**PROJEKT INŻYNIERSKI**

Analiza błędów pomiaru położenia platformy mobilnej

**Daniel CHYDZIŃSKI**

**Nr albumu: 296781**

**Kierunek:** Automatyka i Robotyka

**Specjalność:** Automatyka Procesowa

**PROWADZĄCY PRACĘ**

**dr Aleksander Staszulonek**

**KATEDRA Automatyki i Robotyki**

**Wydział Automatyki, Elektroniki i Informatyki**

**Gliwice 2024**

**Tytuł pracy**

Analiza błędów pomiaru położenia platformy mobilnej

**Streszczenie**

Projekt, wykonanie i testy platformy mobilnej opartej na mikroprocesorze, silnikach prądu stałego z enkoderami i regulatorach PID.

**Słowa kluczowe**

Silnik, druk 3D, regulator PID, mikroprocesor

**Thesis title**

Mobile platform position errors analysis

**Abstract**

Design, development and testing of a mobile platform based on a microprocessor, direct current motors with encoders and PID controllers.

**Key words**

Motor, 3D printing, PID controller, microprocessor

**Spis treści**

[**1 Wstęp** **1**](#_Toc15336)

[**2 Analiza tematu** **3**](#_Toc15337)

[2.1 Detekcja położenia wału silnika 3](#_Toc15338)

[2.2 Regulacja sygnału sterującego 6](#_Toc15339)

[**3 Założenia i narzędzia** **9**](#_Toc15340)

[3.1 Założenia 9](#_Toc15341)

[3.2 Narzędzia 12](#_Toc15342)

[**4 Projekt i wykonanie** **15**](#_Toc15343)

[**5 Eksperyment** **17**](#_Toc15344)

[**6 Podsumowanie i wnioski** **19**](#_Toc15345)

[**Bibliografia** **22**](#_Toc15346)

[**Spis skrótów i symboli** **25**](#_Toc15347)

[**Źródła** **27**](#_Toc15348)

[**Lista dodatkowych plików, uzupełniających tekst pracy** **29**](#_Toc15349)

[**Spis rysunków** **31**](#_Toc15350)

[**Spis tabel** **33**](#_Toc15351)

**Rozdział 1**

# Wstęp

Robotyka to obszar badawczy i techniczny poświęcony teorii, konstrukcji oraz praktycznym zastosowaniom robotów. Elementami wykonawczymi układów zrobotyzowanych są najczęściej silniki lub siłowniki, te drugie nierzadko napędzane wewnętrznie silnikami. Rodzajem maszyny zamieniającej jeden rodzaj energii — w robotyce najczęściej elektryczną — na energię mechaniczną, czego celem jest wprawienie w ruch elementów ruchomych.

W przypadku prostych układów (Definicja 1), takich jak podajnik taśmowy napędzany pojedynczym silnikiem, dokładność (Definicja 3) sterowania nie ma wysokiego priorytetu. Najważniejsze jest, by element znajdujący się na taśmie przejechał z punktu A do punktu B z pewną prędkością, a jego położeniem zajmą się inne czujniki. Jednak gdy silnik napędza ramię robota, pojazd lub drona, ważne jest, by utrzymywał stałą prędkość i/lub wykonywał określoną ilość obrotów.

**Definicja 1** (Układ sterowania)**.** *Układ służący do kontroli pożądanego urządzenia przy pomocy wybranego zasobu narzędzi, w tym zamkniętej pętli z ujemnym sprzężeniem zwrotnym (Definicja 2).*

**Definicja 2** (Pętla)**.** *Typ struktury układu umożliwiający mu reakcję na informację zwrotną o jego stanie (sprzężenie zwrotne) pochodzącą z czujników.*

**Definicja 3** (Dokładność)**.** *Stopień zgodności pomiędzy rzeczywistymi wartościami a wartościami określonymi lub mierzonymi.*

Z tego powodu, jednym z wyzwań, z jakimi mierzyli się pionierzy automatycy-robotycy jest dokładne sterowanie tworzonymi przez siebie układami. Jest to kwestia o tyle istotna, że gdy odpowiedź układu odbiega — nawet w niewielkim stopniu — od wartości zadanej, staje się on znacząco trudniejszy w użytkowaniu (sterowaniu), a w skrajnych przypadkach bezużyteczny. W przypadku ramienia robota, brak dokładności sterowania doprowadzi układ do osiągnięcia innej lokalizacji niż pożądana. W przypadku drona, prawdopodobnie całkowicie uniemożliwi lot z powodu nierówności sił nośnych. Zaś w przypadku pojazdów autonomicznych, przejechanie odległości innej niż zadana przez operatora. Jako rozwiązanie tego problemu, powstał poddział robotyki zwany odometrią (Definicja 4).

**Definicja 4** (Odometria)**.** *Dział nauk technicznych na pograniczu robotyki i miernictwa, zajmujący się użyciem różnego rodzaju czujników w celu oszacowania położenia ruchomego układu względem pozycji startowej w przestrzeni fizycznej.*

Współcześni automatycy-robotycy będący na początku swojej ścieżki edukacji/kariery, lub zajmujący się robotyką jedynie hobbystycznie, również mierzą się prędzej czy później z problemem dokładności sterowania układu. W obecnych czasach istnieje wiele sposobów rozwiązania go.

