1. Marcin Przestrzelski, 2. Yaroslav Zolin

Politechnika Warszawska, Wydział Mechatroniki (1), Politechnika Warszawska, Wydział Mechatroniki (2)
ORCID: 1. Xxxxx; 2.xxxxxxx

doi: xxxxx.

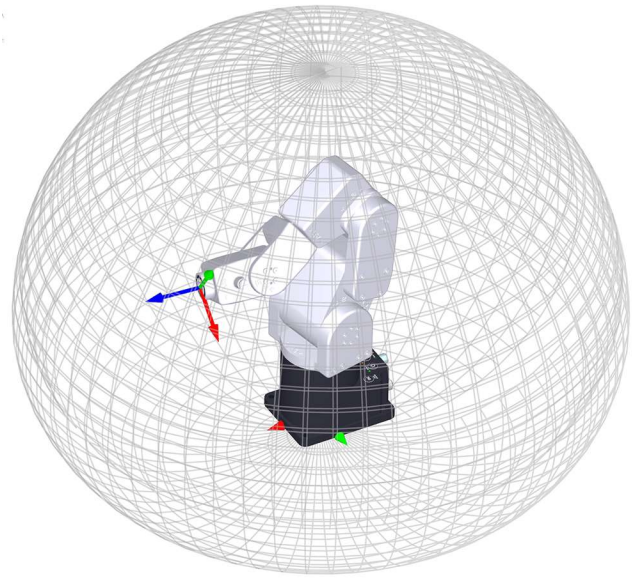
Sterowanie ramienia robota przy pomocy środowiska MATLAB

Streszczenie. *Raport z opisem sterowania manipulatorem za pomocą Arduino. Program napisany w środowisku MATLAB.*

Słowa kluczowe: serwomechanizm, manipulator, MATLAB, sterowanie, programowanie, robotyka, chwytak, Arduino

Wstęp

Obecnie stało się niemal standardem, przekazywanie obowiązków robotom przemysłowym. Przeważają one nad ludźmi między innymi precyzją wykonywanej pracy i siłą jaką są w stanie działać na obiekty. Dodatkowo mogą one pracować w trudnych warunkach, w których ludzie nie byliby w stanie i wykonywać powierzone im zadania bez przerw. Jednym z częściej wykorzystywanych typów robotów są manipulatory. Są to narzędzia w kształcie ramienia, zakończone chwytakiem. Różne manipulatory mogą się poruszając w różny sposób, w zależności od ich konstrukcji. Zazwyczaj ramiona składają się z członów. Człony dzielimy na takie, które zapewniają posuw liniowy i takie które pozwalają wykonywać ruch obrotowy. Każdy człon ma za zadanie dodanie jeden stopień swobody do ruchu. Aby robot mógł dotrzeć do każdego punktu w przestrzeni, w swoim zasięgu, musi mieć co najmniej trzy stopnie swobody. Można jednak dodać ich więcej. Robot wtedy jest w stanie dotrzeć do wskazanego punktu i jednocześnie przestrzegać innych ograniczeń przestrzeni roboczej. Na przykład może operować obiektem od wskazanej strony. Wybraliśmy robota o trzech stopniach swobody, poruszanych przez trzy człony obracające (RRR).



Rys.1. Pole robocze manipulatora [7]

Taki układ pozwala operować na dużym polu roboczym, o zasięgu równym w każdym kierunku. Rozwiązanie takie jednak posiada wady. Między innymi, aby znać pozycję chwytaka manipulatora, konieczne jest przekształcenie matematyczne, pozycji kątowych poszczególnych członów, na współrzędne kartezjańskie.

Do zaprogramowania robotów wyokrzystuje się wiele oprogramowani. Bardzo częstym programem jest ROS (Robot Operating System). Jeżeli decydujemy się na użycie mikrokontrolera Arduino, można użyć oficjalnego środowiska programistycznego. Rozwiązanie to nie pozwala jednak na stworzenie satysfakcjonującego interfejsu użytkowego i wizualizacji ruchu robota. Z tego powodu zdecydowaliśmy się na użycie środowiska MATLAB. W tym programie można użyć gotowej biblioteki Robotic Toolbox lub napisać program w czystym MATLAB-ie. Ze względu na to, że projekt ma mieć charakter edukacyjny, zdecydowaliśmy się na napisanie kodu od podstaw. Do projektu użyliśmy jedynie rozszerzeń *„App Designer”*, aby stworzyć interfejs graficzny i *„Arduino Support for MATLAB”*, aby korzystać z funkcji które upraszczają programowanie mikrokontrolera.

