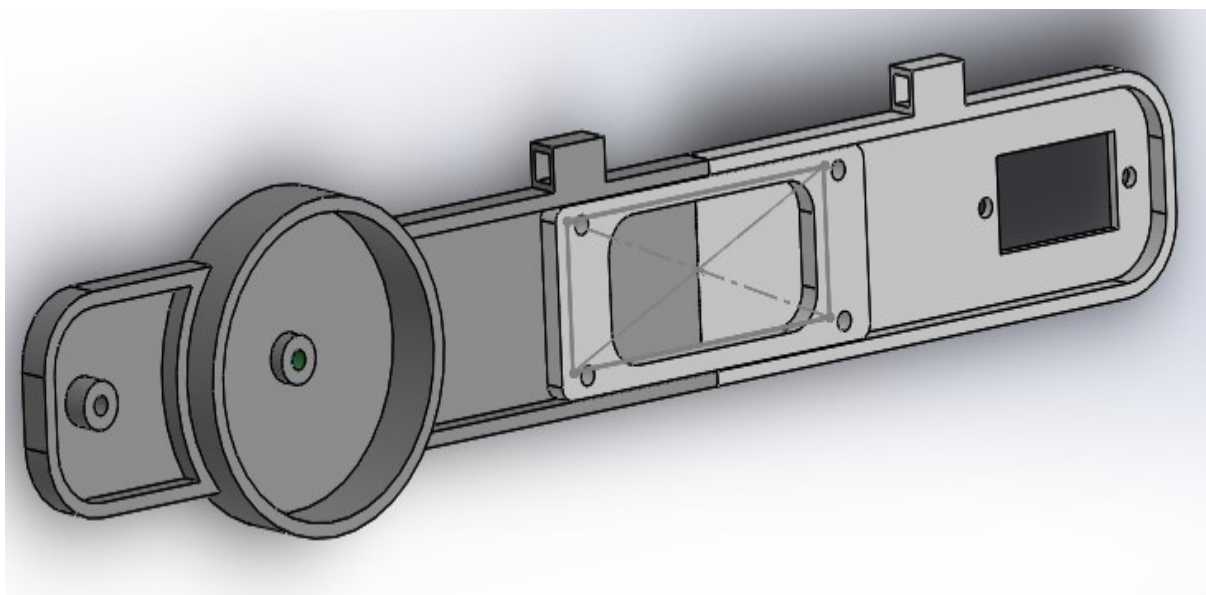
Rysunek 32 przedstawia rozkład naprężeń zredukowanych w płycie oraz punkcie styku dwóch części ramienia. Maksymalne ich wartości nie przekraczają 15 MPa.



Rysunek 33. Wykres rozkładu przemieszczeń w ramieniu.

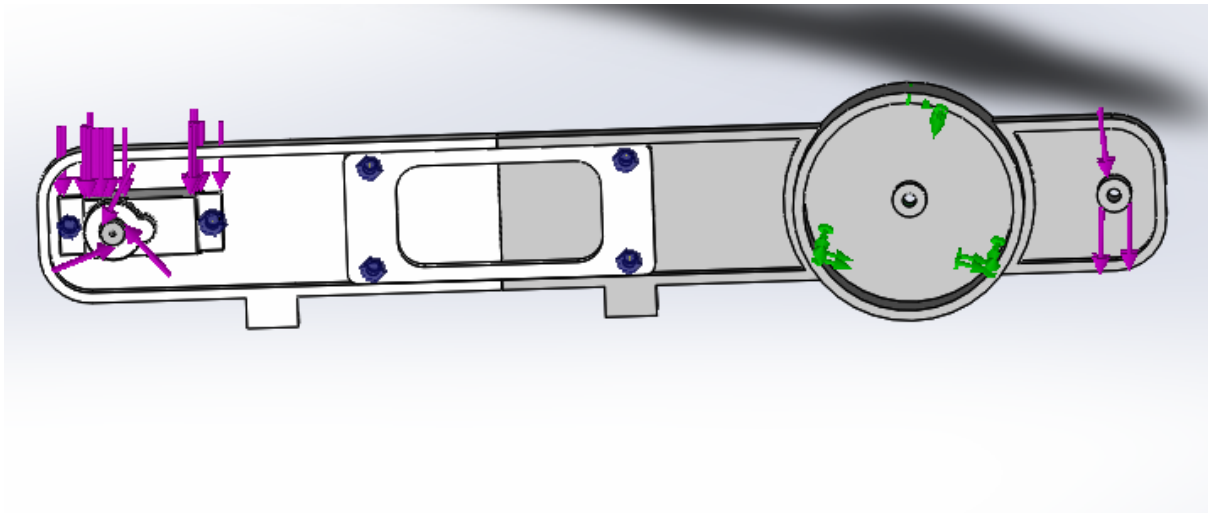
Rysunek 33 przedstawia rozkład przemieszczeń obciążonego ramienia. Maksymalna strzałka ugięcia wynosi 2,45 mm.

e) Ramię górne



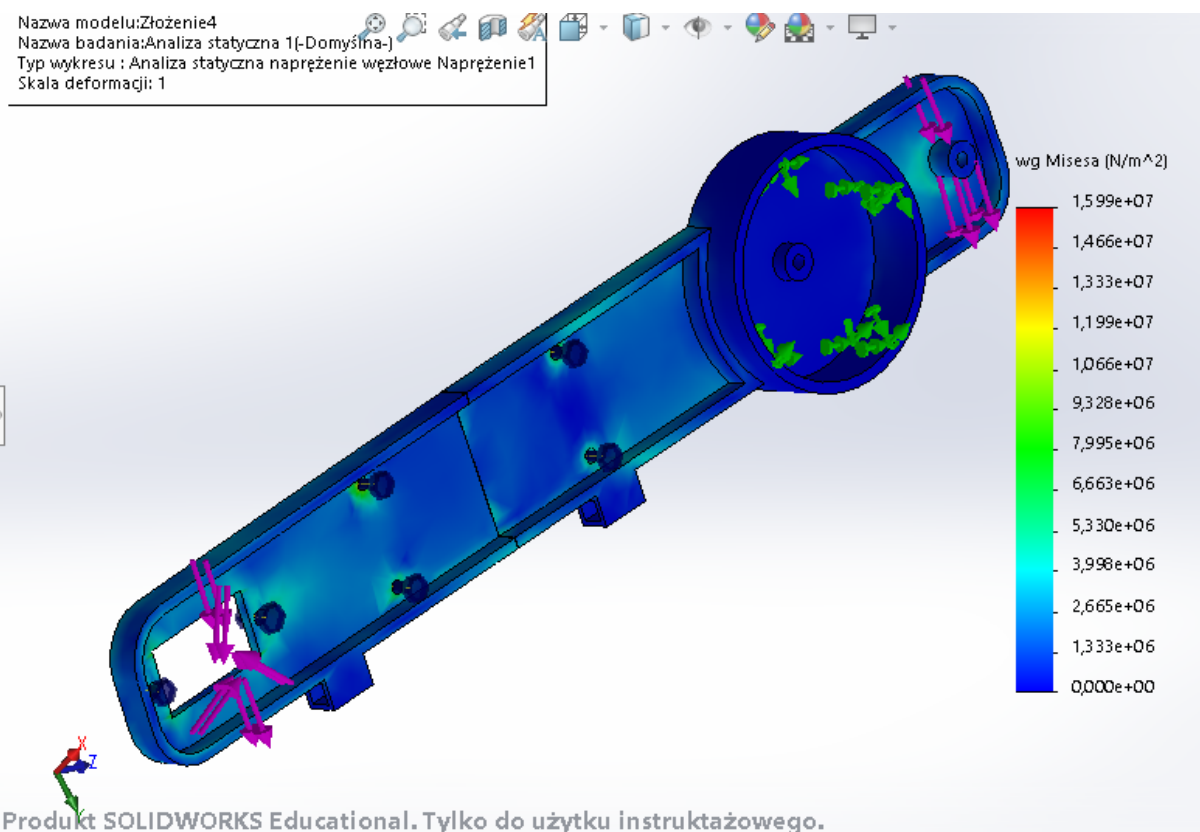
Rysunek 34. Model ramienia górnego.

Rysunek 34 przedstawia model ramienia górnego. Składa się ono z trzech części: części dolnej, górnej oraz płytki łączącej oba elementy. Górna część posiada wnękę w której został umieszczony serwomechanizm SG-92, który odpowiada za obrót osprzętu roboczego.



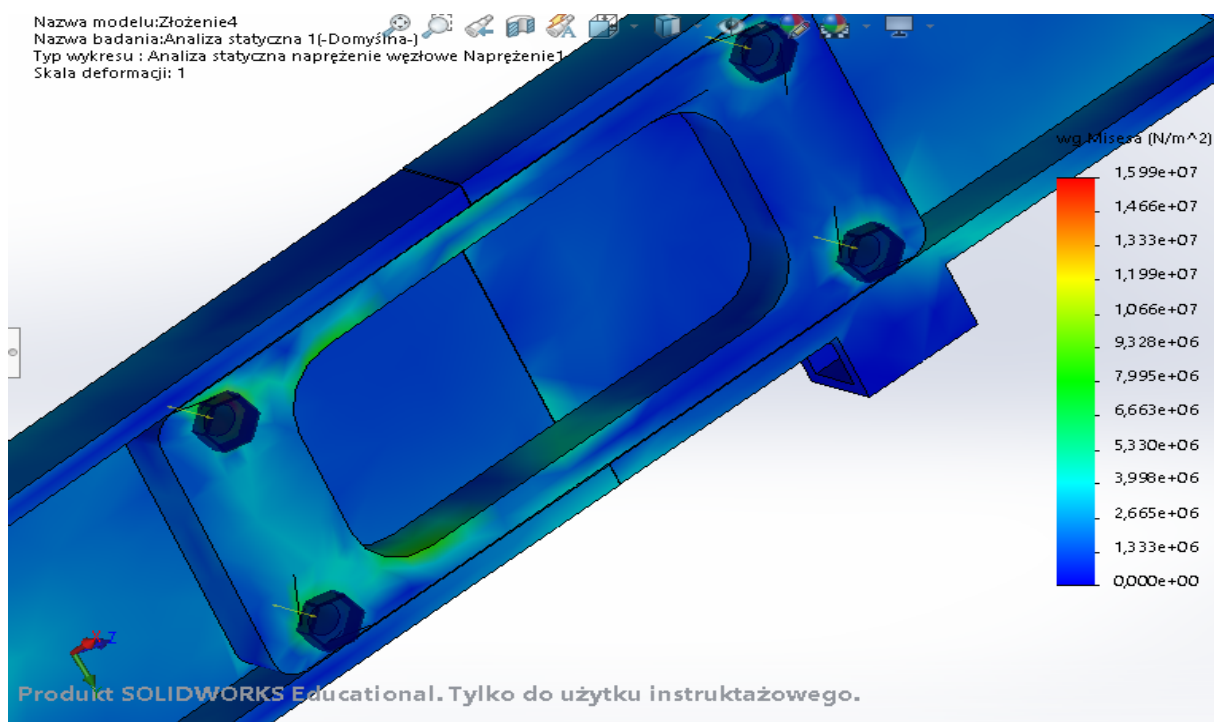
Rysunek 35. Schemat obciążenia ramienia górnego.

Rysunek 35 przedstawia schemat obciążenia ramienia górnego. Unieruchomiona została wewnętrzna ściana walcowa, która jest powierzchnią styku z ramieniem dolnym. Element został obciążony momentem o wartości $1,5 \text{ Nm}$ w miejscu występowania serwomechanizmu odpowiedzialnego za obrót podstawy osprzętu chwytającego, względem osi obrotu ramienia. W celu osiągnięcia warunków równowagi wprowadzono obciążenie momentem o wartości $1,5 \text{ Nm}$ w miejscu powierzchni walcowej położonej na skrajnej prawej stronie. Ponadto zasymulowano obciążenie serwomechanizmem odpowiedzialnego za obrót podstawy osprzętu roboczego momentem o wartości $0,25 \text{ Nm}$. Dodatkowo zostały zdefiniowane kontakty pomiędzy wewnętrznymi ścianami ramienia a płytką oraz napięcia wstępne śrub o wartości 100 N .