Celem niniejszej pracy inżynierskiej jest zaprojektowanie i wykonanie fizycznego modelu platformy mobilnej (zwanej dalej pojazdem) oraz aplikacji do jej kontroli. Następnie zastosowanie i przetestowanie eksperymentalne wybranej metody zwiększenia dokładności układu z użyciem ujemnego sprzężenia zwrotnego. Na końcu otrzymane wyniki zostaną przeanalizowane pod kątem użyteczności i skuteczności wybranej metody w warunkach rzeczywistych.

Praca podzielona została na następujące rozdziały[16]:

1. Wstęp — wprowadzenie do tematu, krótkie omówienie istoty problemu, zakres pracy, opis rozdziałów.
2. Analiza tematu — omówienie tematu w kontekście aktualnego stanu wiedzy o poruszanym problemie (ang. *state of art*), oraz szczegółowe jego sformułowanie.
3. Założenia i narzędzia — opis wymagań postawionych przy tworzeniu projektu wraz z uzasadnieniem wyboru, a także lista użytych narzędzi.
4. Projekt i wykonanie — szczegółowe omówienie sposobu wykonania modelu pojazdu, układu elektronicznego, aplikacji, kodu mikroprocesora i narzędzi pobocznych. Każdemu elementowi odpowiada podrozdział.
5. Eksperyment i weryfikacja — zastosowane metody badawcze i wykonane eksperymenty.
6. Podsumowanie i wnioski — skomentowanie uzyskanych wyników pod kątem osiągnięcia założonych celów, analiza i dalsze kroki.

**Rozdział 2**

# Analiza tematu

## 2.1 Detekcja położenia wału silnika

Problem synchronizowania silników elektrycznych znany jest w dziedzinie automatyki od dziesiątek lat. Początki badania silników sięgają XIX wieku, kiedy to Michael Faraday oraz inni naukowcy eksperymentowali ze wykorzystaniem elektromagnetyzmu[2]. Pierwsze silniki elektryczne były prymitywne i nie miały zaawansowanych metod sterowania. Wczesne próby pozycjonowania opierały się głównie na prostych mechanizmach, takich jak przekładnie i sprzęgła.

W miarę postępu technologicznego, szczególnie w XX wieku, rozwijano bardziej zaawansowane metody pozycjonowania. Pojawiły się pierwsze systemy sterowania, wykorzystujące technologię zwrotną informacji, mającą na celu monitorowanie i regulację położenia wałów silników. Jednak precyzja tych rozwiązań była ograniczona, a dokładność pozycjonowania nie zawsze spełniała wymagania coraz bardziej zaawansowanych zastosowań.

Dopiero wprowadzenie enkoderów (Definicja 5) elektronicznych w latach 60. XX wieku[14] stało się przełomem.

**Definicja 5** (Enkoder obrotowy)**.** *Urządzenie, generujące sygnały elektryczne odpowiadające ruchowi obrotowemu wału silnika celem określenia jego pozycji.*

Początkowo enkodery były oparte na szczotkach stykających się z dyskiem zawierającym serię odpowiednio zakodowanych pierścieni koncentrycznych (Rysunek 2.1), wypełnionych otworami o odpowiedniej długości[9]. Są one tanie w produkcji, jednak mają swoje ograniczenia związane ze zużyciem mechanicznym elementów stykowych, niską maksymalną dozwoloną prędkością silnika i wymaganiami konserwacji. Ten typ enkoderów spotykany jest do dziś, na przykład w multimetrach cyfrowych.

Rozwój technologii przyniósł enkodery optyczne, wykorzystujące diody LED i fotodetektory. Później pojawiły się enkodery magnetyczne. To właśnie one — enkodery optyczne i magnetyczne — są do dnia dzisiejszego najczęściej spotykane i oferują najwyższą dokładność sterowania przy niskich kosztach i niewielkim stopniu skomplikowania. To właśnie na nich skupiono się w dalszej części pracy.

Enkodery można podzielić ze względu na[9]:

* Metodę używaną do odczytania pozycji: kontaktowe i bezkontaktowe.
* Rodzaj sygnału wyjściowego: pozycja absolutna lub szereg inkrementujących/dekrementujących wartości.
* Zjawisko fizyczne wykorzystane do przesłania sygnału pozycyjnego: przewodzenie elektryczne, magnetyzm, zjawiska optyczne lub pojemnościowe.