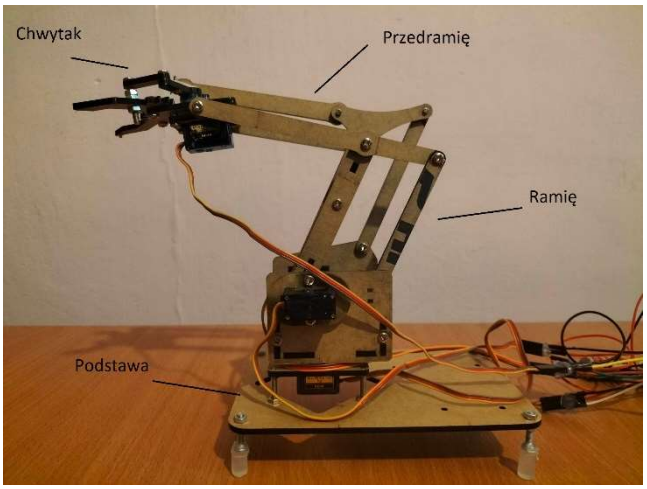
Założenia projektowe

Celem projektu było stworzenie oprogramowania służącego do sterowania modelarskim ramieniem robota. Robot sterowany miał być przy pomocy mikrokontrolera Arduino Uno programowanego przez środowisko programistyczne MATLAB. Podstawowym zadaniem jakie program ma spełniać jest sterowanie manipulatorem i tworzenie wizualizacji jego ruchu. Dodatkowo ma mieć możliwość tworzenia kolejki kroków które robot w zastosowaniu przemysłowym mógłby powtarzać. Ma być też możliwa regulacja prędkości ruchu robota. Aby sprawdzić jakość realizowania poleceń użytkownika, potrzebna jest też analiza odchyłek i czasu jaki jest potrzebny, aby manipulator dochodził do pozycji zadawanej.

Ramię robota

W projekcie wykorzystano ramię o trzech stopniach swobody, typu RRR. Robot, składa się z podstawy, ramienia, przedramienia i chwytaka kształtowego. Ramię i przedramię mają po 79mm długości. Manipulator napędzany jest

czterema serwomechanizmami.



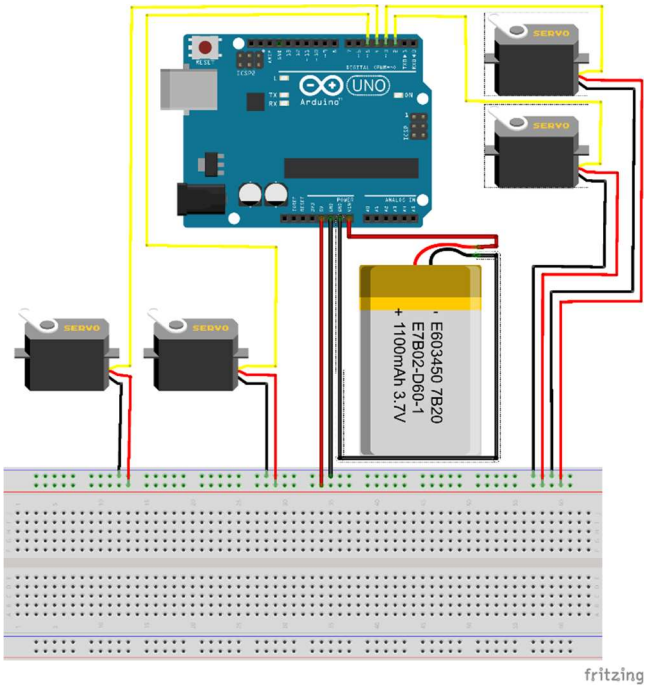
Rys.2. Budowa manipulatora

Trzy obracają robotem w trzech osiach prostopadłych do siebie. Czwarty serwomechanizm odpowiedzialny jest za zamykanie i otwieranie chwytaka.

