

Model robota przemysłowego o równoległej strukturze kinematycznej-Delta

W artykule został zaprezentowany proces budowy oraz zasada działania algorytmów sterujących robotem typu delta. Robot został opracowany przez studentów kierunku Automatyka i Robotyka na wydziale Elektrotechniki i Informatyki Politechniki Rzeszowskiej. Głównym założeniem projektu było wykonanie manipulatora z jak największą minimalizacją kosztów z tym związanych, przy jednoczesnym zachowaniu zalet wynikających z równoległej struktury kinematycznej. W pracy przedstawione zostały kolejne etapy rozwoju, które pozwalały eliminować napotkane przeszkody.

Wprowadzenie

Robot został zaprojektowany na zajęciach pracowni problemowej przez studentów kierunku Automatyka i Robotyka. Na tych zajęciach tworzone są różnego rodzaju roboty. W tym przypadku chciano stworzyć manipulator mogący być, pod pewnymi względami, konkurencyjny dla konstrukcji dostępnych na rynku.

Roboty o równoległej strukturze kinematycznej typu delta charakteryzują się dużą prędkością i zwiększonym udźwigniem spowodowanym zastosowaniem trzech napędów. Jednak ceny rynkowe takich konstrukcji są bardzo wysokie. W tym projekcie głównym założeniem jest maksymalne zmniejszenie kosztów produkcji i jak najlepsze zachowanie zalet. Proces budowy został podzielony na kilka etapów takich jak projekt mechaniczny, wykonanie elementów, złożenie całej konstrukcji i regulacja długości ramion, dodanie części elektronicznej, napisanie algorytmu sterowania i, w końcowym etapie, poprawki mające na celu poprawę działania. Jak w przypadku profesjonalnych robotów, została dodana część funkcjonalności umożliwiająca definiowanie trajektorii z poziomu operatora oraz sterowanie manualne za pomocą specjalnie zaprojektowanego pilota. Projekt zawiera również wyświetlacz LCD, na którym opiera się poziom operatorski. Podczas ruchu końcówki manipulatora na wyświetlaczu znajdują się aktualne współrzędne położenia, natomiast gdy robot nie pracuje, wyświetlane jest menu zawierające funkcje robota. Głównymi problemami napotkanymi podczas pracy były:

- A. Brak synchronizacji czasowej;
- B. Drgania wprowadzane przez dużą dynamikę układu;
- C. Brak możliwości zastosowania regulatora PID, wynikający z typu zastosowanych serwomechanizmów.

Ogólny schemat działania, podobnie jak pierwsze wymogi postawione przed manipulatorem zostały sformułowane zgodnie z materiałami dostępnymi w książce [1].

1. Konstrukcja mechaniczna robota

Pierwszym zadaniem przy konstruowaniu manipulatora było założenie jaki udźwignie będzie on posiadał oraz wyznaczenie pola roboczego. Uznaliśmy że udźwignie około 200 gram oraz pole robocze 120mm x 120 mm będzie wystarczające. W projekcie wykorzystano kod źródłowy z kalkulatora pola roboczego dla manipulatorów typu delta zaimplementowany na stronie internetowej [2]. Najlepsze wyniki przy jednoczesnym zachowaniu małych wymiarów robota uzyskano przy długości ramienia czynnego 110 mm oraz biernego 140 mm. Następnie został dobrany napęd ramienia czynnego, ze względu na ograniczony budżet użyto serwomechanizmu modelarskiego. Zastosowano serwomechanizm firmy

TowerPro MG-996R. Jego udźwig to 11kg na cm, co przy zastosowanych ramionach umożliwia uzyskanie 3kg podnoszonego ciężaru , dzięki czemu uzyskano zadowalającą dynamikę ruchu. Następnie zaprojektowano manipulator w programie Autodesk Inventor 2018. Po wykonaniu wszystkich rysunków technicznych zaczęto wykonywać elementy. Materiał z jakiego został wykonany robot to w większości aluminium gatunku PA6, wybrany materiał wpłynął na ograniczenie masy robota, a w szczególności jego ruchomych elementów. Większość elementów została wykonana przez autorów na maszynach konwencjonalnych, jedynie płytę dolną robota zlecono firmie zewnętrznej z braku dostępu do obrabiarki sterowanej numerycznie. Przeguby kulowe, które zostały wybrane do projektu posiadają gwinty prawe oraz lewe, dzięki czemu stworzyliśmy śrubę rzymską, w celu łatwego i dokładnego ustalenia długości ramienia biernego.