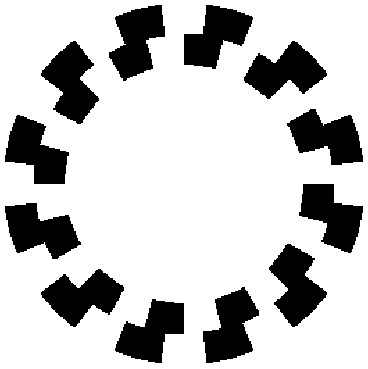
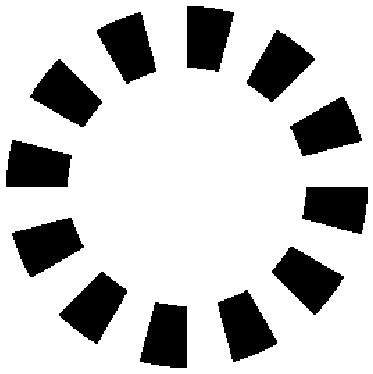
Najważniejszy jest podział ze względu na rodzaj sygnału wyjściowego. Mimo, że zarówno enkodery absolutne jak i inkrementalne posiadają dyski kodujące, różnią się one działaniem. Enkodery absolutne jako sygnał wyjściowy podają precyzyjną pozycję wału silnika, najczęściej zakodowaną w słowie bitowym. Przykładowy wygląd dysku kodującego widoczny jest na Rysunku 2.1. Istotną cechą tego rodzaju enkoderów jest możliwość określenia pozycji nawet po utracie zasilania.

Rysunek 2.1: Poglądowy schemat dysku enkodera absolutnego z 3-bitowym kodem Graya

(Źródło: [4])

Enkodery inkrementalne u podstaw działają w ten sam sposób, tzn. opierają się na dyskach kodujących, z tą różnicą, że nie są w stanie podać dokładnej wartości położenia. Zamiast tego, podają na wyjściu odpowiedni impuls przy obrocie w danym kierunku. Następnie w oprogramowaniu impulsy te są zliczane w celu oszacowania aktualnej pozycji względem pozycji startowej. Ze względu na wyzerowanie liczby impulsów przy utracie zasilania, ten typ enkodera nie jest w stanie podać dokładnej pozycji w przypadku utraty zasilania.

Rozdział 2. Analiza tematu



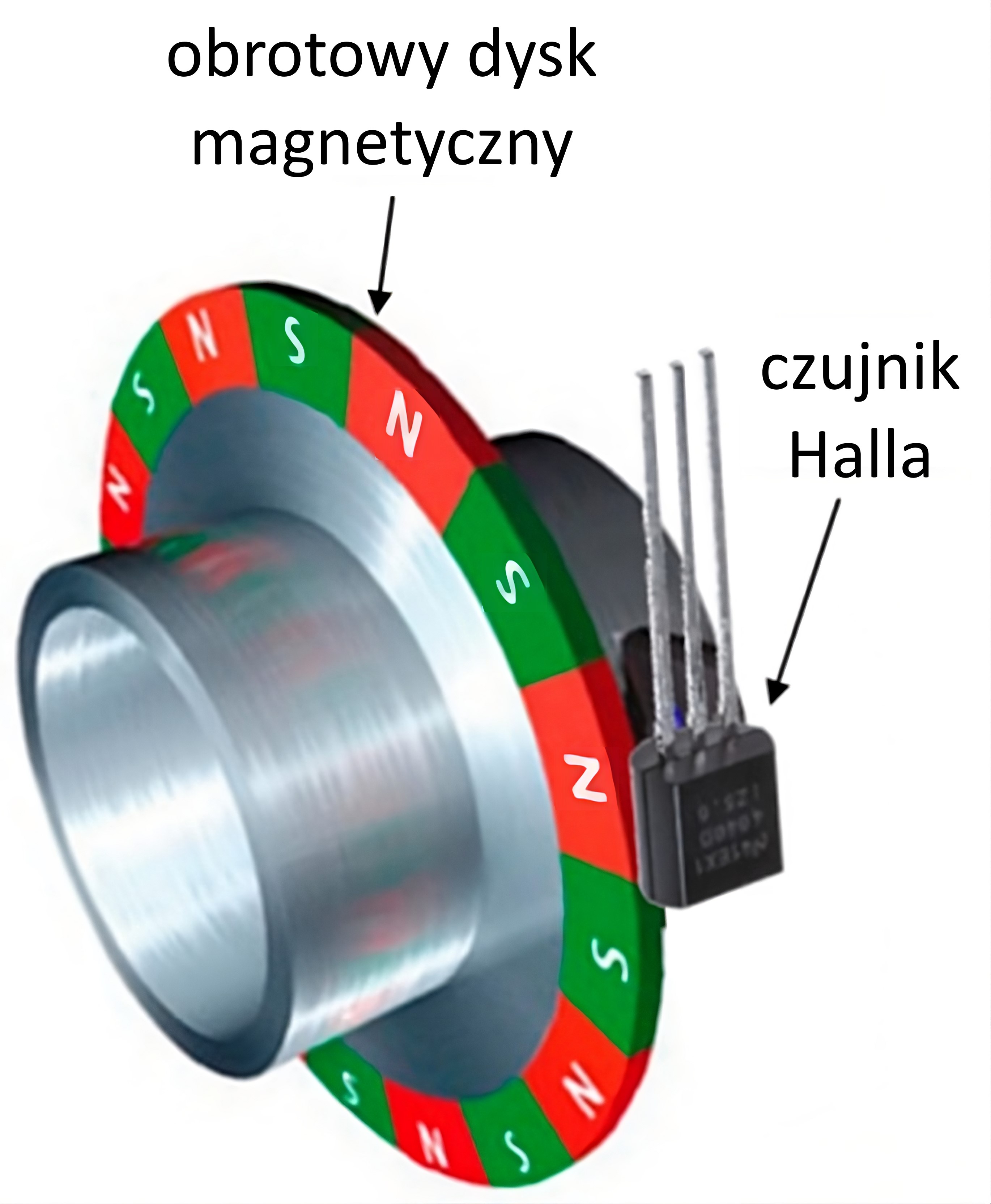
(a) Enkoder pojedynczy (b) Enkoder podwójny (kwadra-

towy)