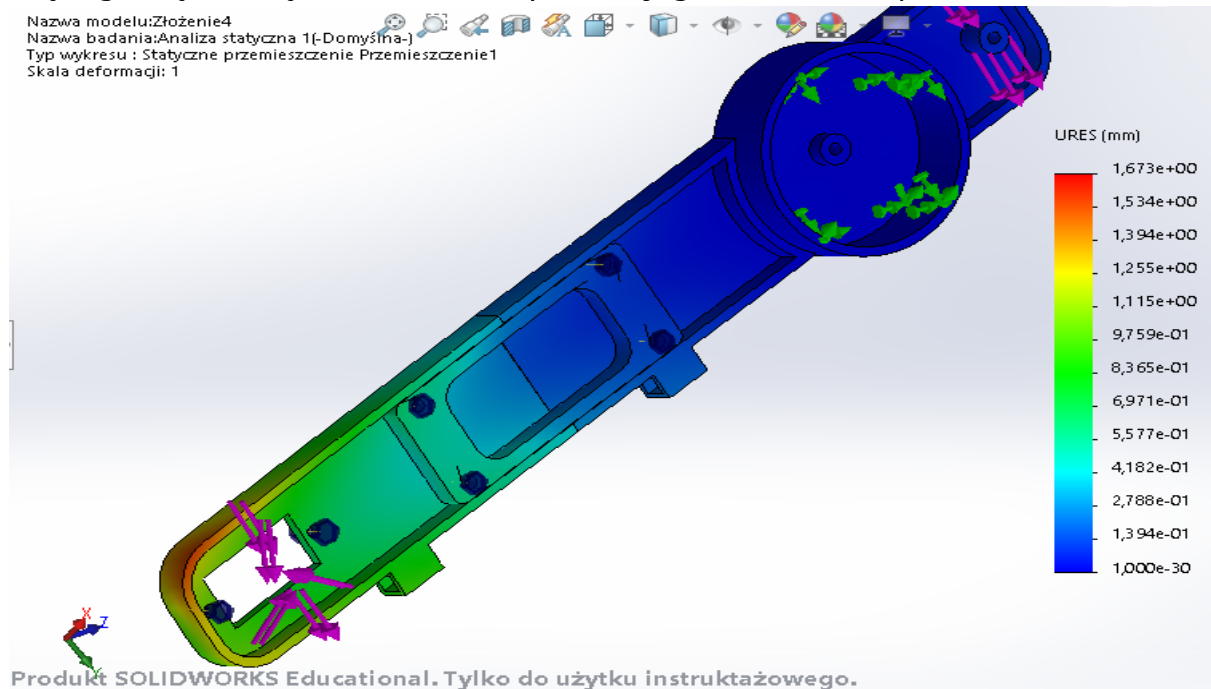
Rysunek 36. Wykres rozkładu naprężeń zredukowanych w ramieniu górnym.

Rysunek 36 przedstawia rozkład naprężeń zredukowanych w ramieniu górnym. Najbardziej obciążonym miejscem jest otwór pod płytkę łączącą. Maksymalne naprężenie zredukowane według hipotezy Hubera-Misesa nie przekracza 16 MPa.



Rysunek 37. Wykres rozkładu naprężeń zredukowanych w płytce.

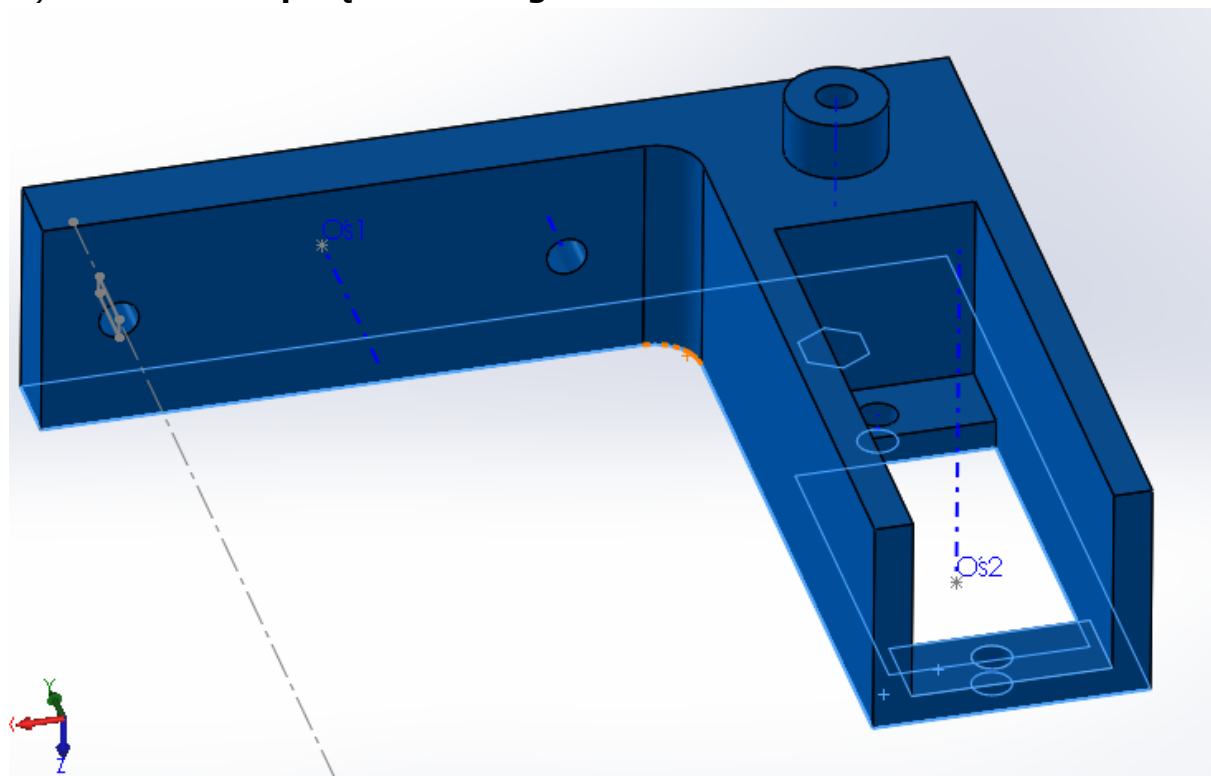
Rysunek 37 przedstawia rozkład naprężeń zredukowanych w płytce łączącej część górną i dolną ramienia. Maksymalna jego wartość nie przekracza 10 MPa.



Rysunek 38. Wykres przemieszczenia ramienia górnego.

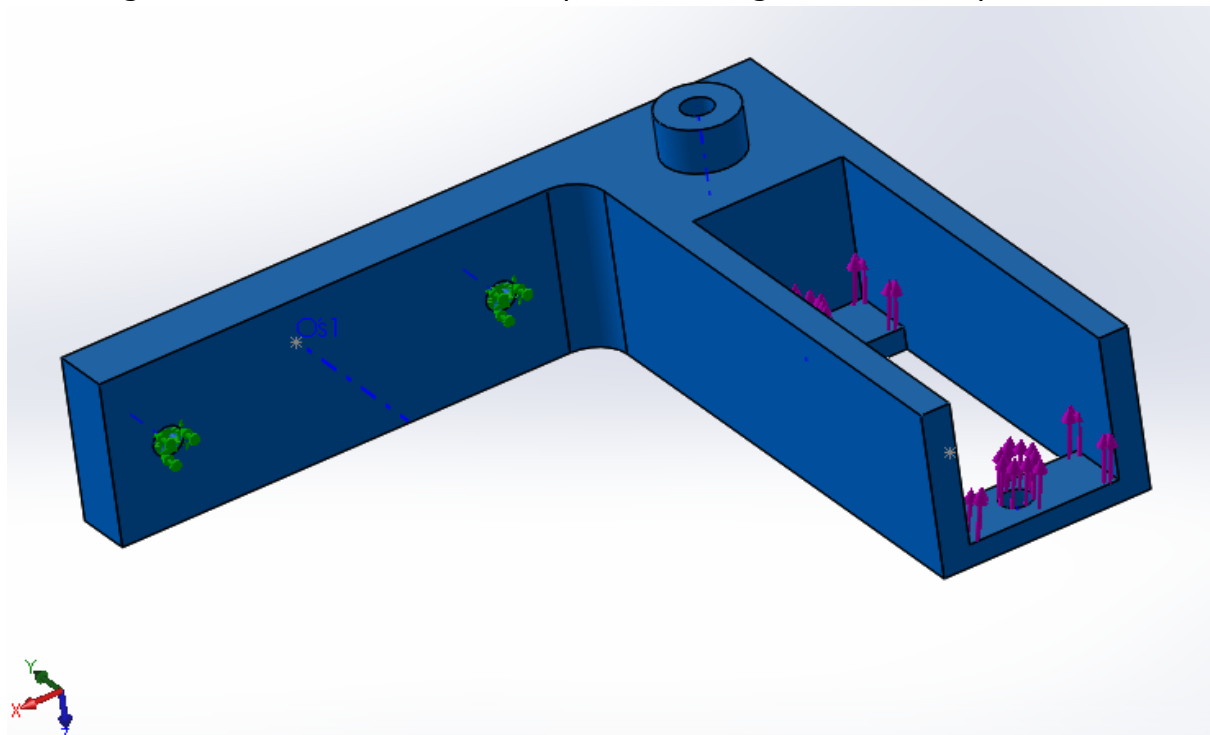
Rysunek 38 przedstawia rozkład przemieszczeń w ramieniu górnym. Maksymalne przemieszczenie wynosi 1,67 mm.

f) Podstawa osprzętu roboczego



Rysunek 39. Model podstawy osprzętu roboczego.

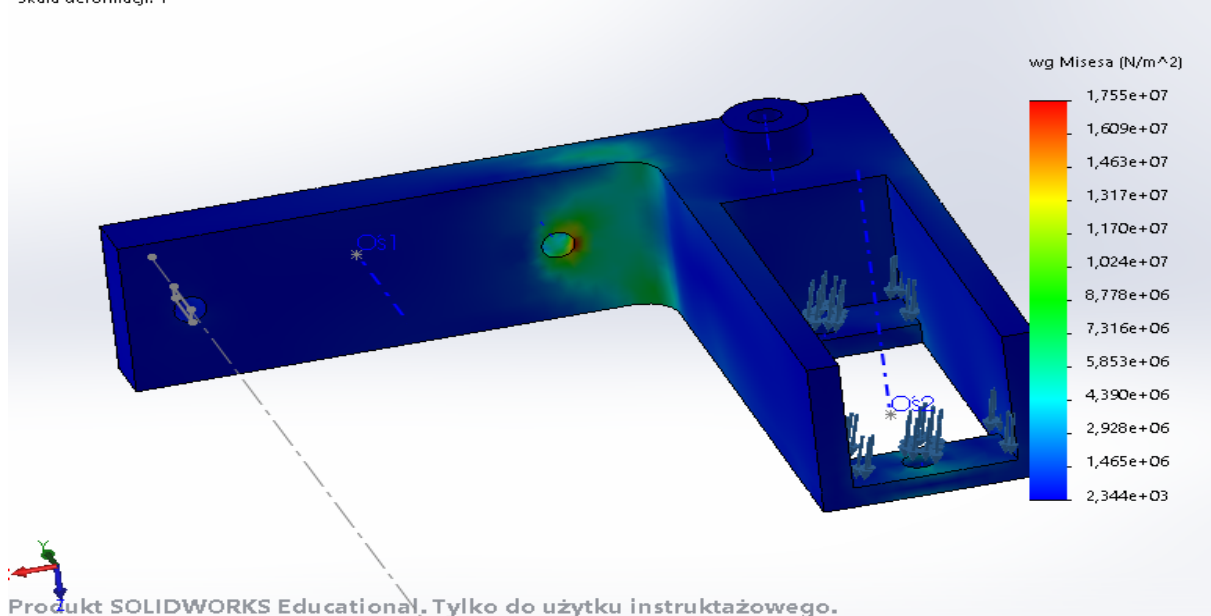
Rysunek 39 przedstawia model podstawy osprzętu roboczego. Przytwierdzony jest za pomocą wkrętów do orczyka serwomechanizmu SG-92 umiejscowionego w ramieniu górnym manipulatora. Zadaniem podstawy jest umocowanie osprzętu roboczego oraz serwomechanizmu odpowiedzialnego za ruch chwytu.



Rysunek 40. Schemat obciążenia podstawy.

Rysunek 40 przedstawia schemat obciążenia podstawy. Element został obciążony momentem o wartości 0,25 Nm, względem osi obrotu podstawy. Otwory mocujące detal do orczyka serwomechanizmu SG-92 zostały unieruchomione.

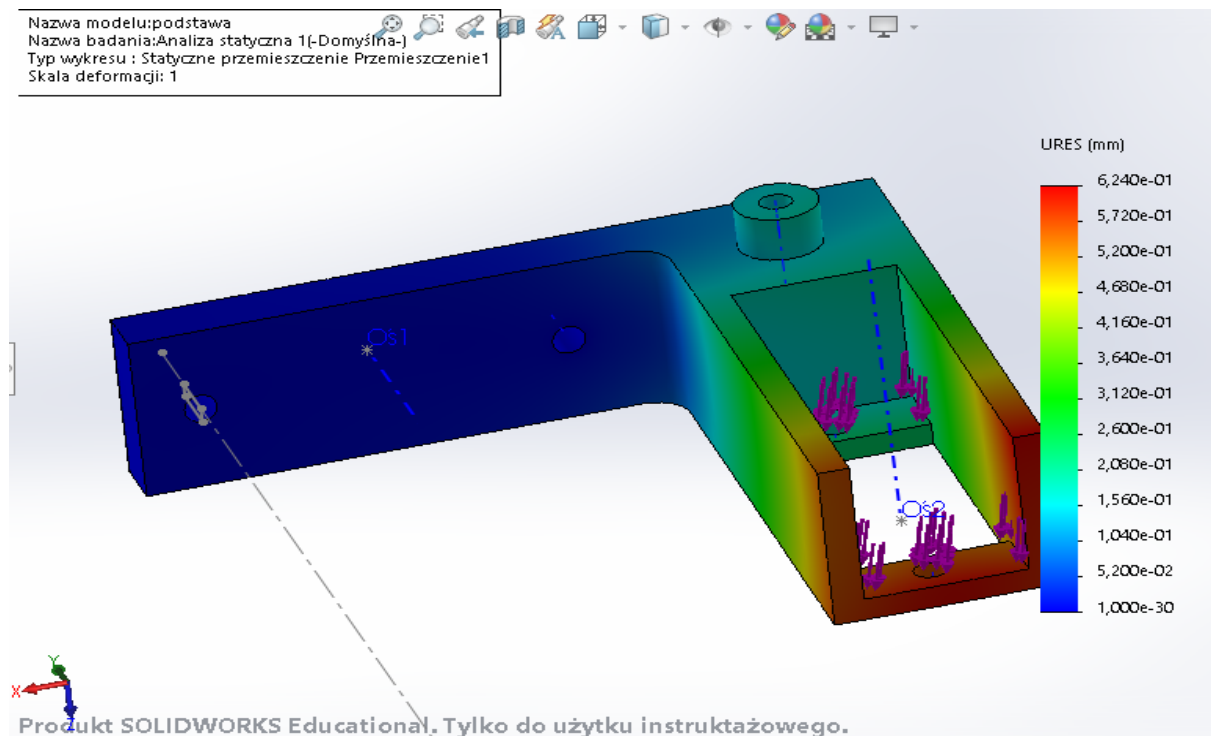
Nazwa modelu: podstawa
 Nazwa badania: Analiza statyczna 1(-Domyślna-)
 Typ wykresu : Analiza statyczna naprężenie węzłowe Naprężenie1
 Skala deformacji: 1



Produkt SOLIDWORKS Educational. Tylko do użytku instruktażowego.