Podstawa posiada cztery podkładki, które amortyzują i stabilizują pozycję robota podczas szybkiego trybu pracy robota.

Elektronika

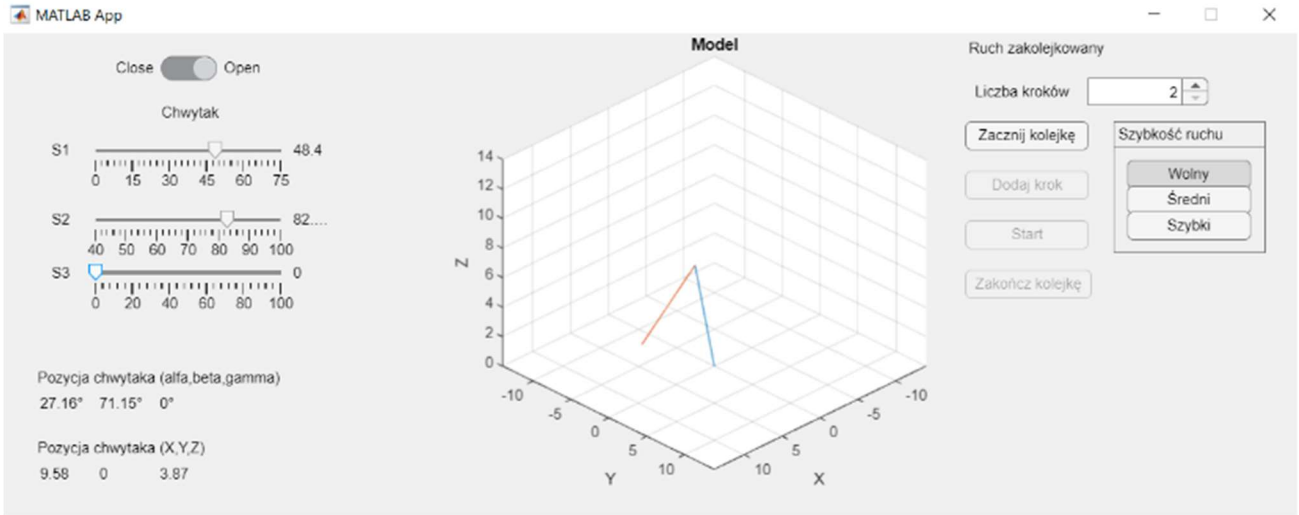
Robot sterowany jest za pomocą mikrokontrolera Arduino Uno. Jest to powszechnie wykorzystywany mikrokontroler, do tworzeniu robotów edukacyjnych. Wynika to głownie z niskiej ceny, szerokiej dostępności i niskiego progu wejścia dla początkujących. W tym projekcie parametry oferowane przez ten mikrokontroler będą wystarczające. Zasilany jest on pięcioma voltami z komputera z programem, przez wejście USB-B. Posiada także dodatkowe źródło zasilania w postaci akumulatora litowo-jonowo-polimerowego. Zapewnia on stałe źródło wysokiego prądu na serwomechanizmach. Do Arduino połączone są cztery serwomechanizmy do pinów cyfrowych PWM.



Rys.3. Schemat połączenia elementów elektronicznych

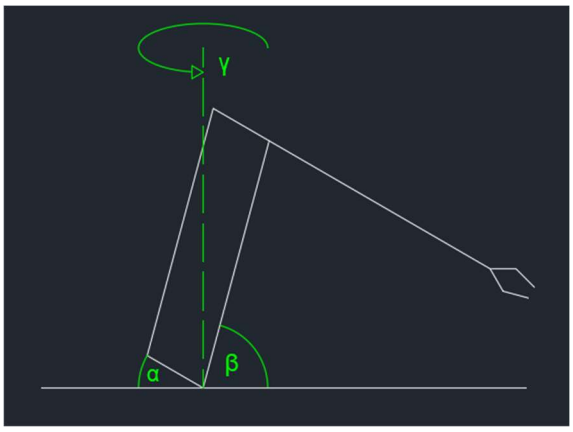
GUI

Robot sterowany jest przy pomocy GUI (Graphical User Interface. Zostało one przygotowane przy pomocy rozszerzenia do MATLAB-a, *„App Designer”*.