Rysunek 2.2: Poglądowe schematy dysku enkodera inkrementalnego (Źródło: [8])

Istotny jest również podział enkoderów ze względu na wykorzystywane zjawisko fizyczne. Dwa główne typy to enkodery optyczne oraz magnetyczne. Oba rodzaje występują zarówno w wariancie pojedynczym (Rysunek 2.2a) jak i podwójnym (Rysunek 2.2b). W przypadku enkoderów optycznych, kolorowi białemu odpowiada szczelina, zaś kolorowi czarnemu blokada. W przypadku enkoderów magnetycznych, kolorom odopowiadają bieguny S i N.

Jako że w projekcie wykorzystane zostały enkodery magnetyczne, zasada ich działania została opisana, zaś enkodery optyczne pominięto. W enkoderze magnetycznym umieszcza się magnesy na obracającej się lub przemieszczającej się części obiektu — czyli dysku — a czujniki magnetyczne — najczęściej czujniki Halla (Definicja 6) — znajdujące się na stałej części enkodera rejestrują zmiany pola magnetycznego (Rysunek 2.3).



Rysunek 2.3: Zobrazowanie zasady działania enkodera magnetycznego dla tarczy z jednym rzędem (Źródło: [12])

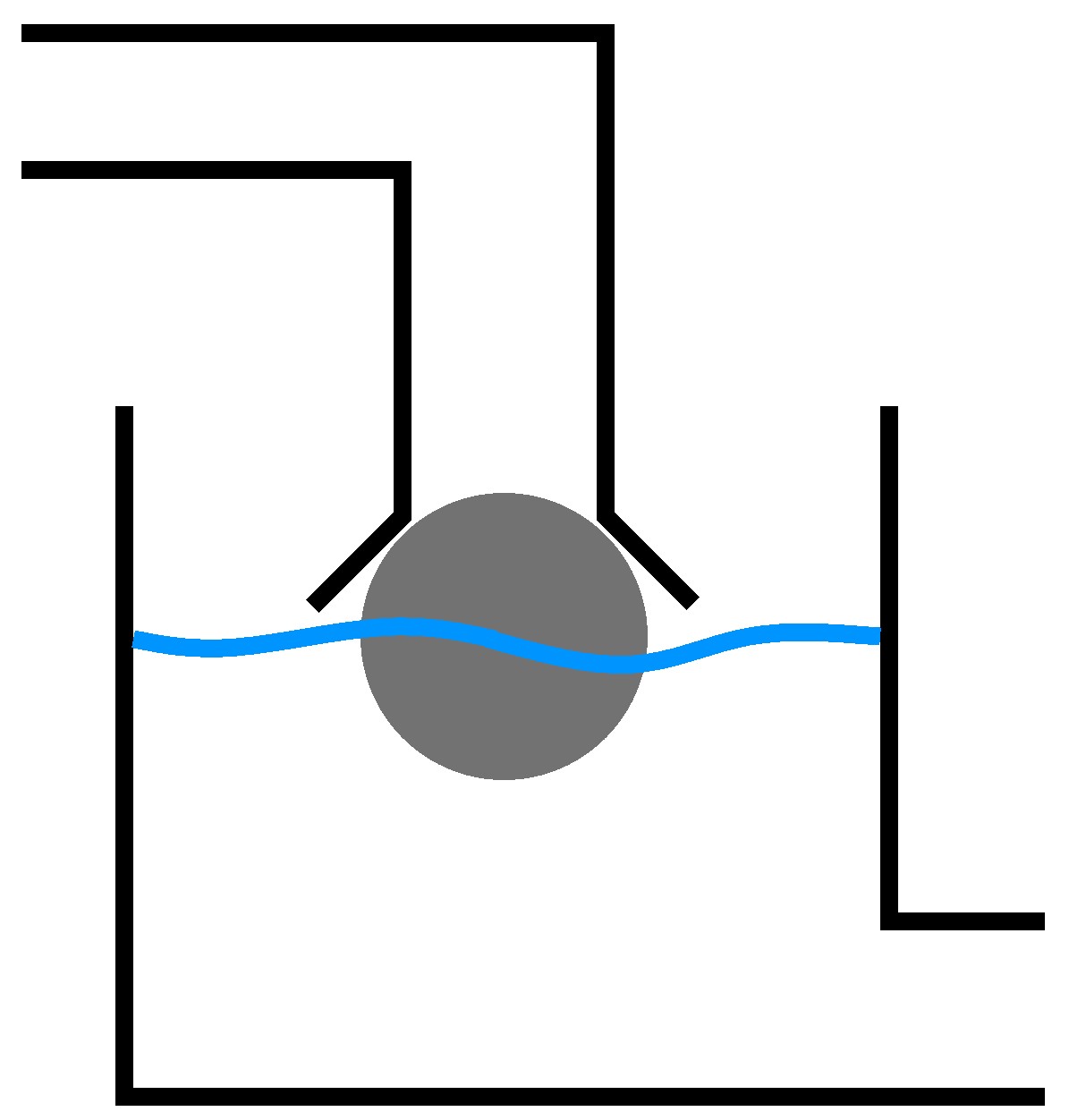
**Definicja 6** (Czujnik Halla)**.** *Czujnik pola magnetycznego i prądu, wykorzystujący zjawisko Halla.*