Rysunek 41. Wykres rozkładu naprężeń zredukowanych w podstawie.

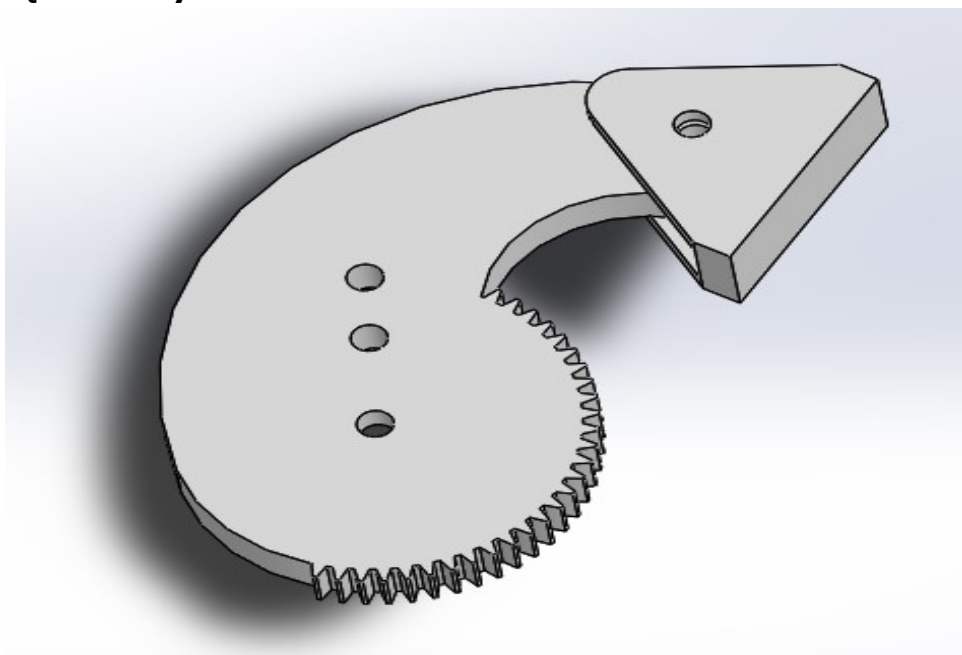
Na rysunku 41 przedstawiono wykres naprężeń zredukowanych w konstrukcji podstawy osprzętu roboczego. Maksymalne naprężenie wynosi 17,55 MPa i jest umiejscowione w otworze mocującym serwomechanizm.



Rysunek 42. Wykres rozkładu przemieszczeń podstawy.

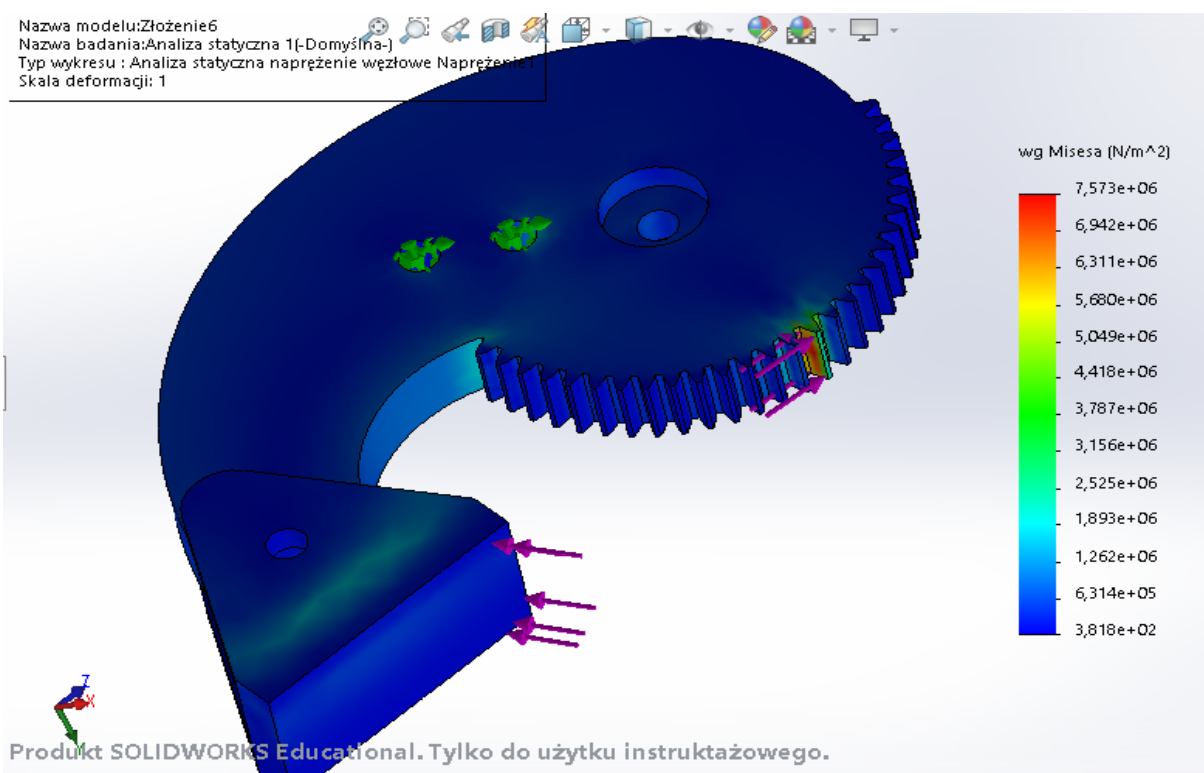
Rysunek 42 przedstawia rozkład przemieszczeń podstawy osprzętu roboczego. Maksymalne ugięcie elementu wynosi 0,62 mm.

g) **Osprzęt roboczy**



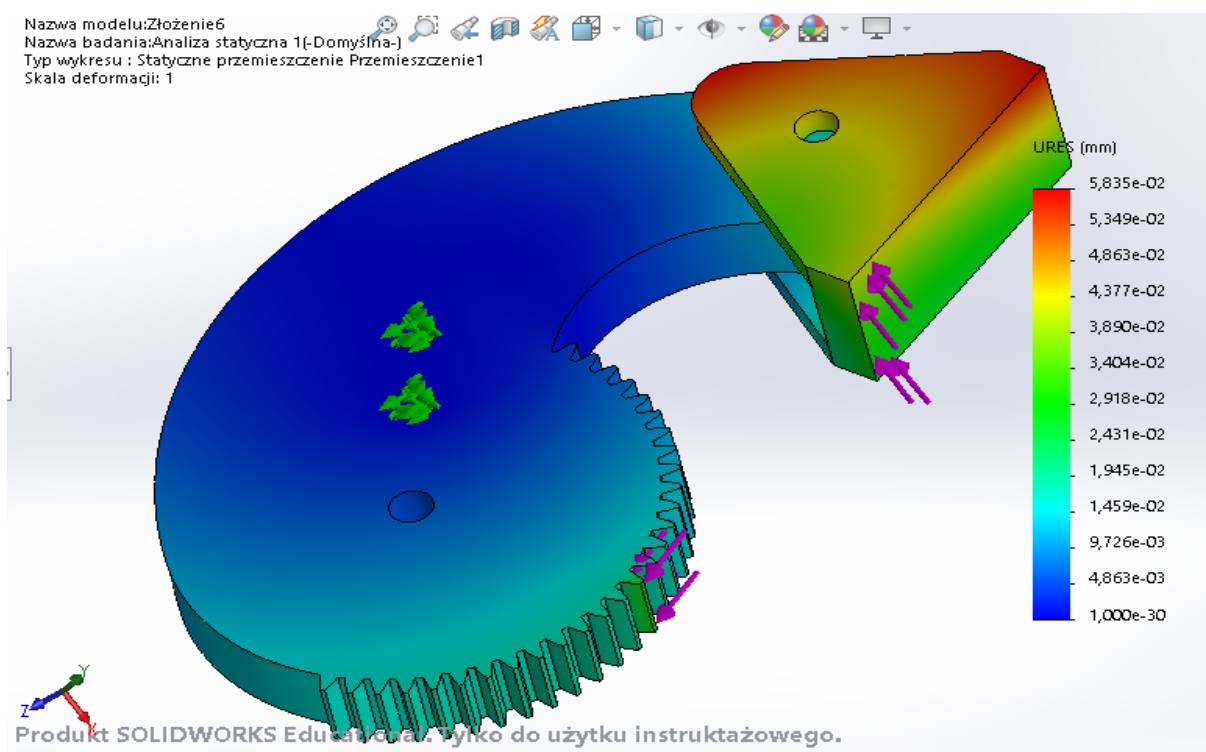
Rysunek 43. Model osprzętu roboczego.

Rysunek 43 przedstawia model osprzętu roboczego. Składa się on z koła zębatego oraz chwytu, dzięki któremu możliwe jest unieruchomienie przenieszonego ładunku. Koło zębate mocowane jest za pomocą wkrętów do orczyka serwomechanizmu SG-92, umiejscowionego w podstawie osprzętu. Koło zębate posiada 28 zębów o module 0,5. Całość została obciążona momentem o wartości 0,25 Nm względem osi obrotu, czyli maksymalną wartością jaką może wytworzyć serwomechanizm. Punktem przyłożenia został ząb oraz powierzchnia styku chwytu.



Rysunek 44. Wykres naprężeń w osprzęcie roboczym.

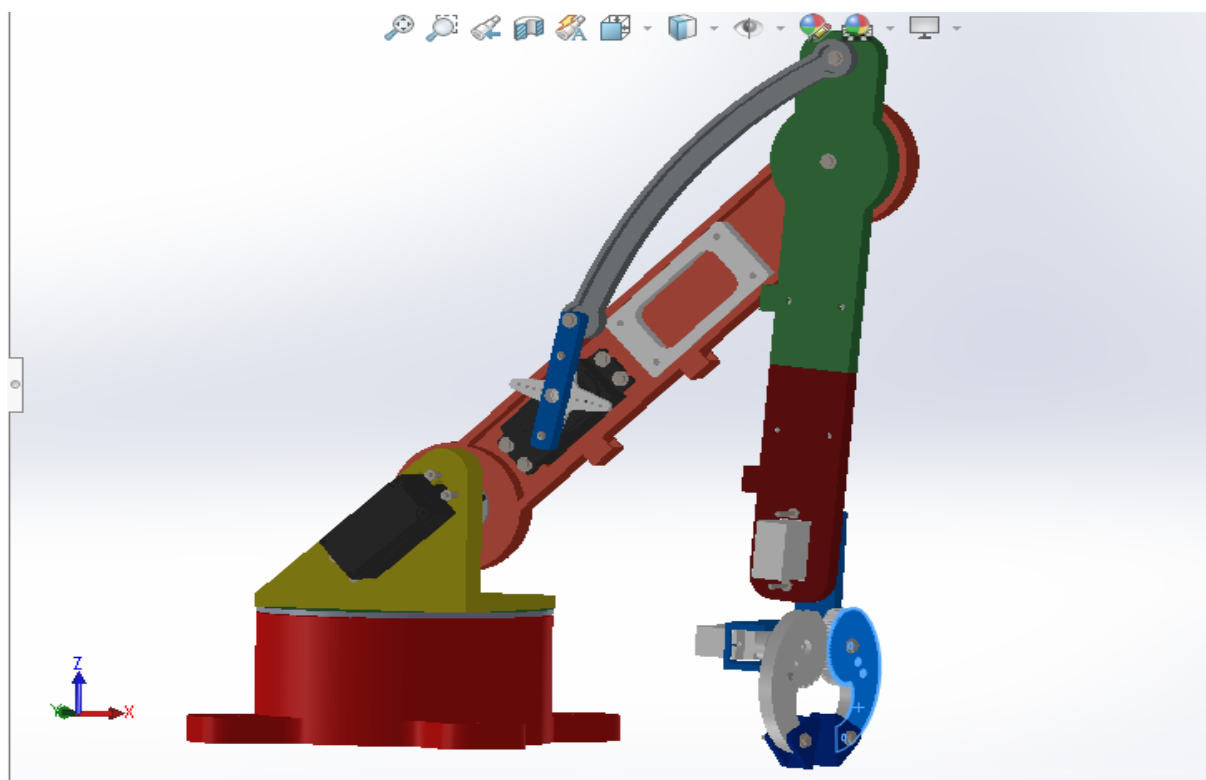
Rysunek 44 przedstawia rozkład naprężeń zredukowanych w osprzęcie roboczym manipulatora. Maksymalne naprężenie wynosi 7,57 MPa, najbardziej obciążonym elementem jest ząb koła zębatego, który współpracuje z drugim kołem.