Rys.4. GUI

Na środku znajduje się model manipulatora, który ukazuje jego pozycję w czasie rzeczywistym. W lewym górnym rogu znajduje się przycisk pozwalający otworzyć i zamknąć chwytak. Pod nim są suwaki którymi steruje się trzema serwomechanizmami oddzielnie, tak aby ustalić żądaną pozycję chwytaka. W projekcie zostały zastosowane suwaki, aby użytkownik mógł w sposób jak najbardziej płynny ustalać pozycję manipulatora. Poniżej znajduje się na bieżąco wyznaczana pozycja chwytaka, wyrażona w dwóch układach współrzędnych: biegunowym wyrażona w kątach *(α, β, γ)* Rys.4. i kartezjańskim, wyrażonym w odległościach od punktu *(0,0,0) (x, y, z)*. Dla układu kartezjańskiego zakładamy, że punkt *(0,0,0)* znajduje się w punkcie styku ramienia z dwoma serwomechanizmami pozwalającymi zmieniać kąty α i β.



Rys.5. Schemat układu biegunowego

Po prawej stronie znajduje się panel sterowania do trybu pracy automatycznego.

Tryby pracy

Robot jest w stanie pracować w dwóch trybach:
- manualnym
- automatycznym

Tryb manualny pozwala na ustalenie wybranej pozycji. Przy użyciu trybu automatycznego, można stworzyć listę kroków którą następnie robot będzie powtarzał. Aby stworzyć kolejkę należy najpierw wybrać liczbę kroków kolejki. Gdy to ustalimy należy wcisnąć przycisk *Zacznij kolejkę*. Teraz przy pomocy panelu trybu manualnego ustalić pozycję, która ma się znajdować w kolejce. Aby zatwierdzić tą pozycję nasikamy przycisk *Dodaj krok*. Następnie ustalamy kolejną interesującą nas pozycję. Następnie wybieramy w którym trybie szybkości chcemy, aby manipulator dochodził do kolejnych pozycji. Gdy to zrobimy, musimy, aby je odtworzyć należy wcisnąć przycisk *Start*. Robot teraz przejdzie przez wszystkie zadeklarowane pozycje po kolei. Jeżeli chcemy stworzyć nową kolejkę, musimy najpierw usunąć poprzednią, przez wciśnięcie przycisku *Zakończ kolejkę*.

Modelowanie

Jednym z problemów w projekcie było stworzenie modelu przetwarzającego sygnały wejściowego z suwaków *s* na sygnały zadające napięcie na serwomechanizmy *v*. Wartości sygnałów na serwomechanizmach mieszczą się w zakresie (0, 1). Chcieliśmy, aby użytkownik na suwakach poruszał się w procentach, więc przekształcenie musiało przyjmować następujące równanie:

$$v = \frac{s}{100} \ (1)$$

Następnie sprawdziliśmy czy konstrukcja ramienia pozwala na zadawanie na serwomechanizm całego jego zakresu dopuszczalnych wartości. Podczas prób okazało się, że pozycje dopuszczalne dla serwomechanizmów to:
- dla s_1 zakres to (0, 0.75)
- dla s_2 zakres to (0.4, 1)
- dla s_3 zakres to (0, 1)

Dostosowaliśmy to przez ograniczenie zakresów na suwakach.

Kolejnym z problemów które należało przeanalizować, była kwestia tego jaka jest zależność między wartościami na suwakach a wartościami kątów w układzie biegunowym. Z pomiarów wartości kątów dla wartości nastawionych na suwakach dla poszczególnych serwomechanizmów wyznaczono następujące równania:

$$\alpha = \frac{5850 - 90 \cdot s_1}{55} \ (2)$$

$$\beta = 195 - \frac{3}{2} s_2 \ (3)$$

$$\gamma = 1.8 \cdot s_3 \ (4)$$

Zakresy tych kątów wynoszą:
- dla α (-16.36°, 106.36°)
- dla β (45°, 135°)
- dla γ (0°, 180°)

Tak wyznaczone kąty wyrażone są w stopniach.