W dalszej części umieszczone są zdjęcia poszczególnych najważniejszych elementów wraz z opisem.

Płyta górna (rys 1.1) została wykonana z aluminium o grubości 10 mm i średnicy 225 mm. Otwory mocujące resztę elementów do płyty zostały wywiercone z dużą dokładnością na wiertarce współrzędnościowej. Wycięcia o szerokości 20mm zostały wyfrezowane co 120 stopni.



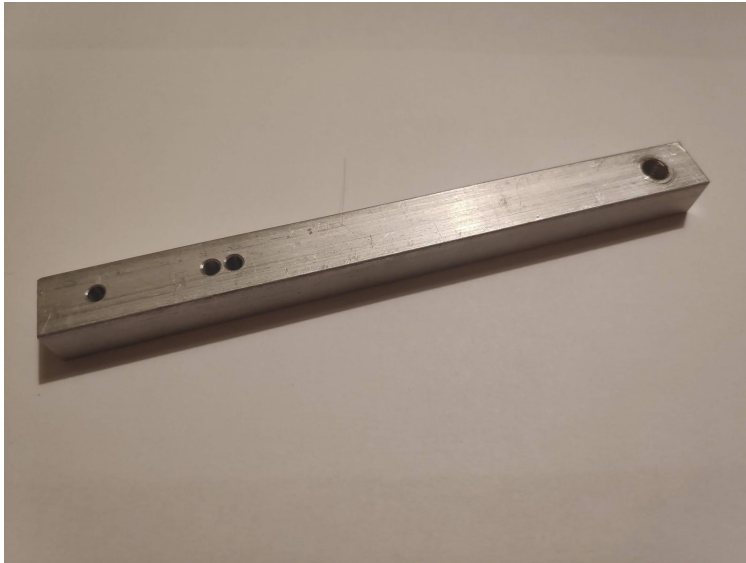
Rysunek 1.1 - Płyta górna

Mocowanie serwomechanizmów modelarskich (rys 1.2). Aby pewnie zamocować serwo modelarskie należało wykonać uchwyt w kształcie litery C.



Rysunek 1.2 - Mocowanie serwomechanizmów

Ramię napędowe czynne (rys 1.3) wykonane zostało z płaskownika aluminiowego 10x12mm, Ramię ma długość 130mm. W celu zachowania precyzji wykonania, otwory do mocowania z orczykiem serwomechanizmu oraz łącznika z następnym ramieniem zostały wywiercone w rozstawie 110 mm na wiertarce współrzędnościowej.



Rysunek 1.3 - Ramię napędowe czynne

Ramię napędowe bierne (rys 1.4) jest to złożenie przegubów wraz z łącznikiem. Łącznik został wykonany w wałka aluminiowego $\varnothing 10\text{mm}$ o długości 100mm. Na jednym z końców został wykonany gwint lewy zewnętrzny, natomiast po przeciwległej stronie został wywiercony otwór, który został nagwintowany.



Rysunek 1.4 - Ramię napędowe bierne

Płyta (rys 1.5) dolna jest zrobiona z jednego kawałka aluminium $\varnothing 90$, który został obrobiony na frezarce CNC. Po wykonaniu elementu, otwory z gwintami mocujące łączniki zostały wywiercone 36,5 mm od osi płyty dolnej.