Rysunek 45. Wykres przemieszczeń osprzętu roboczego.

Rysunek 45 przedstawia wykres przemieszczeń osprzętu roboczego. Maksymalne przemieszczenie występuje na końcu chwytu i wynosi 0,06 mm.

Konstrukcja całego manipulatora została przedstawiona na Rysunku 46.

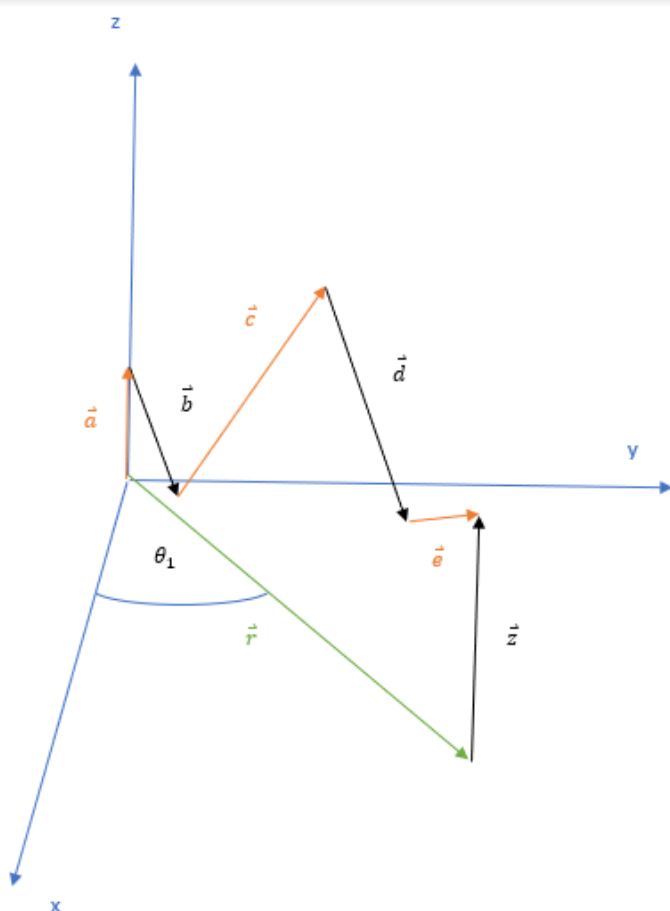


Rysunek 46. Model manipulatora.

4.5 Kinematyka prosta i odwrotna manipulatora

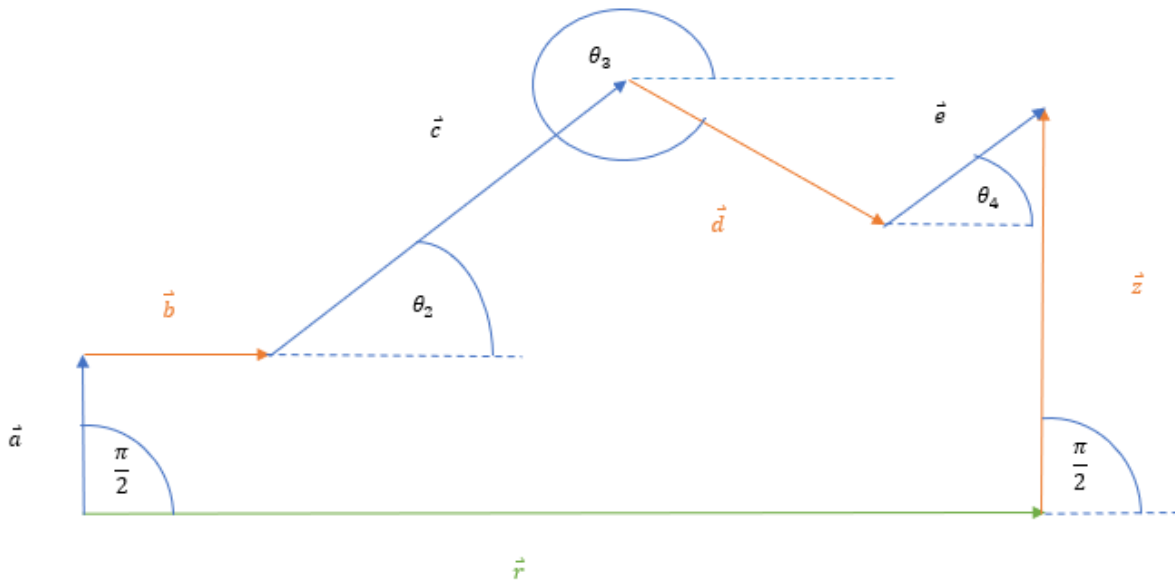
4.5.1 Kinematyka prosta

Zadaniem kinematyki prostej manipulatora jest określenie położenia, prędkości oraz przyspieszeń bez wnikania w przyczynę ruchu. Istnieje kilka metod wyznaczania parametrów ruchu m. in. wprowadzenie notacji Denavita-Hartenberga i rozwiązywanie równań macierzowych lub metoda analityczna. Metoda analityczna polega na wprowadzeniu globalnego układu współrzędnych, następnie zastąpieniu członów mechanizmu układem wektorów, który tworzy zamknięty wielobok. Końcowym etapem jest wprowadzenie równania wektorowego ich sumy, zrzutowaniem go na osie układu współrzędnych i rozwiązanie.



Rysunek 47. Schemat wektorowy manipulatora.

Rysunek 48 przedstawia schemat zamkniętego wieloboku wektorowego manipulatora przegubowego.



Rysunek 48. Schemat wektorowy manipulatora.

Rysunek 49 przedstawia schemat wektorowy manipulatora, który został rzutowany na płaszczyznę przechodzącą przez wszystkie wektory, aby zachować czytelność rysunku. Równanie zamkniętego wieloboku wektorowego można zapisać wzorem (5):

$$\vec{a} + \vec{b} + \vec{c} + \vec{d} + \vec{e} = \vec{z} + \vec{r} \quad (5)$$

Równanie (5) to można rzutować na osie przechodzące przez wektory \vec{r} oraz \vec{z} :

$$x1: a \cdot \cos\left(\frac{\pi}{2}\right) + b \cdot \cos(0) + c \cdot \cos(\theta_2) + d \cdot \cos(\theta_3) + e \cdot \cos(\theta_4) = r \cdot \cos(0) + z \cdot \cos\left(\frac{\pi}{2}\right) \quad (6)$$

$$y1: a \cdot \sin\left(\frac{\pi}{2}\right) + b \cdot \sin(0) + c \cdot \sin(\theta_2) + d \cdot \sin(\theta_3) + e \cdot \sin(\theta_4) = r \cdot \sin(0) + z \cdot \sin\left(\frac{\pi}{2}\right) \quad (7)$$

Po uproszczeniu (6) i (7) otrzymujemy (8) oraz (9):

$$x1: b + c \cdot \cos(\theta_2) + d \cdot \cos(\theta_3) + e \cdot \cos(\theta_4) = r \quad (8)$$

$$y1: a + c \cdot \sin(\theta_2) + d \cdot \sin(\theta_3) + e \cdot \sin(\theta_4) = z \quad (9)$$

Znając długości odcinków a, b, c, d, e oraz kąty $\theta_2, \theta_3, \theta_4$ oraz θ_1 można określić położenie końcówki osprzętu roboczego (10), (11), (12).

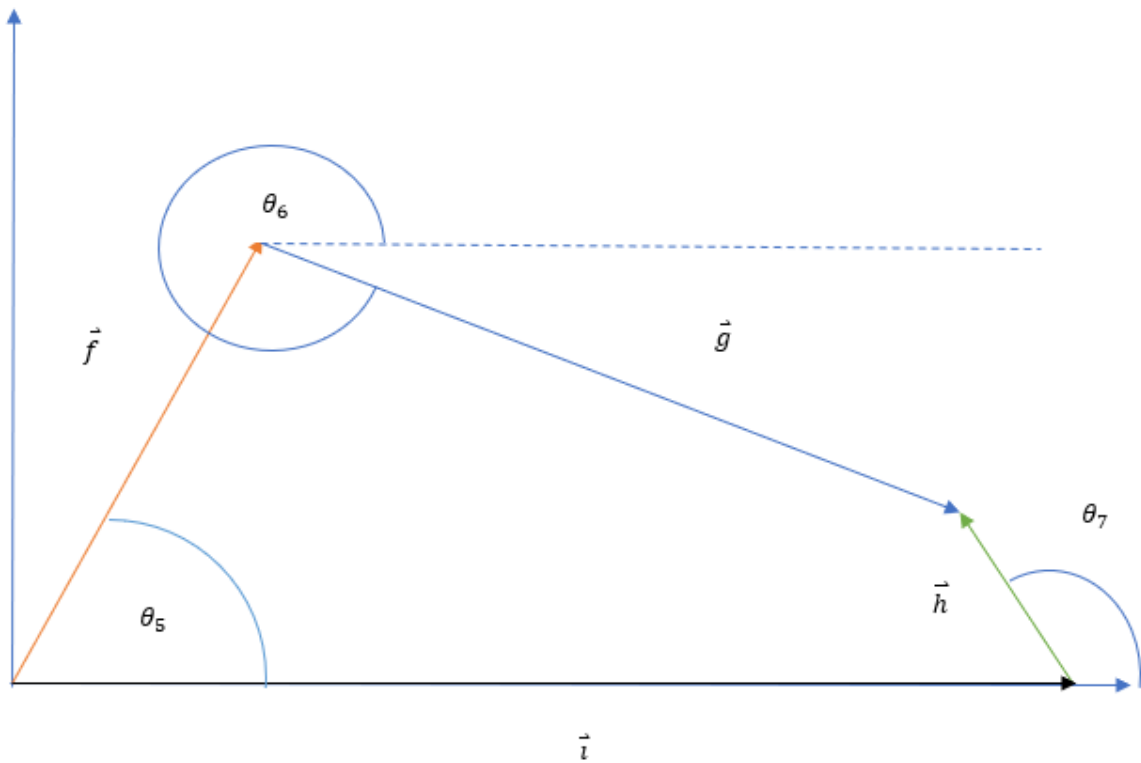
$$x = (b + c \cdot \cos(\theta_2) + d \cdot \cos(\theta_3) + e \cdot \cos(\theta_4)) \cdot \cos(\theta_1) \quad (10)$$

$$y = (b + c \cdot \cos(\theta_2) + d \cdot \cos(\theta_3) + e \cdot \cos(\theta_4)) \cdot \sin(\theta_1) \quad (11)$$

$$z = a + c \cdot \sin(\theta_2) + d \cdot \sin(\theta_3) + e \cdot \sin(\theta_4) \quad (12)$$

Parametry a, b, c, d, e są stałe, zaś kąty $\theta_2, \theta_3, \theta_4, \theta_1$ oraz r i z są niewiadomymi w funkcji czasu.

Kąty $\theta_1, \theta_2, \theta_4$ są określone w sposób jawny, ponieważ są to kąty obrotu serwomechanizmów, zaś kąt θ_3 nie jest, ponieważ do obrotu ramienia górnego zastosowano mechanizm dźwigniowy.



Rysunek 49. Schemat wektorowy mechanizmu dźwigniowego.

Rysunek 49 przedstawia schemat wektorowy mechanizmu dźwigniowego, który służy do napędzania ramienia górnego. Równanie wektorowe zamkniętego wieloboku można zapisać wzorem (13):

$$\vec{f} + \vec{g} = \vec{h} + \vec{i} \quad (13)$$

Równanie (13) można rzutować na osie przechodzące przez wektory \vec{r} oraz \vec{z} :

$$x2: f \cdot \cos(\theta_5) + g \cdot \cos(\theta_6) = h \cdot \cos(\theta_7) + i \cdot \cos(0) \quad (14)$$

$$y2: f \cdot \sin(\theta_5) + g \cdot \sin(\theta_6) = h \cdot \sin(\theta_7) + i \cdot \sin(0) \quad (15)$$

Po uproszczeniu (14) i (15) otrzymujemy (16) oraz (17):

$$x2: f \cdot \cos(\theta_5) + g \cdot \cos(\theta_6) = h \cdot \cos(\theta_7) + i \quad (16)$$

$$y2: f \cdot \sin(\theta_5) + g \cdot \sin(\theta_6) = h \cdot \sin(\theta_7) \quad (17)$$

Parametry e, f, g, h, i są stałe, zaś kąty $\theta_5, \theta_6, \theta_7$ są niewiadomymi w funkcji czasu.