Te zmiany są następnie przetwarzane na sygnały elektryczne, które można interpretować jako informacje o kącie obrotu lub przemieszczeniu. Enkodery magnetyczne charakteryzują się wysoką dokładnością pomiaru, odpornością na wibracje i brakiem kontaktu mechanicznego, dzięki czemu są odporne na zabrudzenia i czynniki zewnętrzne takie jak kurz. Nie ma więc potrzeby osadzania ich w zamkniętej przestrzeni, jak to ma miejsce z czujnikami optycznymi.

## 2.2 Regulacja sygnału sterującego

Oprócz urządzenia (czujnika) dostarczającego sygnał informujący o fizycznym położeniu lub przemieszczeniu wału silnika, konieczne jest również zastosowanie metody wyznaczenia sygnału sterującego *u*.

Trudno jest określić datę powstania pierwszych regulatorów. Pierwotnie nie były one tworzone z myślą o wyznaczaniu sygnału sterującego, lecz zwyczajnie jako część urządzeń. Przykładem jednego z pierwszych regulatorów może być maszyna Ktesibiosa — wynaleziona w III wieku p.n.e.[1] — w której rolę regulatora pełnił pływak dławiący wypływającą ze źródła wodę (Rysunek 2.4). Jest to prawdopodobnie pierwszy na świecie przykład regulatora proporcjonalnego.



Rysunek 2.4: Zobrazowanie zasady działania regulatora Ktesibiosa

Na przestrzeni kolejnych tysięcy lat w technologii regulatorów nie dokonał się w zasadzie żaden znaczący postęp. Dopiero w XIX i XX wieku rozpoczęły się badania nad teorią sterowania, w których udział miało wielu naukowców z całego świata. Lata prac doprowadziły do formalnego opracowania w roku 1922 regulatora PID (ang. *Proportional–Integral–Derivative*) (Definicja 7) przez rosyjskiego naukowca, Nicolasa Minorsky’ego[7].

**Definicja 7** (Regulator PID (Proporcjonalno-Integrująco-Różniczkujący))**.** *Rrodzaj regulatora, który składa się z trzech elementów: proporcjonalnego (P), który reaguje na bieżącą wartość błędu (Definicja 8) proporcjonalnie do niego; całkującego (I), który integruje bieżący błąd w czasie i reaguje na jego sumę; oraz różniczkującego (D), które reaguje na szybkość zmiany błędu.*

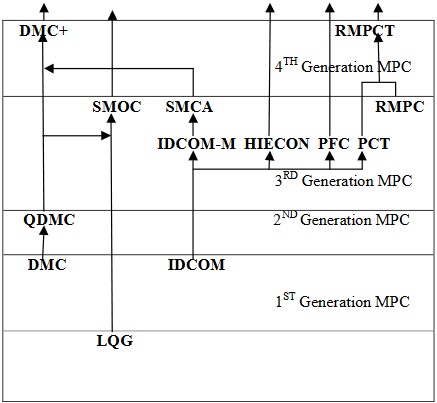
Rozdział 2. Analiza tematu

**Definicja 8** (Błąd; uchyb)**.** *W układach automatyki jest to różnica między wartością zadaną a zmierzoną wartością rzeczywistą.*

Jego popularyzacja nastąpiła w latach ’50 XX wieku, gdy układy elektroniczne stały się tańsze, dotępniejsze i bardziej niezawodne. Dziś jest powszechnie stosowany w automatyce, robotyce, elektronice i wielu innych dziedzinach inżynierii. Regulator PID jest stosowany najczęściej ze względu na prostotę, bardzo niski koszt implementacji i wystarczające działanie w większości przypadków.

Drugim najczęściej stosowanym regulatorem jest regulator predykcyjny MPC (ang. *Model Predictive Control*). W przeciwieństwie do tradycyjnych regulatorów które reagują na sprzężenie zwrotne z wyjścia układu, regulator predykcyjny działa z wyprzedzeniem, zanim zdąży nastąpić zmiana na wyjściu układu.