Rysunek 1.5 - Płyta dolna

Elementem wykonawczym (rys 1.6) jest zaprojektowany długopis, posiadający w sobie sprężynę, która umożliwia niewielki nacisk na kartkę papieru, dzięki czemu kartka nie zostaje przecięta podczas pisania. Wykorzystanie długopisu umożliwia łatwą weryfikację odchyleń od zadanej trajektorii.



Rysunek 1.6 - Element wykonawczy

Podstawa widoczna poniżej (rys 1.7) została wykonana z płyty MDF, do której został przykręcony wałek aluminiowy wraz z płytą, aby zminimalizować ilość użytego materiału wałek został zrobiony z trzech części połączonych ze sobą dzięki rozszerzalności materiałów. Element wkładany w środek walca został schłodzony natomiast elementy przymocowane do górnej i dolnej części zostały ogrzane spowodowało to rozszerzenie się materiału i dało możliwość umieszczenia schłodzonej części w środku. W normalnych warunkach elementy posiadające podobną temperaturę są zaciśnięte. Efektem tego jest sztywna konstrukcja i warunki zewnętrzne nie mają wpływu na działanie robota.



Rysunek 1.7 - Podstawa

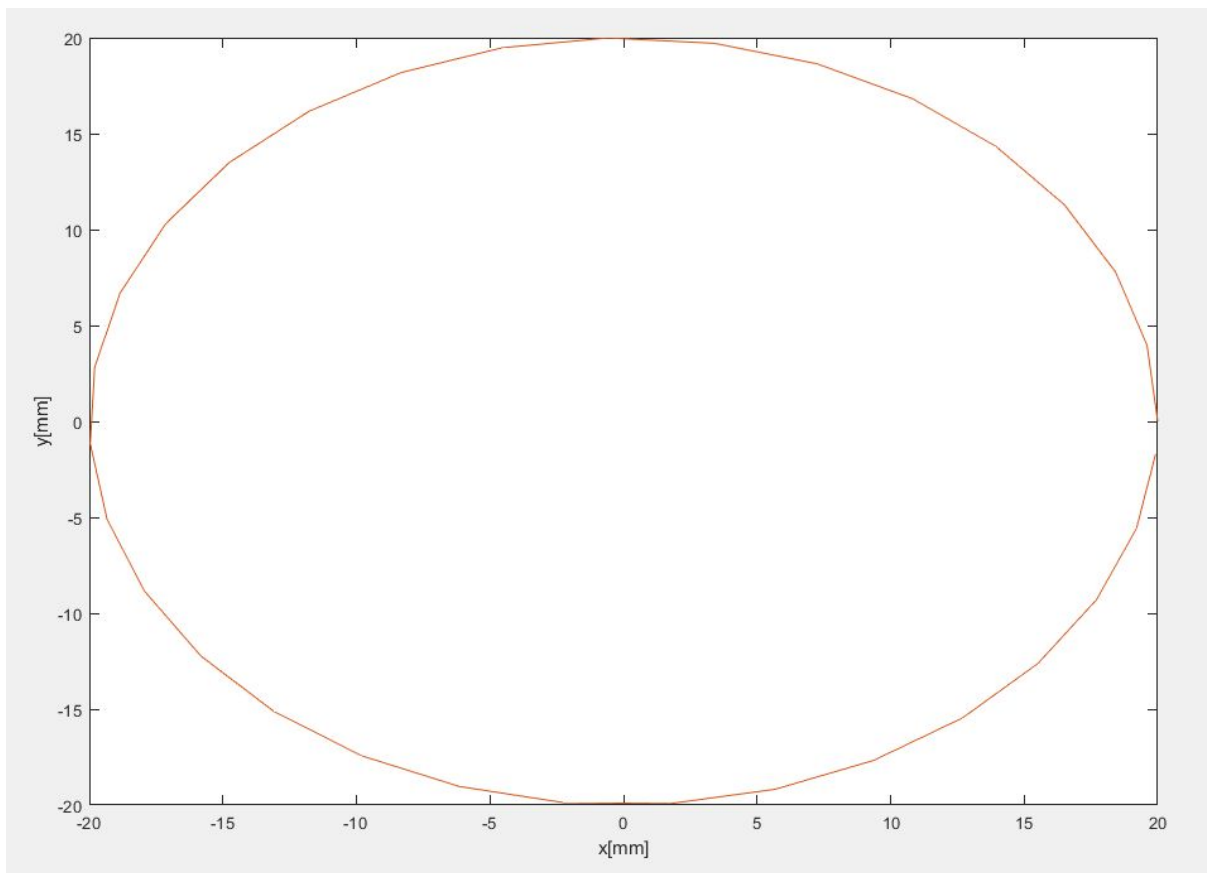
Po złożeniu konstrukcji prezentuje się ona następująco (rys 1.8)