Zadaniem jest określenie kąta θ_7 w zależności od kąta θ_5 , który jest kątem obrotu serwomechanizmu. Można jest rozwiązać na dwa sposoby: analitycznie oraz numerycznie. Metoda analityczna polega na określeniu związku pomiędzy kątami za pomocą funkcji. Wymaga ona wielu przekształceń geometrycznych wykorzystując tożsamości trygonometryczne. Metoda ta jest dość skomplikowana do zaimplementowania na mikrokontrolerze. Metoda numeryczna jest metodą

iteracyjną, polegającą na obliczaniu kolejnych przybliżeń. Zbieżność wyników jest uzależniona od algorytmu oraz warunków początkowych. Najprostszą z nich jest algorytm wykorzystujący metodę Newtona oraz macierz Jakobiego. Jakobian analityczny definiuje się jako macierz pochodnych cząstkowych kinematyki manipulatora (18).

$$J = \frac{\partial w(\theta)}{\partial \theta} \quad (18)$$

Wzór algorytmu ma postać (19):

$$q(k+1) = q(k) + J(q)^{-1} \cdot (y_d - k(q)) \quad (19)$$

Gdzie:

$q(k+1), q(k)$ – wartość zmiennej w chwili następnej i poprzedniej

$J(q)$ – aktualna macierz Jakobiego

$k(q)$ – położenie chwytaka przy obecnych współrzędnych

y_d – położenie docelowe

Funkcję w można zapisać (20):

$$w = \begin{cases} f \cdot \cos(\theta_5) + g \cdot \cos(\theta_6) - h \cdot \cos(\theta_7) - i = 0 \\ f \cdot \sin(\theta_5) + g \cdot \sin(\theta_6) - h \cdot \sin(\theta_7) = 0 \end{cases} \quad (20)$$

Zmiennymi są kąty θ_6 i θ_7 , zaś θ_5 jest znane. Macierz Jakobiego wynosi (21):

$$\begin{bmatrix} -g \cdot \sin(\theta_6) & h \cdot \sin(\theta_7) \\ g \cdot \cos(\theta_6) & -h \cdot \cos(\theta_7) \end{bmatrix} \quad (21)$$

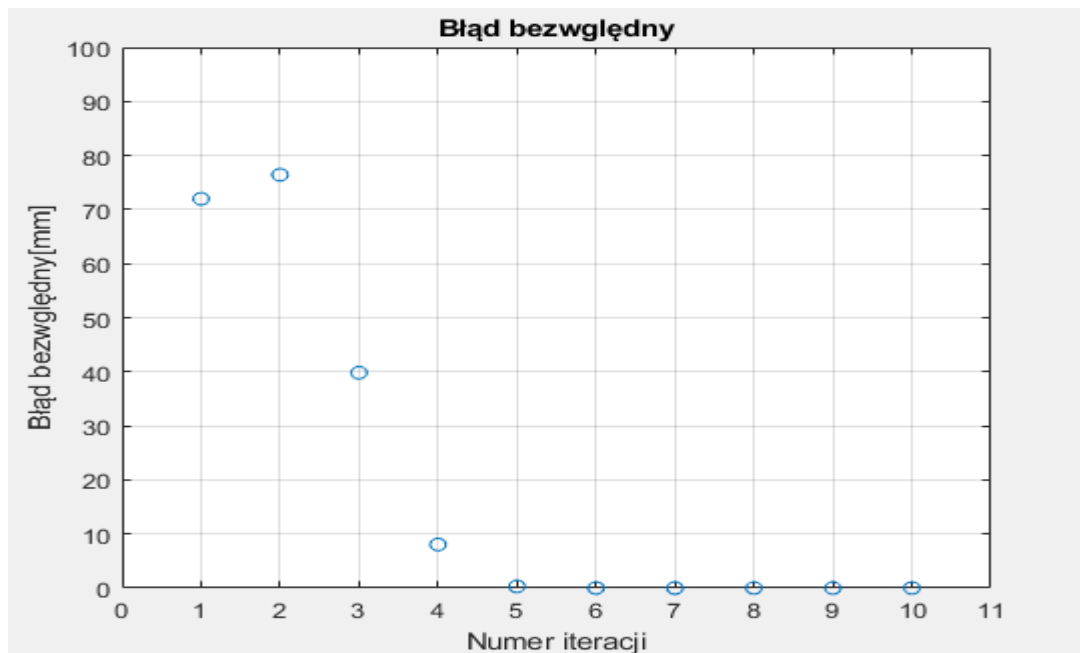
Algorytm obliczania kątów został napisany w programie Matlab: (rys. 50)

```
e=78
f=30
g=140
h=40
i=[135;0]
theta5=0
theta6=0.5
theta7=1
theta=[theta6;theta7]

for z=1:1:10
    uchyb=i-[f*cos(theta5)+g*cos(theta(1))-h*cos(theta(2)) ; f*sin(theta5)+g*sin(theta(1))-h*sin(theta(2))]
    J=[-g*sin(theta(1)),h*sin(theta(2));g*cos(theta(1)), -h*cos(theta(2))]
    theta=theta+inv(J)*uchyb
    blad(z)=norm(uchyb)
    f*sin(theta5)+g*sin(theta(1))-h*sin(theta(2))
end

plot(blad)
title("Błąd bezwzględny")
xlabel("Numer iteracji")
ylabel("Błąd bezwzględny[mm]")
grid on
```

Rysunek 50. Skrypt Matlaba do obliczania kinematyki mechanizmu dźwigniowego.



Rysunek 51. Wykres błędu bezwzględnego położenia.

Rysunek 51 przedstawia błąd bezwzględny położenia końca wektora \vec{i} . Błąd bezwzględny przy założonych warunkach początkowych na początku iteracji wynosi 70 mm. Po piątej serii obliczeń uchyb wynosi niż 0,3 mm. Przedstawiony algorytm jest bardzo skuteczny oraz szybki, co sprawia że idealnie nadaje się do zaimplementowania na mikrokontrolerze.

Przestrzeń robocza manipulatora została wyznaczona za pomocą skryptu Matlaba przedstawionego na rysunku 52.

```
theta2=0
gamma=0.591:0.01:2.3446

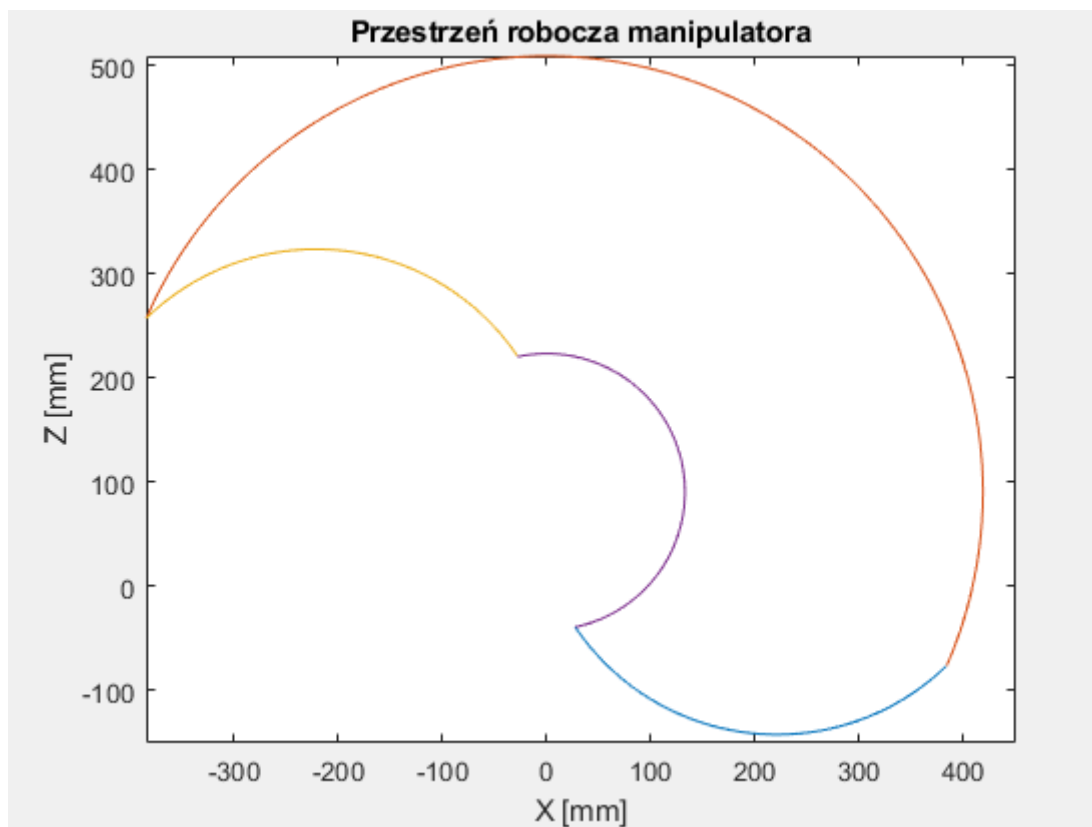
x=a*cos(pi/2)+b*cos(theta2)+c*cos(theta2)+d*cos(theta2+gamma+pi)+e*cos(theta2+gamma+pi)
y=a*sin(pi/2)+b*sin(theta2)+c*sin(theta2)+d*sin(theta2+gamma+pi)+e*sin(theta2+gamma+pi)
plot(x(1,:),y(1,:))
hold on

theta2=0:0.01:pi
gamma=2.3446
x=a*cos(pi/2)+b*cos(theta2)+c*cos(theta2)+d*cos(theta2+gamma+pi)+e*cos(theta2+gamma+pi)
y=a*sin(pi/2)+b*sin(theta2)+c*sin(theta2)+d*sin(theta2+gamma+pi)+e*sin(theta2+gamma+pi)
plot(x(1,:),y(1,:))

hold on
theta2=pi
gamma=2.3446:-0.01:0.591
x=a*cos(pi/2)+b*cos(theta2)+c*cos(theta2)+d*cos(theta2+gamma+pi)+e*cos(theta2+gamma+pi)
y=a*sin(pi/2)+b*sin(theta2)+c*sin(theta2)+d*sin(theta2+gamma+pi)+e*sin(theta2+gamma+pi)
plot(x(1,:),y(1,:))

hold on
theta2=pi:-0.01:0
gamma=0.591
x=a*cos(pi/2)+b*cos(theta2)+c*cos(theta2)+d*cos(theta2+gamma+pi)+e*cos(theta2+gamma+pi)
y=a*sin(pi/2)+b*sin(theta2)+c*sin(theta2)+d*sin(theta2+gamma+pi)+e*sin(theta2+gamma+pi)
plot(x(1,:),y(1,:))
title("Przestrzeń robocza manipulatora")
xlabel("X [mm]")
ylabel("Y [mm]")
```

Rysunek 52. Skrypt Matlaba określającą przestrzeń roboczą manipulatora.



Rysunek 53. Wykres przestrzeni roboczej manipulatora.

Rysunek 53 przedstawia wykres trajektorii ruchu końca osprzętu roboczego manipulatora, który został wyznaczony za pomocą obrotów ramienia dolnego oraz górnego. Niebieska linia została wyznaczona poprzez ruch górnego ramienia, przy kącie położenia ramienia dolnego wynoszącym zero. Czerwona poprzez ruch dolnego ramienia, przy maksymalnym wychyleniu ramienia górnego. Żółta poprzez ruch powrotny ramienia górnego, zaś fioletowa dolnego.

4.5.2 Kinematyka odwrotna manipulatora

Zadaniem odwrotnym kinematyki jest określenie współrzędnych konfiguracyjnych w celu umieszczenia osprzętu roboczego w wymaganym położeniu względem globalnego układu współrzędnych. W tym przypadku współrzędnymi konfiguracyjnymi są położenia kątowe serwo mechanizmów.

Strategie rozwiązywania kinematyki można podzielić na dwie klasy: rozwiązania w postaci jawnej oraz rozwiązania numeryczne. Postać jawna rozwiązania wymaga analitycznych wyrażeń, których wyznaczenie oraz implementacja mogą sprawiać trudności. Metoda numeryczna jest metodą iteracyjną, polegającą na kolejnych przybliżeniach. Tak jak w rozdziale 4.5.1 wykorzystano algorytm Newtona z wykorzystaniem macierzy Jakobiego.

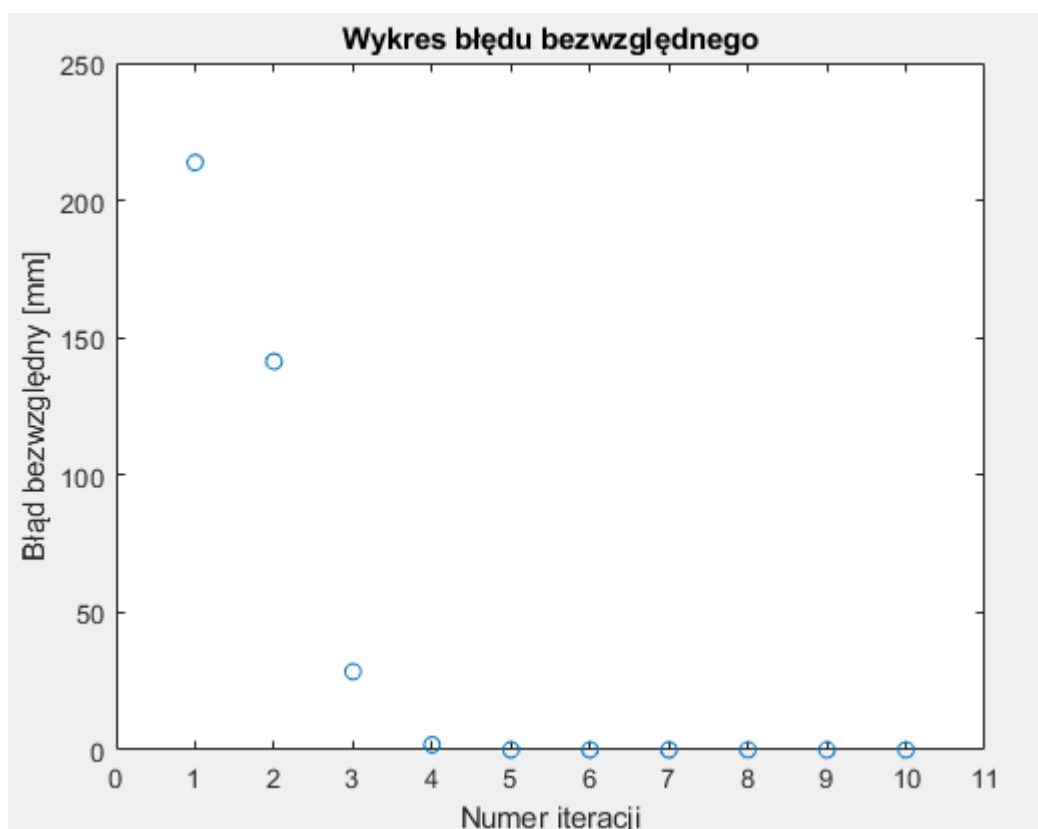
```
clear all
close all
clc
a=91;
b=21
c=200
d=155
e=78
f=30
g=140
h=40
i=[135;0]
theta5=1
theta6=0
theta7=0.5
kat=[theta5;theta6];
theta=[0.5;6;0.1]
Z=[150;250;320];
Y=[sqrt(Z(1)^2+Z(2)^2);Z(3)]
theta1=atan(Z(2)/Z(1));
for z=1:1:10
X=[(b+c*cos(theta(1)))+d*cos(theta(2))+e*cos(theta(3));
a+c*sin(theta(1))+d*sin(theta(2))+e*sin(theta(3))]
uchyb=Y-X
J=[(-c*sin(theta(1))), d*(-sin(theta(2))), e*(-sin(theta(3)));
c*cos(theta(1)), d*cos(theta(2)), e*cos(theta(3))]
theta=theta+J'*inv(J*J')*uchyb
blad(z)=norm(uchyb)
end
figure
plot(blad)
title("Wykres błędu bezwzględnego")
xlabel("Numer iteracji")
ylabel("Błąd bezwzględny [mm]")
gamma=theta(2)-theta(1)-pi
for z=1:1:10
uchyb_1=i-[f*cos(kat(1))+g*cos(kat(2))-h*cos(gamma) ;
f*sin(kat(1))+g*sin(kat(2))-h*sin(gamma)]
J=[-f*sin(kat(1)), -g*sin(kat(2)); f*cos(kat(1)), g*cos(kat(2))]
kat=kat+inv(J)*uchyb_1
end
```