Znając pozycje w biegunowym układzie współrzędnych, jesteśmy przejść do układu kartezjańskiego. Analizując geometrię manipulatora dochodzimy do następujących równań:

$$x_1 = 7.9 \cdot \sin\beta \ (5)$$

$$y_1 = 7.9 \cdot \cos\beta \cdot \cos\gamma \ (6)$$

$$z_1 = 7.9 \cdot \cos\beta \cdot \sin\gamma \ (7)$$

$$x_2 = x_1 - 7.9 \cdot \cos(90° - \alpha) \ (8)$$

$$y_2 = y_2 + 7.9 \cdot \sin(90° - \alpha) \cdot \cos\gamma \ (9)$$

$$z_2 = z_2 + 7.9 \cdot \sin(90° - \alpha) \cdot \sin\gamma \ (10)$$

Przy czym *(x_1, y_1, z_1)* to pozycja końca ramienia i początku przedramienia manipulatora, a pozycja *(x_2, y_2, z_2)* to punkt końca przedramienia i punkt, w którym znajduje się chwytak. Punkt początku przedramienia ma współrzędne (0, 0, 0). Teraz jesteśmy w stanie wyrysować model manipulatora w postaci prostych między tymi trzema punktami. Powyższe obliczenia i wyrysowanie dzieje się w czasie rzeczywistym za każdym razem jak użytkownik zmieni wartości na suwakach lub włączy ruch automatyczny.

Regulacja prędkości

Użytkownik ma do wyboru trzy prędkości ruchu manipulatora: wolną, średnią i szybką. Powodują one podzielenie drogi jaką mniejsze części do których manipulator po kolei się przemieszcza. Między każdym z tych fragmentów następuje krótka przerwa w ruchu więc podzielenie na większą liczbę spowoduje wydłużenie czasu przebycia całej drogi. Dla poszczególnych prędkości liczba odcinków jest ustalona:
- dla ruchu wolnego trasa jest dzielona na piętnaście części
- dla ruchu średniego trasa jest dzielona na pięć części
- dla ruchu szybkiiego trasa jest dzielona na dwie części

Realizacja założeń projektowych

Udało się osiągnąć główny cel projektu. Stworzono oprogramowanie, przy pomocy którego można sterować manipulatorem. Udało się zamodelować ruch robota i zwizualizować go na panelu sterowania. Także udało się stworzyć możliwość tworzenia kolejki kroków, które robot powtarza. Nie udało się jednak dokonać analizy dokładności ruchu. Wynikało to , ze tego, że aby badać dokładność potrzebny jest zewnętrzny dokładniejszy, system kontroli ruchu niż ten występujący w samych serwomechanizmach. Nie udało się też przeprowadzić analizy wpływu trybów prędkości na rzeczywisty czas ruchu.

Plany na przyszłość

Następnym krokiem w rozwoju projektu, byłoby spełnienie reszty założeń projektu. Analizę dokładności robota, można by przeprowadzić dodając wizyjny układ kontroli ruchu. Robot zyskałby wtedy znaczniki, które następnie byłby wykrywane, przez dwie prostopadle ułożone kamery. Znaczniki znajdowałby się w punktach *(x_1, y_1, z_1)* i

(x_2, y_2, z_2). Kolejnym etapem mogłoby być poprawienie szkieletu robota, tak aby zamienić w nim połączenia gwintowe, na połączenia z wykorzystaniem łożysk. Kolejnym etapem mogłoby być danie możliwości sterowania robotem przy pomocy suwaków, jednak nie w układzie biegunowy, a w układzie kartezjańskim. Wymagałoby to przeprowadzenia obliczeń kinematyki odwrotnej manipulatora.

LITERATURA

[1] The MathWorks., Pakiet wsparcia MATLAB dla dokumentacji sprzętu Arduino , MATLAB Support Package for Arduino Hardware , 2022

[2] Jakub Kałka., Maciej Figiel., Oskar Pacelt., Marcin Lesiński., Aleksandra Kempińska., Anna Wieczorek., Sandra Marcinkowska., Mateusz Mróz., Botland, 2022

[3] arduino.cc, 2022

[4] kodfilemon@github., kodfilemon.blogspot.com, 2022

[5] All Abour Circuits, 2022

[6] Damian Szymański., FORBOT, 2022

[7] mecademic.com

[8] Teresa Zielińska, Maszyny kroczące (2014)