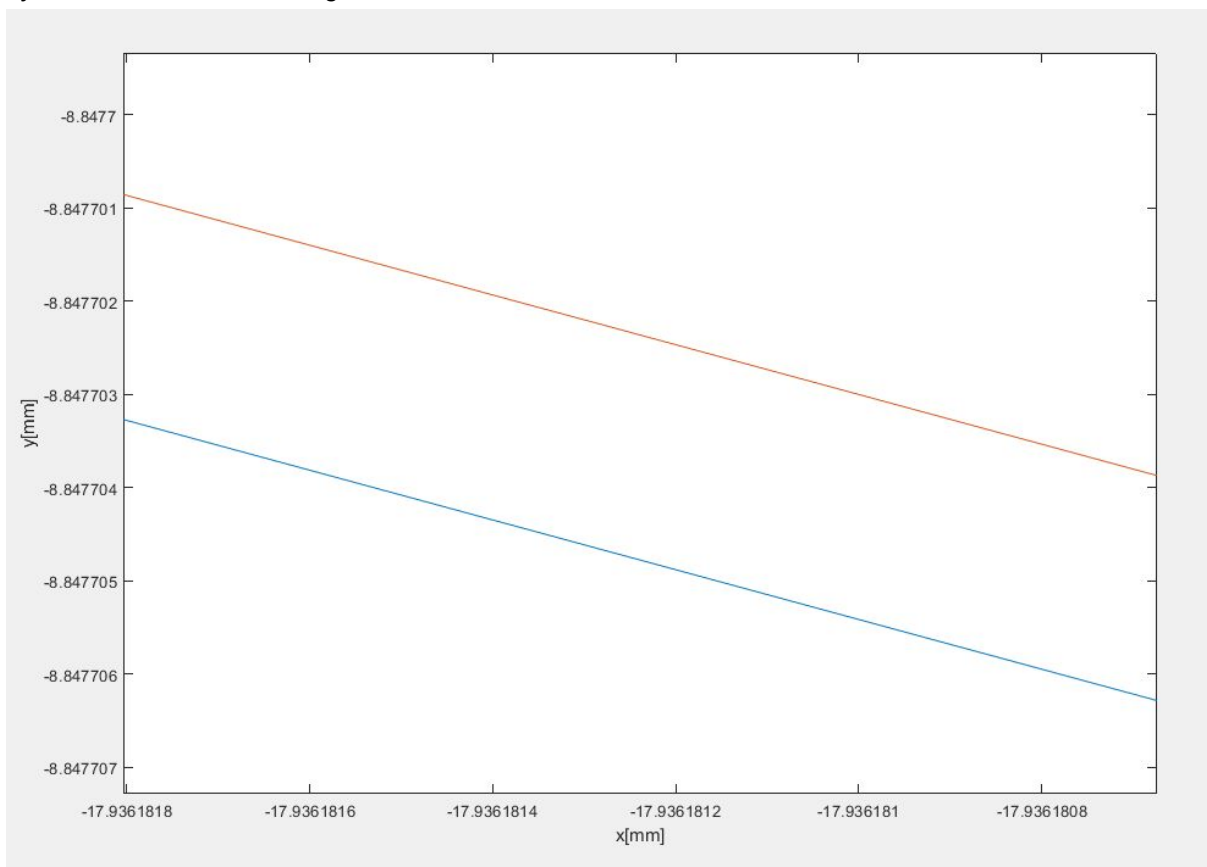
Rysunek 1.8 - Konstrukcja robota

2. Kinematyka

Jednym z kłopotów w sterowaniu robotów jest wyznaczenie równań kinematyki. Aby przyspieszyć budowę układu sterowania, wykorzystano gotowe wyprowadzenie wzorów, które znajdowało się na stronie [3]. W celu sprawdzenia poprawności zastosowanego algorytmu, stworzono w środowisku Matlab skrypt przeliczający punkty w przestrzeni roboczej manipulatora na wartości przegubowe, a następnie, dzięki zastosowaniu kinematyki prostej, zostały one przeliczone na punkty w układzie