Rysunek 54. Skrypt Matlaba obliczający kinematykę odwrotną.

Rysunek 54 przedstawia skrypt Matlaba do obliczenia zadania odwrotnego kinematyki. Pierwsza pętla for służy do określenia kątów θ_2 , θ_3 oraz θ_4 . Macierz Jakobiego nie jest macierzą kwadratową, więc nie można określić macierzy odwrotnej. W tym przypadku należy odpowiednio przekształcić macierz za pomocą wzoru (22):

$$J^\#(q) = J^T(q) \cdot [J(q) \cdot J^T(q)]^{-1} \quad (22)$$

Kolejna pętla ma za zadanie obliczyć kąt θ_5 , który jest kątem obrotu serwomechanizmu mechanizmu dźwigniowego odpowiedzialnego za obrót ramienia górnego.



Rysunek 55. Wykres błędu bezwzględnego.

Rysunek 55 przedstawia wynik błędu położenia osprzętu roboczego w kolejnych iteracjach. Po piątej iteracji uchyb wynosi poniżej 0,01 mm. Szybkość zbieżności algorytmu jak i jego rozwiązania ściśle zależą od początkowych wartości kątów. Przy innych wartościach początkowych można uzyskać różne rozwiązania, które również będą poprawne, ponieważ w pewnych przypadkach istnieje kilka rozwiązań.

4.6 Sterowanie

Połączenie z chmurą danych oraz sterowanie pozycją silników zostało zrealizowane za pomocą płytki NodeMCU V2. Rysunek 56a i 56cd przedstawia szkic programu Arduino IDE odpowiedzialne za wspomniane wyżej funkcje.

```
#include <ThingyESP8266.h> // biblioteka odpowiedzialna za komunikację z chmurą
#include <Servo.h> // biblioteka odpowiedzialna za sterowanie serwomechanizmami
#include <BasicLinearAlgebra.h> // biblioteka do obliczeń macierzowych
using namespace BLA;
#define USERNAME "WojciechWysokinski" // nazwa chmury
#define DEVICE_ID "Praca_Dyplomowa" // nazwa panelu interfejsu
#define DEVICE_CREDENTIAL "Praca_Dyplomowa" // hasło dostępu do chmury

#define SSID "Praca_Przejsciowa" // nazwa punktu dostępu Wi-Fi
#define SSID_PASSWORD "123456789" // hasło do punktu dostępu
Servo servo1; // obiekt związany z obsługą serwomechanizmu
Servo servo2;
Servo servo3;
Servo servo4;
Servo servo5;
int a = 91; // wysokość mocowania pierwszego ramienia obrotowego [mm]
int b = 21; // odległość osi obrotu ramienia pierwszego od osi obrotu podstawy [mm]
int c = 200; // długość ramienia pierwszego
int d = 155; // długość ramienia drugiego
int e = 78; // odległość końca osprzętu od osi obrotu
int f = 30; //długość orczyka serwomechanizmu obrotu ramienia górnego
int g = 140; // długość dźwigni mechanizmu obrotu ramienia górnego
int h = 40; // odległość osi obrotu ramienia górnego do mocowania dźwigni
int i = 135; //odległość osi obrotu serwomechanizmu ramienia górnego do osi obrotu ramienia górnego
int theta1 = 90, theta2 = 90, theta3 = 0, theta4 = 90, theta5 = 45; // początkowe wartości kątów [deg]
int X = 150, Y = 250, Z = 320; // początkowa wartość położenia osprzętu roboczego w sterowaniu automatycznym [mm]
boolean Ster; // zmienna odpowiedzialna za sterowanie automatyczne
ThingyESP8266 thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);
void setup() // funkcja wykonana tylko raz po uruchomieniu
{
  thing.add_wifi(SSID, SSID_PASSWORD); // inicjalizacja połączenia z punktem dostępu
  servo1.attach(16); // ustawienie pinu 16 jako wyjście PWM
  servo2.attach(5);
  servo3.attach(4);
  servo4.attach(2);
  servo5.attach(0);
  thing["theta1"] << inputValue(theta1); // odbiór zmiennej theta1 z chmury
  thing["theta2"] << inputValue(theta2);
  thing["theta3"] << inputValue(theta3);
  thing["theta4"] << inputValue(theta4);
  thing["theta5"] << inputValue(theta5);
  thing["Ster"] << inputValue(Ster);
  thing["X"] << inputValue(X);
  thing["Y"] << inputValue(Y);
  thing["Z"] << inputValue(Z);
}
void loop() // funkcja wykonywana ciągle
{
  thing.handle();
  sterowanie(); // funkcja związana ze sterowaniem serwomechanizmami
}
```

Rysunek 56a. Szkic Arduino IDE.

```

    void sterowanie()
{
    if (Ster == 0) {
        servo1.write(theta1); // ustawienie katowe serwomechanizmu
        servo2.write(abs(theta2 - 180));
        servo3.write(theta3);
        servo4.write(abs(theta4 - 180));
        servo5.write(theta5);
    }
    else if (Ster == 1) // funkcja związana ze sterowaniem automatycznym
    {
        BLA::Matrix<3> theta = {0.5, 6, 0.1};
        BLA::Matrix<2> zad1 = {sqrt(X * X + Y * Y), Z};
        for (int z = 0; z < 10; z++) {
            BLA::Matrix<2, 3>
            J1 = { -c * sin(theta(0)), -d * sin(theta(1)), -e * sin(theta(2)),
                  c * cos(theta(0)), d * cos(theta(1)), e * cos(theta(2))
            }; // Macierz Jakobiego
            BLA::Matrix<2>
            X1 = {b + c * cos(theta(0)) + d * cos(theta(1)) + e * cos(theta(2)),
                  a + c * sin(theta(0)) + d * sin(theta(1)) + e * sin(theta(2))
            };
            BLA::Matrix<2> uchyb1 = {zad1 - X1};
            BLA::Matrix<3, 2> Jt = ~J1; // macierz transponowana
            BLA::Matrix<2, 2> Z = (J1 * Jt);
            BLA::Matrix<2, 2> P = Invert(Z);
            BLA::Matrix<3, 2> J_z = Jt * P;
            theta = theta + J_z * uchyb1; // kolejne przybliżenie katów
        }
        double gamma = theta(1) - theta(0) - 3.14;
        BLA::Matrix<2> kat = {1, 0};
        for (int z = 0; z < 10; z++) {
            BLA::Matrix<2>
            X2 = {f * cos(kat(0)) + g * cos(kat(1)) - h * cos(gamma),
                  f * sin(kat(0)) + g * sin(kat(1)) - h * sin(gamma)
            };
            BLA::Matrix<2, 2> J2 = { -f * sin(kat(0)), -g * sin(kat(1)),
                                      f * cos(kat(0)), g * cos(kat(1))
            };
            BLA::Matrix<2> zad2 = {i, 0};
            BLA::Matrix<2> uchyb2 = zad2 - X2;
            kat = kat + Invert(J2) * uchyb2;
        }
        theta1 = round(atan2(Y, X) * 180 / PI);
        theta2 = round(theta(0) * 180 / PI);
        theta3 = round(kat(0) * 180 / PI);
        theta4 = round((PI / 2 + 2 * PI - theta(1) + theta(2)) * 180 / PI);
        if (theta(0) <= PI && theta(1) <= 2 * PI && theta(2) <= PI ) {
            servo1.write(theta1); // ustawienie katowe serwomechanizmu
            servo2.write(abs(theta2 - 180));
            servo3.write(theta3);
            servo4.write(abs(theta4 - 180));
            servo5.write(theta5);
        }
    }
    else {}
}

```

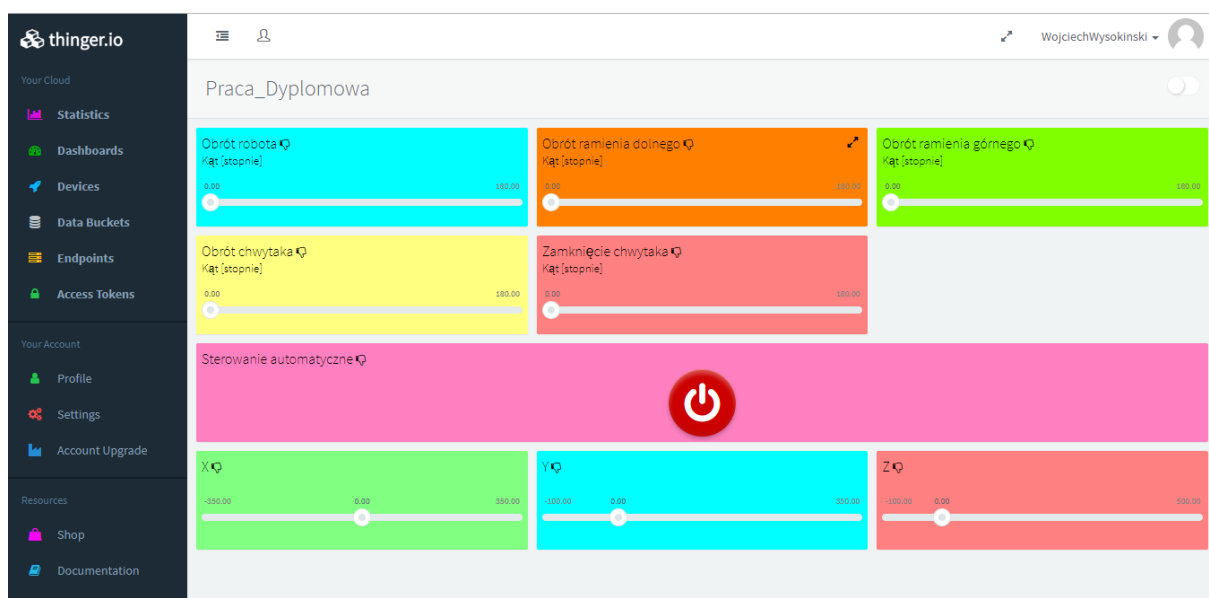
Rysunek 56cd. Szkic Arduino IDE.

Biblioteka ThingerESP8266.h jest odpowiedzialna za łączność pomiędzy modulem Wi-Fi a chmurą Thinger.io. Biblioteka Servo.h ma za zadanie sterować pracą silników poprzez wysyłanie sygnałów PWM o zmiennym wypełnieniu. BasicLinearAlgebra.h jest biblioteką odpowiedzialną za obliczenia macierzowe, głównie kinematyki odwrotnej. Zmienne USERNAME, DEVICE_ID, DEVICE_CREDENTIALS, SSID oraz SSID_PASSWORD służą do określenia nazwy użytkownika chmury, hasła do niej, nazwy sieci Wi-Fi, z którą NodeMCU ma się połączyć oraz hasła do niej.

Obiekty servo1, servo2, servo3, servo4 i servo5 definiują nazwy serwomechanizmów a funkcja ThingerESP8266 thing uruchamia transmisję danych pomiędzy chmurą. W funkcji setup zdefiniowano funkcję odpowiedzialną za połączenie się z siecią Wi-Fi, piny które będą sterowały silnikami: 16, 5, 4, 2, 0 oraz nazwy zmiennych do których będą przypisane wartości wysłane z chmury.