Pierwsze wzmianki o tym typie kontroli datowane są na wczesne lata ’60 XX wieku i rozważania Rudolfa E. Kálmána na temat systemów liniowych. Od tego czasu, technologia regulatora predykcyjnego była rozwijana niezależnie przez wiele osób i instytucji. W późnych latach ’70 XX wieku regulator ten można było już spotkać w zastosowaniach przemysłowych. W roku 1979 zaprezentowano generację pierwszą, zaś najnowszą, 4 generację stanowią w roku 1998[11]. Pełna genealogia algorytmów MPC przedstawiona została na Rysunku 2.5.



Rysunek 2.5: Genealogia algorytmów MPC (Źródło: [11])

Regulatory predykcyjne znajdują zastosowanie w układach o dużym opóźnieniu i wysokim rzędzie, gdzie regulatory PID są niewystarczające. Przykładem są sondy kosmiczne.

**Rozdział 3**

# Założenia i narzędzia

## 3.1 Założenia

Założenia podzielone zostały na kilka podsekcji, po jednej dla każdej części projektu.

Wyjaśnienie poszczególnych części/modułów znajduje się w Rozdziale 4.

### Założenia dla układu elektronicznego

* 2 silniki DC (ang. *Direct Current*)
* sterownik silników
* czujnik laserowy z przodu pojazdu
* enkodery optyczne do pomiaru położenia wałów silników
* sterowanie przy użyciu mikroprocesora
* LED (ang. *Light Emitting Diode*) sygnalizujący stan oprogramowania mikroprocesora
* głośnik sygnalizujący stan oprogramowania mikroprocesora
* bezpieczniki zabezpieczające układ elektroniczny
* przełączniki źródeł prądowych

### Założenia dla modelu pojazdu

* 4 koła
* możliwa jazda do przodu, tyłu oraz skręcanie jak w samochodzie osobowym
* kilka rozmiarów kół dla różnorodności eksperymentalnej
* modułowość pozwalająca na modyfikację w razie potrzeby
* projekt wizualny podobny do samochodu osobowego
* w miarę możliwości niska masa
* obudowa wydrukowana na drukarce 3D (ang. *3 dimensions*)

### Założenia dla oprogramowania mikroprocesora

* system FreeRTOS (ang. *Real Time Operating System*)
* wykonanie w języku C++
* klient UDP (ang. *User Datagram Protocol*)
* serwer UDP
* interpretacja danych z ramek pakietów UDP
* sterowanie możliwe w układzie otwartym lub zamkniętym
* wartość zadana odbierana z aplikacji mobilnej przez Wi-Fi (ang. *Wireless Fidelity*)
* rozpoczęcie i zakończenie pomiaru na komendę z aplikacji mobilnej
* awaryjne zakończenie pomiaru w przypadku rozłączenia Wi-Fi
* odczytywanie kierunku i położenia enkoderów
* synchronizacja silników
* obliczanie sygnałów sterujących silników
* regulatory PID
* plik konfiguracyjny

### Założenia dla aplikacji mobilnej

* działanie na systemie Android
* prostota wykonania
* prostota użytkowania
* krótki czas tworzenia (ang. *development*)
* klient UDP

Rozdział 3. Założenia i narzędzia

* serwer UDP
* interpretacja danych z ramek pakietów UDP
* zmienny docelowy adres IP (ang. *Internet Protocol*)
* parametryzacja regulatorów PID
* ustawianie wartości zadanej
* ustawianie prędkości

### Założenia dla skryptu odbierającego dane

* działanie na systemie Windows
* wykonanie w języku Python
* serwer UDP
* interpretacja danych z ramek pakietów UDP
* zapis danych do pliku .csv (ang. *Comma-Separated Values*)
* działanie w pętli; możliwość odbioru wielu pomiarów

### Założenia dla skryptu wizualizujacego dane

* działanie na systemie windows
* odczytywanie danych z pliku .csv
* wizualizacja odczytanych danych (tworzenie wykresów)
* zapisywanie wykresów do pliku .eps (ang. *Encapsulated PostScript*)

## 3.2 Narzędzia

### Narzędzia fizyczne

#### Zestaw lutowniczy

Lutowanie to proces łączenia elementów elektronicznych przez stopienie spoiwa lutowniczego na przylegających do siebie elementach metalowych, a następnie jego zastygnięcie.

Wykorzystane zostały następujące narzędzia:

* lutownica transformatorowa 100 W (Rysunek 3.1)
* topnik w żelu
* plecionka do rozlutowywania
* cyna lutownicza bezołowiowa z 3.8% srebra
* alkohol izopropylowy 100%
* zestaw przewodów typu prototypowego (ang. *jumper wires*)



Rysunek 3.1: Lutownica transformatorowa typu B

Lutownica jest najważniejszym narzędziem użytym w projekcie.

#### Taśma izolacyjna

Czarna taśma izolacyjna służąca do izolacji elementów elektrycznych.

Rozdział 3. Założenia i narzędzia

#### Klej na gorąco

Pistolet do kleju posłużył do przytwierdzania elementów w miejscu (Rysunek 3.2).



Rysunek 3.2: Pistolet do kleju model PS-PK100

#### Multimetr

Przyrząd pomiarowy do pomiaru wielkości elektrycznych. W projekcie użyto modelu UNI-T M830BUZ (Rysunek 3.3) z funkcją mierzenia napięcia, natężenia, rezystancji i ciągłości.