kartezjańskim. Błąd jaki uzyskano zawierał się w granicach e^{-10} . Wyniki tych testów zostały zaprezentowane poniżej (rys 2.1 oraz rys 2.2).



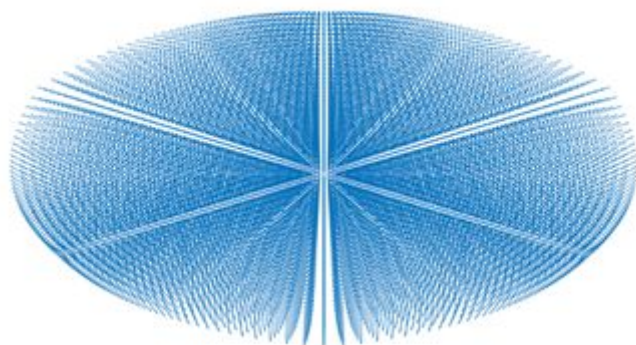
Rysunek 2.1 - Testowana figura



Rysunek 2.2 - Różnica kinematyk

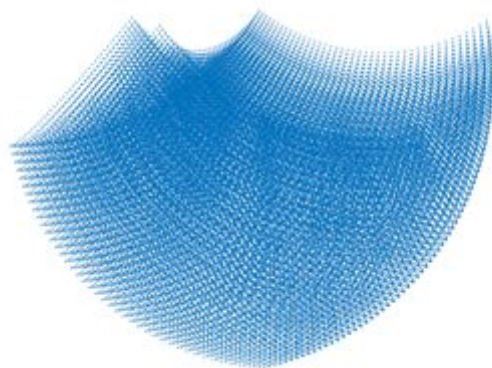
Z wykorzystaniem kinematyki prostej została wyznaczona przestrzeń robocza, pozwoliło to rozmieścić bierne części konstrukcji tak, aby elementy nie uległy zderzeniu.

Widok przestrzeni roboczej z góry (rys 2.2):



Rysunek 2.2 - Przestrzeń robocza z góry

Widok z boku (rys 2.3):