W pętli loop wywoływana jest funkcja thing.handle odpowiedzialna za ciągłe przesyłanie danych oraz funkcja sterująca pracą silników. W funkcji tej sprawdzany jest warunek czy zmienna odpowiedzialna za sterowanie automatyczne wynosi 0 lub 1. Dla Ster==0 manipulator pracuje w trybie pracy ręcznej tzn. operator wprowadza położenia kątowe wszystkich 5 serwomechanizmów. Jeżeli Ster==1 to obliczane są położenia kątowe silników z wykorzystaniem algorytmu Newtona na podstawie wprowadzonych przez operatora położenia końca osprzętu roboczego we współrzędnych X, Y, Z.

Jeżeli operator wprowadzi dane dla których rozwiązanie zadania kinematyki odwrotnej nie jest możliwe, mikrokontroler zapamiętuje poprzednie prawidłowe dane.

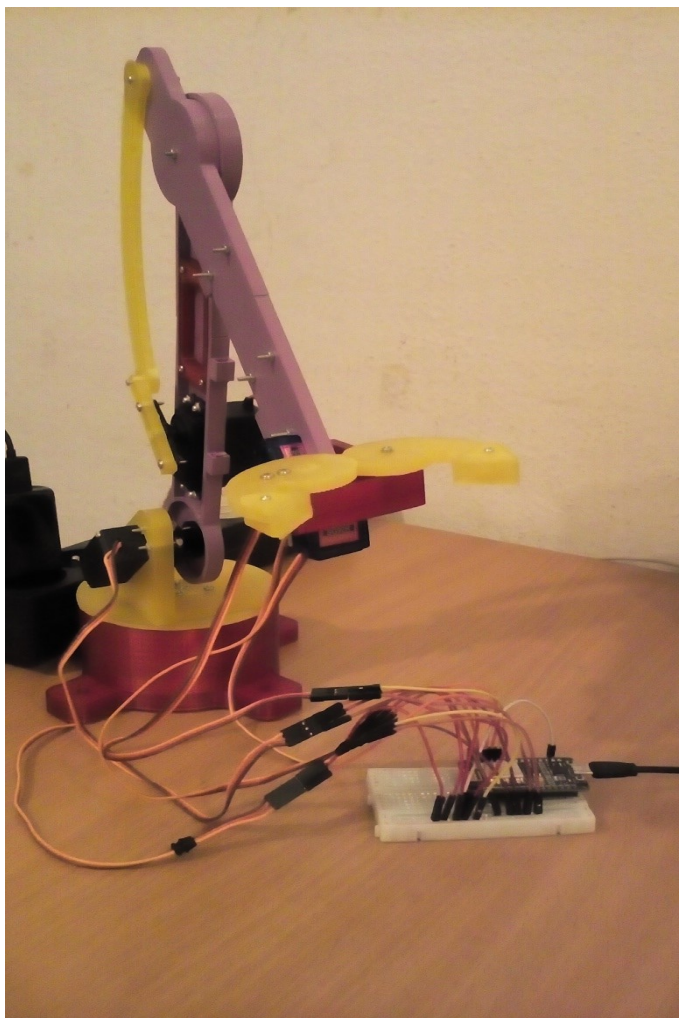


Rysunek 57. Panel operatora.

Rysunek 57 przedstawia panel operatora w chmurze danych Thinger.io. Zdefiniowano tam 5 suwaków odpowiedzialnych za sterowanie ręczne położeniami kątowymi serwomechanizmów, 3 suwaki od sterowania automatycznego oraz przycisk przełączający rodzaj sterowania.

4.7 Gotowe urządzenie

Rysunek 58 przedstawia złożone urządzenie, które zostało przygotowane do przeprowadzenia badania. Mikrokontroler wraz z przewodami zasilającymi układ wraz z silnikami został umieszczony na płytce stykowej. Na czas testów NodeMCU V2 jest zasilany z komputera poprzez przewód USB, zaś serwomechanizmy z zasilacza o parametrach wyjściowych: napięcie 5 woltów oraz maksymalny prąd 2.1 A. W razie konieczności podany układ można zasilać z akumulatorów lub baterii ogniów.



Rysunek 58. Złożony układ gotowy do pracy.

5 Badania funkcjonalne manipulatora

Po złożeniu wszystkich komponentów urządzenia dokonano szeregu badań funkcjonalnych, mających na celu określenie zalet i wad zastosowanych rozwiązań oraz ograniczeń. W celu ich udokumentowania nagrano kilka filmów w formacie mp4, do których dostęp może mieć każdy zainteresowany:

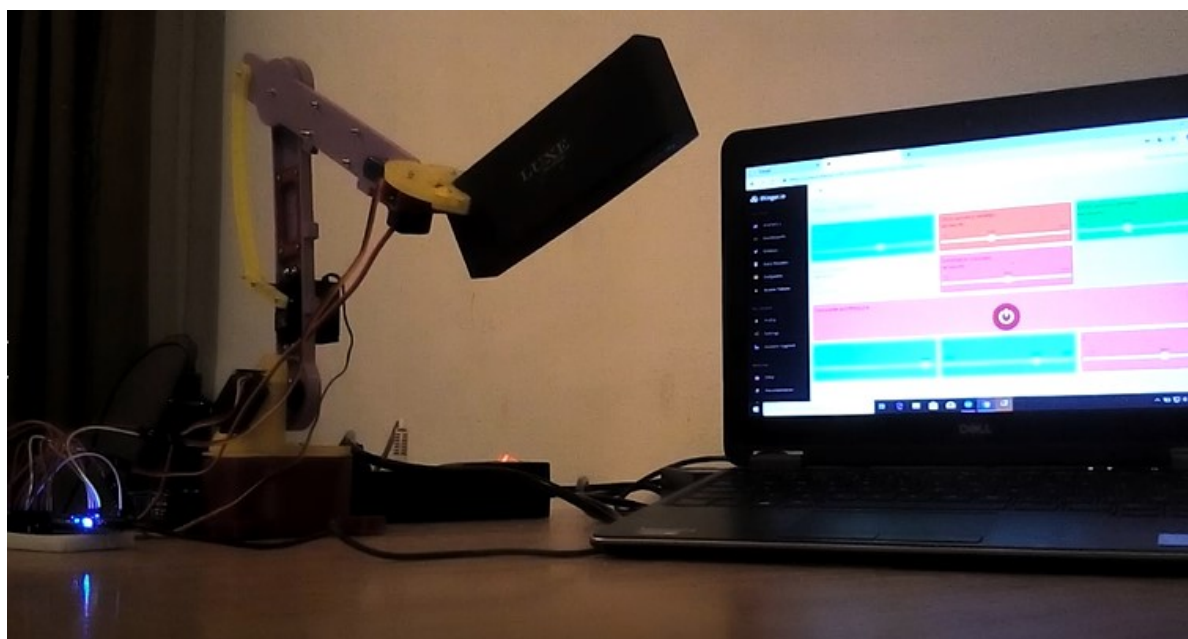
Film I: <https://youtu.be/i-NNPwamYdE>

Film II: <https://youtu.be/9AWUBQXgeHM>

Film III: <https://youtu.be/wS8GCI9G7s>

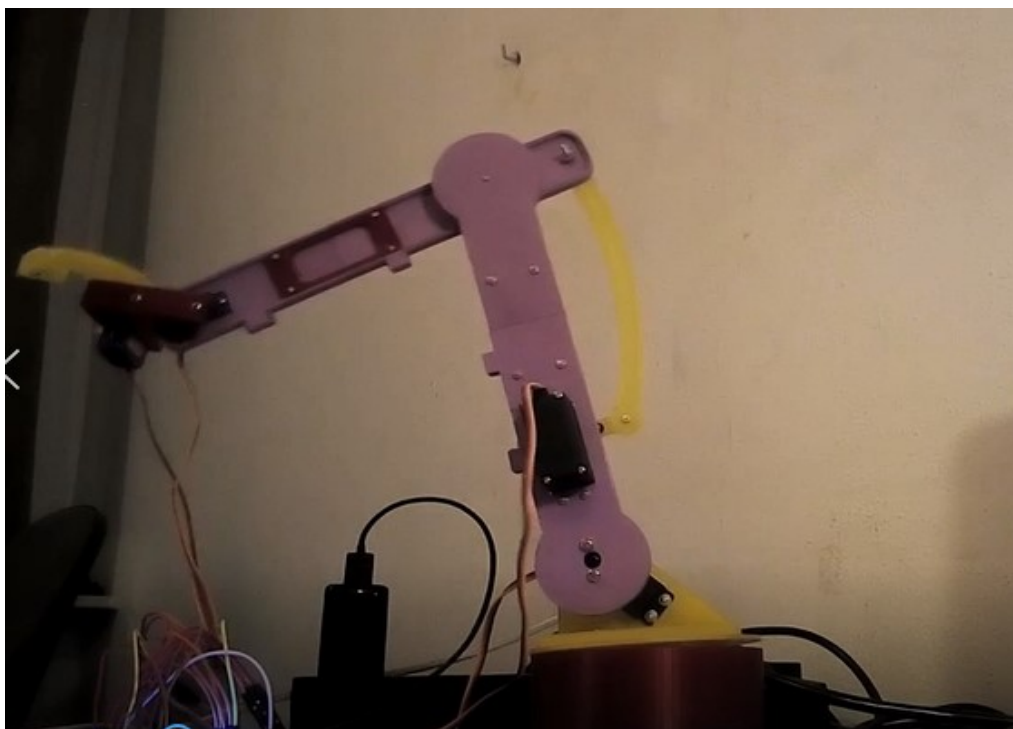
Pierwsze badanie miało na celu określenie poprawności algorytmu sterowania ręcznego. Polegało one na przeniesieniu elementu z jednego miejsca do drugiego (Rysunek 59). Scenariusz eksperymentu był następujący:

- Start z pozycji początkowej
- Dojazd do pozycji w której znajduje się przedmiot
- Chwycenie przedmiotu
- Przeniesienie elementu do drugiego miejsca
- Przejazd do pozycji początkowej



Rysunek 59. Manipulator z przenoszonym elementem.

W trakcie badań wszystkie serwomechanizmy manipulatora pracowały z drobnymi szarpnięciami oraz drganiem całej konstrukcji. Zauważono także pewną bezwładność zmieniania wartości kąta. Czas oraz jakość cyklu roboczego urządzenia jest w bardzo dużym stopniu zależne od doświadczenia oraz zręczności operatora. Jednym z ograniczeń, które stwierdzono w czasie badania był brak możliwości zmiany prędkości obrotowej serwomechanizmów.



Rysunek 60. Badanie sterowania automatycznego.

Drugie badanie miało na celu określenie jakości algorytmu sterowania automatycznego (Rysunek 60). Polegało ono na wprowadzeniu pozycji osprzętu roboczego we współrzędnych globalnych X, Y, Z a następnie zmierzeniu rzeczywistej pozycji. Osie X oraz Y przechodzą przez otwory w podstawie, zaś oś Z przez oś obrotu serwomechanizmu umieszczonego w podstawie manipulatora. Parametry zadane oraz zmierzone określono w Tabeli 2. Każdy wynik pomiaru został zaokrąglony do pełnych dziesiątek milimetrów.

Numer pomiaru	Współrzędne zadane [mm]			Współrzędne zmierzone [mm]			Błąd bezwzględny pozycji [mm]
	X _z	Y _z	Z _z	X	Y	Z	
1.	150	250	320	160	230	300	30±9
2.	50	100	200	80	130	190	44±9
3.	30	180	260	40	170	250	17±9
4.	150	120	60	120	130	70	33±9
5.	-130	260	170	-100	290	150	47±9
6.	-230	220	80	-200	240	70	37±9

Tabela 2. Współrzędne zadane oraz zmierzone.

Błąd pozycji został wyznaczony z wzoru (22):

$$b = \sqrt{(X_z - X)^2 + (Y_z - Y)^2 + (Z_z - Z)^2} \quad (22)$$

Gdzie:

b – błąd bezwzględny pozycji

X_z, Y_z, Z_z - zadane współrzędne

X, Y, Z - współrzędne zmierzone

Niepewność błędu pozycji została określona za pomocą wzoru (23):

$$\Delta p = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \quad (23)$$

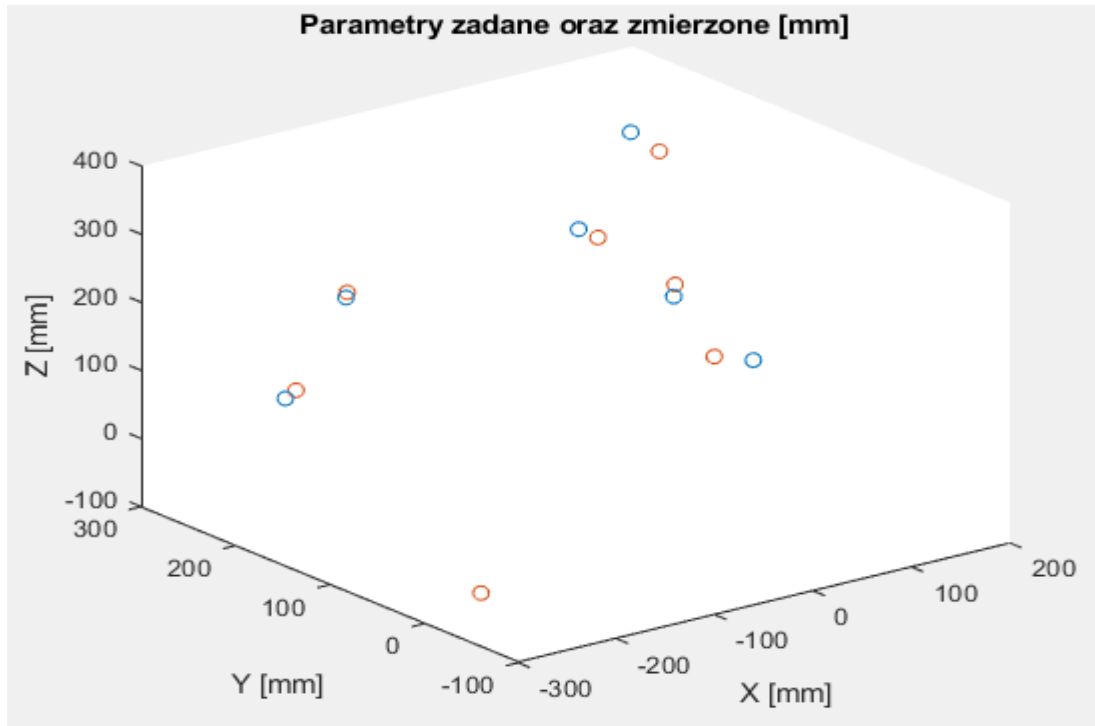
Gdzie:

Δp – niepewność błędu pozycji

Δx – niepewność pomiaru współrzędnej x $\Delta x = \pm 5mm$

Δy – niepewność pomiaru współrzędnej y $\Delta y = \pm 5mm$

Δz – niepewność pomiaru współrzędnej z $\Delta z = \pm 5mm$



Rysunek 61. Wykres położenia zadanego oraz zmierzonego.