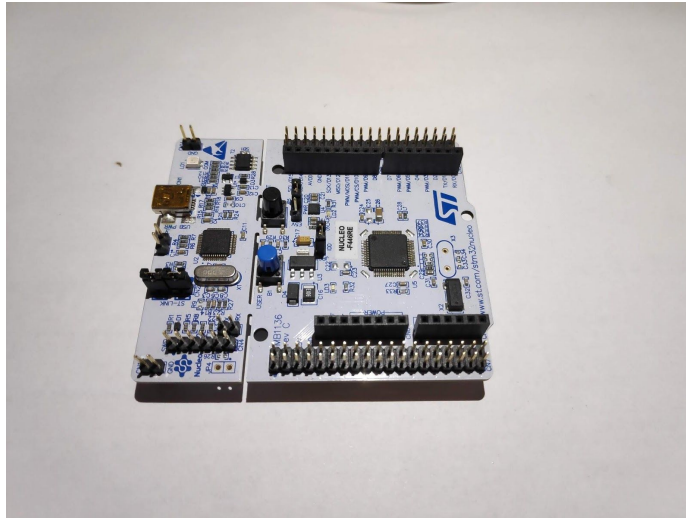


Rysunek 2.3 - Przestrzeń robocza z boku

3. Elektronika

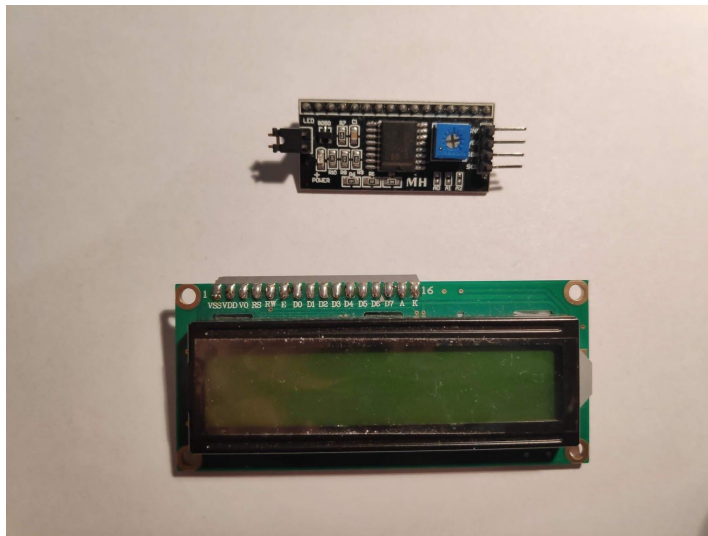
Sterowanie odbywa się poprzez STM32 NUCLEO-F446RE (rys 3.1). Mikrokontroler ten posiada taktowanie 180 Mhz, co jest istotne podczas sterowania serwomechanizmami. Zastosowanie wolniejszej jednostki skutkuje niezadowalającą prędkością wyznaczania punktów, co pośrednio przekłada się na błędy trajektorii. Kolejną cechą mającą wpływ na wybór tego stm jest wielkość pamięci flash - 512 kB, dzięki temu program sterujący mieścił

się w niej bez problemu. Wykorzystanie trzydziestodwu bitowego mikrokontrolera jest optymalne, biorąc pod uwagę ograniczenia finansowe, sterowniki ośmiobitowe są w tym przypadku niewystarczające.



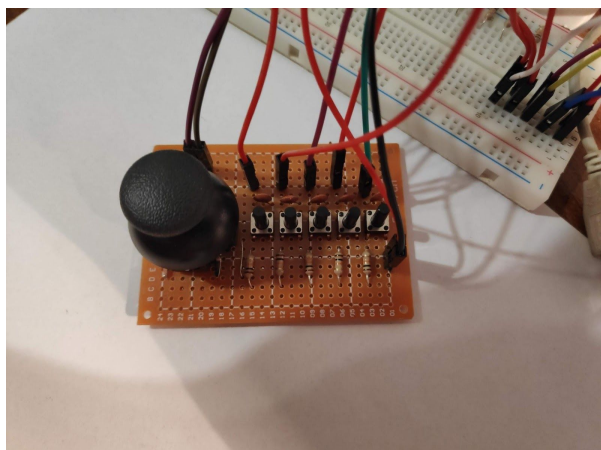
Rysunek 3.1 - STM32 NUCLEO-F445RE

Projekt został wyposażony w wyświetlacz LCD 16x2 wraz z płytką umożliwiającą komunikację po magistrali I2C (rys 3.2).



Rysunek 3.2 - Wyświetlacz LCD

Wygląd płytki służącej do sterowania ruchem robota, a także umożliwiającą wybór funkcji z menu na wyświetlaczu, został zaprezentowany poniżej (rys 3.3).



Rysunek 3.3 - Moduł sterowania

Na płytce został umieszczony joystick, aby uzyskać możliwość jednoczesnej kontroli nad kierunkiem i prędkością końcówki manipulatora.

4. Układ sterowania

W początkowym etapie projektu wykorzystano płytkę Arduino UNO. Dzięki wsparciu środowiska Matlab dla danych urządzeń oparto na nim system sterowania. Pierwszym krokiem było zapoznanie się z funkcjami oferowanymi w bibliotece „Arduino Support from MATLAB”. Na jej podstawie został stworzony główny skrypt odpowiedzialny za wybór wykonywanej czynności oraz funkcje odpowiedzialne za obliczanie punktów w przestrzeni kartezjańskiej, przeliczanie ich na kąty i sterowanie serwomechanizmami. Po przeprowadzeniu testów zauważono problem w determinizmie czasowym, który jest najbardziej widoczny przy rysowaniu wierzchołków trójkąta. Spowodowane jest to nierównym cyklem obliczeń w systemie Windows. Obiekt wykonawczy nie otrzymywał danych w równych odstępach czasu, przez co miał tendencję do pomijania punktów na zadanej trajektorii. Po konsultacjach system sterowania został zmieniony na płytkę wykonawczą w postaci STM32 NUCLEO-F446RE, dzięki czemu obliczenia wykonują się szybciej i z większą precyzją, zrezygnowano też z używania programu Matlab do zadawania trajektorii. Program sterujący został napisany w środowisku programistycznym Arduino IDE. Ponadto zastosowano odmienny algorytm sterowania. Na początku obliczane są wartości w układzie kartezjańskim, które przechowywane są w dynamicznie tworzonej tablicy, następnie układ oblicza wartości kątów dla danego ramienia i zapisuje je w tablicach. W ostatnim kroku program w różnych cyklach wysyła do serwomechanizmów zadane kąty. Dzięki tym zmianom uzyskano zadowalające rezultaty, mając na uwadze niedokładności użytych napędów.

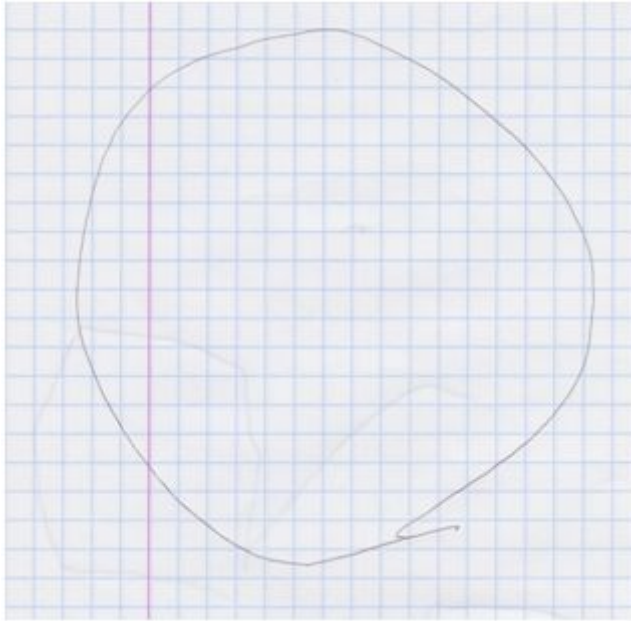
Kolejną próbą polepszenia działania manipulatora było uzyskanie informacji ze sprzężenia zwrotnego, poprzez wykorzystanie żyroskopu i akcelerometru MPU 6650. Jednak przy maksymalnych prędkościach akcelerometr ulega rozregulowaniu i to jest błędem uniemożliwiającym zastosowanie go w naszym projekcie.