Rysunek 61 przedstawia wykres położenia zadanego oraz zmierzonego. Kolorem niebieskim oznaczono parametry zadane, zaś czerwonym zmierzone. Błąd bezwzględny położenia osprzętu roboczego dla przeprowadzonych prób wynosi od 17 mm do 47 mm. Jest to stosunkowo niska wartość w porównaniu z wartością współrzędnych zadanych. Błąd położenia może wynikać z luzów w mechanizmie dźwigniowym oraz z ugięć ramion pod wpływem własnego ciężaru oraz ciężaru osprzętu roboczego.

Trzecie badanie dotyczyło także sterowania automatycznego. Celem było określenie jakie będzie zachowanie układu podczas zmiany położenia wzdłuż jednej osi układu współrzędnych, gdy pozostałe są stałe. W pierwszej serii zmianom podlegała współrzędna X , zaś $Y=0$ a Z było stałe. W czasie badania układ zmieniał swoje położenie tylko wzdłuż osi X , wysokość osprzętu roboczego zmieniała się nieznacznie, co może być związane z luzem powstałym w mechanizmie dźwigniowym. W drugiej serii badań X oraz Z było stałe, zaś zmianom podlegała zmienna Y . W tym przypadku manipulator poruszał się wzdłuż tylko jednej osi, prostopadłe do osi w pierwszej serii pomiarów. Wysokość osprzętu roboczego

pozostawała stała. Trzecia seria badań pozwoliła określić jakość sterowania automatycznego w czasie ruchu wzdłuż osi Z, podczas gdy parametry X oraz Y były niezmiennie. W trakcie tego badania także określono poprawność przycisku sterowanie automatyczne na panelu operatora. Podczas gdy przycisk był wciśnięty nie można było zmieniać położenia robota za pomocą suwaków związanych z obrotem każdego z serwomechanizmów. Zmiana położenia następowała podczas zmiany parametru X, Y, Z.

6 Zakończenie

6.1 Wnioski

Badania funkcjonalne manipulatora pozwalają stwierdzić, że urządzenie działało poprawnie. Manipulatorem można sterować poprzez wprowadzenie kątów obrotów każdego z serwomechanizmów lub za pomocą współrzędnych osprzętu roboczego realizując algorytm Newtona kinematyki odwrotnej.

Badając algorytm sterowania ręcznego, zauważono pewne problemy związane z płynnością ruchów oraz ich dokładnością, które wynikają one z braku doświadczenia operatora. W przypadku sterowania automatycznego stwierdzono nieznaczne błędy położenia, które nie przekraczają 30 mm na każdą z osi X, Y, Z. Przyczyną tego mógł być luz powstający na serwomechanizmach oraz mechanizmie dźwigniowym, a także metoda pomiarowa, która była obciążona pewnym błędem.

Mikrokontroler nie sygnalizował błędów związanych z obsługą chmury danych jak i sterowaniem aktuatorami, świadczy to o tym że algorytmy te były określone poprawnie bez zbędnych obciążeń. Obliczenia kinematyki odwrotnej metodą Newtona nie powodowały krytycznego obciążenia pamięci oraz procesora NodeMCU V2.

6.2 Posumowanie pracy

Po zbudowaniu oraz badaniach manipulatora przegubowego z wykorzystaniem koncepcji Internetu rzeczy stwierdzono, że urządzenie zostało zaprojektowane oraz zbudowane poprawnie, realizując w ten sposób cele postawione na początku pracy.

Konstrukcja mechaniczna manipulatora została zaprojektowana i zrealizowana w sposób poprawny. Urządzenie potrafiło wytrzymać obciążenia podczas realizowania cyklu roboczego.

Szkic programu napisany w Arduino IDE spełniał wymagania postawione w założeniach pracy tj. obsługa połączenia z chmurą oraz obsługą serwomechanizmów. Program nie był zbyt obszerny i skomplikowany, co skutkowało ciągłą i płynną pracą systemu.

Zastosowana przez autora koncepcja Internetu rzeczy może być wykorzystana w wielu urządzeniach automatyki przemysłowej, maszynach roboczych oraz pojazdach autonomicznych. Idea ta nawiązuje do obecnie rozwijającego się Przemysłu 4.0.

7 Bibliografia

[1] *Dokumentacja chmury Thinger.io*. (2019. 12. 09.)

Pobrano z lokalizacji docs.thinger.io

[2] *Historia robotów do roku 1969*. (2019. 12. 09.)

Pobrano z lokalizacji

http://www.asimo.pl/historia/robotyka_kalendarium_1969.php

[3] Kozłowski, K., Dutkiewicz, P. i Wróblewski, W. (2012).

Modelowanie i Sterowanie Robotów. Warszawa: PWN.

[4] ODROBINA, A. (2019. 12. 09).

SERWOMECHANIZM – Z CZEGO SIĘ SKŁADA I JAK DZIAŁA?

Pobrano z lokalizacji <http://abc-modele.pl/serwomechanizmy/>

[5] *Roboty przemysłowe*. (2019. 12. 09).

Pobrano z lokalizacji <http://www.asimo.pl/teoria/roboty-przemyslowe.php>

[6] Szkodny, T. (2011). *Podstawy robotyki*.

Gliwice: Wydawnictwo Politechniki Śląskiej.

[7] Traczyk, W. (2019. 12. 09).

Rynek robotów przemysłowych – rekord goni rekord.

Pobrano z lokalizacji <https://www.magazynprzemyslowy.pl/zarzadzanie-i-rynek/Rynek-robotow-przemyslowych-rekord-goni-rekord,11715,1>

[8] *Tryby pracy Access Pointa na przykładzie urządzenia TP-Link TL-WA901ND*.

(2019. 12. 09).

Pobrano z lokalizacji

https://www.dipol.com.pl/tryby_pracy_access_pointa_na_przykladzie_urzadzenia_tp-link_tl-wa901nd_bib505.htm

[9] Zubrycki, I. (2019. 12. 09).

Kinematyka i Jakobian.

Pobrano z lokalizacji http://gentle.pl/2017/01/02/kinematyka_i_jakobian.html

8 Spis symboli

R_z -naprężenie zredukowane

R_e -wytrzymałość na rozciąganie

X -współczynnik bezpieczeństwa

p -naciski powierzchniowe

F -siła

S -pole powierzchni nacisku

D -średnica okręgu opisanego na nakrętce

d -średnica otworu w nakrętce

a -wysokość mocowania pierwszego ramienia obrotowego

b -odległość osi obrotu ramienia pierwszego od osi obrotu podstawy

c -długość ramienia pierwszego

d -długość ramienia drugiego

e -odległość końca osprzętu od osi obrotu

f -długość orczyka serwomechanizmu obrotu ramienia górnego

g -długość dźwigni mechanizmu obrotu ramienia górnego

h -odległość osi obrotu ramienia górnego do mocowania dźwigni

i - odległość osi obrotu serwomechanizmu ramienia górnego do osi obrotu ramienia górnego

θ_1 -kąt obrotu podstawy manipulatora

θ_2 -kąt obrotu ramienia dolnego

θ_3 -kąt obrotu ramienia górnego

θ_4 -kąt obrotu osprzętu roboczego

θ_5 -kąt obrotu serwomechanizmu obrotu ramienia górnego

θ_6 -kąt obrotu dźwigni mechanizmu obrotu ramienia górnego

θ_7 -kąt obrotu ramienia górnego

J -macierz Jakobiego

$k(q)$ -obecne położenie osprzętu roboczego w danych parametrach

y_d -położenie docelowe osprzętu roboczego

b -błąd bezwzględny pozycji

X_z, Y_z, Z_z -współrzędne zadane

X, Y, Z -współrzędne zmierzone

Δp -niepewność błędu pozycji

Δx – niepewność pomiaru współrzędnej x

Δy – niepewność pomiaru współrzędnej y

Δz – niepewność pomiaru współrzędnej z

9 Spis ilustracji

Rysunek 1. Roboty kartezjańskie.	16
Rysunek 2. Roboty cylindryczne.	17
Rysunek 3. Roboty SCARA.	17
Rysunek 4. Roboty cylindryczne.	18
Rysunek 5. Roboty przegubowe.	18
Rysunek 6. Roboty z równoległą strukturą kinematyczną.	19
Rysunek 7. Przykłady robotów przemysłowych.	20
Rysunek 8. Koncepcja Internetu rzeczy.	21
Rysunek 9. Rodzaje sieci bezprzewodowych.	22
Rysunek 10. Tryb pracy Access Point.	23
Rysunek 11. Tryb pracy BPTP.	23
Rysunek 12. Tryb pracy BPTM.	24
Rysunek 13. NodeMCU V2.	27
Rysunek 14. Rozmieszczenie pinów na płytce NodeMCU V2.	28
Rysunek 15. Podstawowe elementy serwomechanizmu.	29
Rysunek 16. Schemat działania serwomechanizmu.	29
Rysunek 17. Serwomechanizm SG-92.	30
Rysunek 18. Serwomechanizm MG996.	31
Rysunek 19. Wymiary serwomechanizmu MG996.	31
Rysunek 20. Okno interfejsu chmury Thinger.io.	32
Rysunek 21. Schemat działania chmury danych.	33
Rysunek 22. Model podstawy.	34
Rysunek 23. Koncepcja mocowania silnika.	35
Rysunek 24. Model dekla.	35
Rysunek 25. Model mocowania mechanizmu obrotu ramienia.	36
Rysunek 26. Schemat obciążenia mocowania.	37
Rysunek 27. Wykres rozkładu naprężeń zredukowanych w mocowaniu.	37
Rysunek 28. Wykres rozkładu przemieszczeń mocowania.	38
Rysunek 29. Model ramienia dolnego.	38
Rysunek 30. Schemat obciążenia ramienia.	39
Rysunek 31. Wykres rozkładu naprężeń zredukowanych w ramieniu.	40
Rysunek 32. Wykres rozkładu naprężeń w płytce.	40
Rysunek 33. Wykres rozkładu przemieszczeń w ramieniu.	41
Rysunek 34. Model ramienia górnego.	41
Rysunek 35. Schemat obciążenia ramienia górnego.	42
Rysunek 36. Wykres rozkładu naprężeń zredukowanych w ramieniu górnym.	43
Rysunek 37. Wykres rozkładu naprężeń zredukowanych w płytce.	43
Rysunek 38. Wykres przemieszczenia ramienia górnego.	44
Rysunek 39. Model podstawy osprzętu roboczego.	44
Rysunek 40. Schemat obciążenia podstawy.	45
Rysunek 41. Wykres rozkładu naprężeń zredukowanych w podstawie.	45
Rysunek 42. Wykres rozkładu przemieszczeń podstawy.	46
Rysunek 43. Model osprzętu roboczego.	46

Rysunek 44. Wykres naprężeń w osprzęcie roboczym.	47
Rysunek 45. Wykres przemieszczeń osprzętu roboczego.	48
Rysunek 46. Model manipulatora.	48
Rysunek 47. Schemat wektorowy manipulatora.	49
Rysunek 48. Schemat wektorowy manipulatora.	50
Rysunek 49. Schemat wektorowy mechanizmu dźwigniowego.	51
Rysunek 50. Skrypt Matlaba do obliczania kinematyki mechanizmu dźwigniowego.	52
Rysunek 51. Wykres błędu bezwzględnego położenia.	53
Rysunek 52. Skrypt Matlaba określającą przestrzeń roboczą manipulatora.	53
Rysunek 53. Wykres przestrzeni roboczej manipulatora.	54
Rysunek 54. Skrypt Matlaba obliczający kinematykę odwrotną.	55
Rysunek 55. Wykres błędu bezwzględnego.....	56
Rysunek 56a. Szkic Arduinio IDE.	57
Rysunek 56cd. Szkic Arduinio IDE.....	58
Rysunek 57. Panel operatora.....	59
Rysunek 58. Złożony układ gotowy do pracy.	60
Rysunek 59. Manipulator z przenoszonym elementem.	61
Rysunek 60. Badanie sterowania automatycznego.....	62
Rysunek 61. Wykres położenia zadanego oraz zmierzonego.	63

10 Spis tabel

Tabela 1 Właściwości mechaniczne materiałów wykorzystywanych w druku 3d .	33
Tabela 2 Współrzędne zadane oraz zmierzone	62