Rzeczywisty układ sterowania został wyposażony w funkcje rysujące podstawowe figury geometryczne takie jak koło, trójkąt, kwadrat oraz regulacje ich wielkości za pomocą menu, do którego jest dostęp poprzez wyświetlacz LCD. Figury te służą do testowania dokładności robota w

różnych częściach pola roboczego. W menu zawierają się też funkcje operatorskie takie jak sterowanie manualne za pomocą joystick'a znajdującego się na pilocie oraz możliwość definicji własnych trajektorii bez ingerencji w kod programu. Tworzenie własnej trajektorii opiera się na dojechaniu końcówką manipulatora w żądany punkt, następnie zapisanie go i powtórzeniu tego dla kolejnych punktów. Po zaprogramowaniu ruchów można przejść do funkcji odczytującej zapisane punkty.

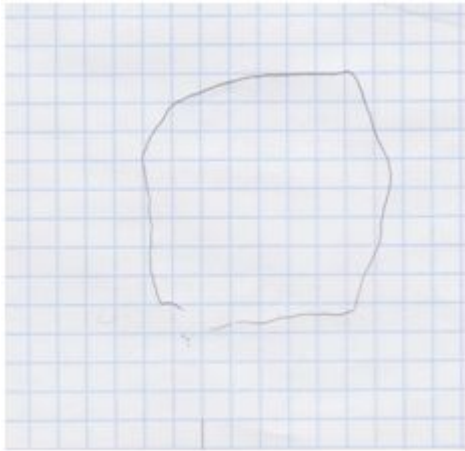
5. Wyniki pracy

Pierwszą figurą, którą wykonano jest koło. Jak widać na rysunku 5.1 koło nie jest idealne. Spowodowane jest to niedokładnością zastosowanych napędów.

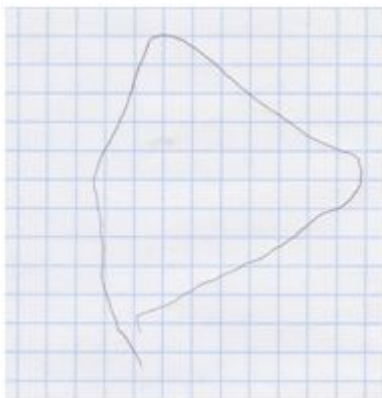


Rysunek 5.1 - Koło

Rysunki kwadratu (rys 5.2) oraz trójkąta (rys 5.3) pozwalają zauważyć kolejne niedokładności wynikające z braku synchronizacji czasowej (łuki zamiast kątów ostrych pomiędzy bokami). Różnice między punktem początkowym i końcowym pozwalają zauważyć wady regulacji typu P oraz dużą dynamikę układu.



Rysunek 5.2 - Kwadrat



Rysunek 5.3 - Trójkąt

6. Wnioski

Wykorzystane w projekcie serwonapędy nie pozwalają uzyskać zadowalającej dokładności robota. Wynika to z regulacji typu P. Przypuszcza się, że zastosowanie droższych napędów w postaci silników krokowych i dodanie do nich enkoderów, które z kolei daje dostęp do informacji ze sprzężenia zwrotnego, umożliwiłoby regulację typu PID eliminującą większość niedokładności. Jednak takie działanie jest sprzeczne założeniami projektu, ponieważ znacznie zwiększyłyby się koszty produkcji.

Założenia odnośnie udźwigu i prędkości działania robota zostały spełnione.

7. Bibliografia

- [1] 1999, *Podstawy robotyki Teoria i elementy manipulatorów i robotów* - Wydawnictwo: WNT
- [2] Źródło internetowe, *Rotary Delta Robot Forward/Inverse Kinematics Calculations*,
<https://www.marginallyclever.com/other/samples/fk-ik-test.html>, data dostępu: 29.04.2019
- [3] Źródło internetowe, *Delta robot kinematics*,
http://forums.trossenrobotics.com/tutorials/introduction-129/delta-robot-kinematics-3276/?fbclid=IwAR1vML4WuLdC8MBLEzFJnQJQeJMdSJog_ZXv7EAq_xtqYztoq-wy_M-J5lc, data dostępu: 29.04.2019