Rysunek 3.3: Multimetr model UNI-T M830BUZ

#### Drukarka 3D

Na drukarce 3D (Rysunek 3.4) wydrukowano obudowę pojazdu.

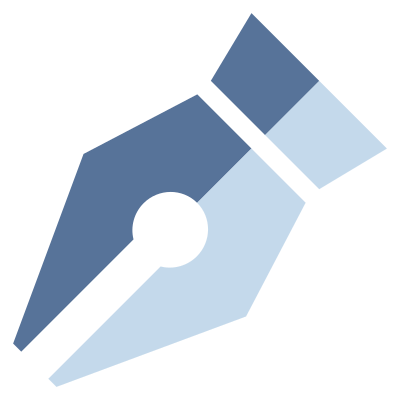


Rysunek 3.4: Drukarka 3D model Sovol SV06 (Źródło: [13])

### Oprogramowanie

#### Visual Studio Code

Do napisania oprogramowania mikroprocesora oraz pracy inżynierskiej użyta została platforma Visual Studio Code (Rysunek 3.5a), z rozszerzeniami LaTeX Workshop (Rysunek 3.5b) i PlatformIO IDE (Rysunek 3.5c).

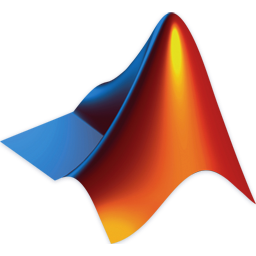


(a) Visual Studio Code (b) LaTeX Workshop (c) PlatformIO IDE

Rysunek 3.5: Logo platformy Visual Studio Code z rozszerzeniami LaTeX Workshop i PlatformIo IDE (Źródła: [6][10][15])

#### MATLAB

MATLABa użyto w celu wizualizacji danych otrzymanych z pojazdu (Rysunek 3.6).



Rysunek 3.6: Logo platformy MATLAB (Źródło: [5])

#### GitHub

GitHub posłużył jako system kontroli wersji, zarówno do kodu mikroprocesora, jak i do skryptów pomocniczych i pracy inżynierskiej (Rysunek 3.7).



Rysunek 3.7: Logo platformy GitHub (Źródło: [3])

**Rozdział 4**

# Projekt i wykonanie

Jeśli „Specyfikacja zewnętrzna”:

* wymagania sprzętowe i programowe
* sposób instalacji
* sposób aktywacji
* kategorie użytkowników
* sposób obsługi
* administracja systemem
* kwestie bezpieczeństwa
* przykład działania
* scenariusze korzystania z systemu (ilustrowane zrzutami z ekranu lub generowanymi dokumentami)

0

*.*

0

0

*.*

2

0

*.*

4

0

*.*

6

0

*.*

8

0*.*00 0*.*02 0*.*04 0*.*06 0*.*08 0*.*10 Rysunek 4.1: Podpis rysunku po rysunkiem.

**Rozdział 5**

# Eksperyment

Work in progress.

**Rozdział 6**

# Podsumowanie i wnioski

Work in progress.

# Bibliografia

1. Wielu edytorów. *Ctesibius Of Alexandria*. url: [https://www.britannica.com/ biography/Ctesibius-of-Alexandria](https://www.britannica.com/biography/Ctesibius-of-Alexandria) (term. wiz. 29.12.2023).
2. Michael Faraday. „On some new Electro-Magnetical Motions, and on the Theory of Magnetism.” W: *The Quarterly Journal of Science, Literature and The Arts* 12 (1822), s. 74–96.
3. Inc GitHub. url: <https://github.com/logos> (term. wiz. 31.12.2023).
4. jjbeard. *A Rotary Encoder Disc with a 3-Bit Binary Reflected Gray Code (BRGC).* url: [https://en.wikipedia.org/wiki/Rotary\_encoder#/media/File: Encoder\_Disc\_(3-Bit).svg](https://en.wikipedia.org/wiki/Rotary_encoder#/media/File:Encoder_Disc_(3-Bit).svg) (term. wiz. 27.12.2023).
5. MathWorks. url: [https://en.wikipedia.org/wiki/File:Matlab\_Logo.png# /media/File:Matlab\_icon.png](https://en.wikipedia.org/wiki/File:Matlab_Logo.png#/media/File:Matlab_icon.png) (term. wiz. 31.12.2023).
6. Microsoft. url: [https://en.wikipedia.org/wiki/Visual\_Studio\_Code# /media/File:Visual\_Studio\_Code\_1.35\_icon.svg](https://en.wikipedia.org/wiki/Visual_Studio_Code#/media/File:Visual_Studio_Code_1.35_icon.svg) (term. wiz. 31.12.2023).
7. Nicolas Minorsky. „DIRECTIONAL STABILITY OF AUTOMATICALLY STEERED BODIES”. W: *American Society of Naval Engineers* 34 (1922). issn: 00281425. doi: [10.1111/j.1559-3584.1922.tb04958.x.](https://doi.org/10.1111/j.1559-3584.1922.tb04958.x)
8. Piotr Mitros. *Optical Encoder Project*. url: [https://groups.csail.mit.edu/ mac/users/pmitros/encoder/](https://groups.csail.mit.edu/mac/users/pmitros/encoder/) (term. wiz. 28.12.2023).
9. University of Hawaii NASA Infrared Telescope Facility (IRTF) Institute for Astronomy. *Techniques For Digitizing Rotary and Linear Motion*. url: [http://irtfweb. ifa.hawaii.edu/~tcs3/tcs3/0306\_conceptual\_design/Docs/05\_Encoders/ encoder\_primer.pdf](http://irtfweb.ifa.hawaii.edu/~tcs3/tcs3/0306_conceptual_design/Docs/05_Encoders/encoder_primer.pdf) (term. wiz. 27.12.2023).
10. PlatformIO. url: [https://commons.wikimedia.org/wiki/File:PlatformIO\_ logo.svg](https://commons.wikimedia.org/wiki/File:PlatformIO_logo.svg) (term. wiz. 31.12.2023).
11. Ruchika i Neha Raghu. „Model Predictive Control: History and Development”. W:

*International Journal of Engineering Trends and Technology (IJETT)* 4 (2013). issn: 2231-5381.

1. Jonathan Seybold, André Bülau, Karl-Peter Fritz, Alexander Frank, Cor Scherjon, Joachim Burghartz i André Zimmermann. „Miniaturized Optical Encoder with Micro Structured Encoder Disc”. W: *Applied Sciences* 9 (2019). doi: [10.3390/](https://doi.org/10.3390/app9030452)

[app9030452.](https://doi.org/10.3390/app9030452)

1. *SV06 User Manual V1.0*. SOVOL TECHNOLOGY CO., LIMITED. Rm 23 9 F Blk G Kwai Shing Ind Bldg Stage 2 42 46 Tai Lin Pai Rd Kwai Chung NT 999077 Hong Kong (SAR), 2022.
2. C. F. Winder. „Shaft Angle Encoders”. W: *Electronic Industries* 18.10 (1959), s. 76– 80.
3. James Yu. url: [https://marketplace.visualstudio.com/items?itemName= James-Yu.latex-workshop](https://marketplace.visualstudio.com/items?itemName=James-Yu.latex-workshop) (term. wiz. 02.01.2024).
4. Politechnika Śląska. *Wymagania do pracy inżynierskiej dla kierunku Automatyka i Robotyka*. 2021. url: [https://www.polsl.pl/rau/wp-content/uploads/ sites/42/2021/11/AiR\_Wymagania\_do\_pracy\_inzynierskiej.pdf](https://www.polsl.pl/rau/wp-content/uploads/sites/42/2021/11/AiR_Wymagania_do_pracy_inzynierskiej.pdf) (term. wiz.

27.12.2023).

**Dodatki**

# Spis skrótów i symboli

*u* sygnał sterujący obiektem wykonawczym w układzie sterowania/regulacji

PID regulator Proporcjonalno-Całkująco-Różniczkujący (ang. *Proportional–Integral–Derivative*)

MPC regulator oparty na modelu predykcyjnym (ang. *Model Predictive Control*)

DC prąd stały (ang. *Direct Current*)

LED dioda emitująca światło (ang. *Light Emitting Diode*)

3D 3 wymiary (ang. *3 dimensions*)

Wi-Fi sieć bezprzewodowa (ang. *Wireless Fidelity*)

RTOS system czasu rzeczywistego (ang. *Real Time Operating System*)

# Źródła

Work in progress.

# Lista dodatkowych plików, uzupełniających tekst pracy

Work in progress.

# Spis rysunków

2.1 Poglądowy schemat dysku enkodera absolutnego z 3-bitowym kodem Graya

(Źródło: [4]) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

2.2 Poglądowe schematy dysku enkodera inkrementalnego (Źródło: [8]) . . . . . 5

2.3 Zobrazowanie zasady działania enkodera magnetycznego dla tarczy z jed-

nym rzędem (Źródło: [12]) . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

2.4 Zobrazowanie zasady działania regulatora Ktesibiosa . . . . . . . . . . . . 6

2.5 Genealogia algorytmów MPC (Źródło: [11]) . . . . . . . . . . . . . . . . . 7

3.1 Lutownica transformatorowa typu B . . . . . . . . . . . . . . . . . . . . . 12

3.2 Pistolet do kleju model PS-PK100 . . . . . . . . . . . . . . . . . . . . . . . 13

3.3 Multimetr model UNI-T M830BUZ . . . . . . . . . . . . . . . . . . . . . . 13

3.4 Drukarka 3D model Sovol SV06 (Źródło: [13]) . . . . . . . . . . . . . . . . 13

3.5 Logo platformy Visual Studio Code z rozszerzeniami LaTeX Workshop

i PlatformIo IDE (Źródła: [6][10][15]) . . . . . . . . . . . . . . . . . . . . . 14

3.6 Logo platformy MATLAB (Źródło: [5]) . . . . . . . . . . . . . . . . . . . . 14

3.7 Logo platformy GitHub (Źródło: [3]) . . . . . . . . . . . . . . . . . . . . . 14

4.1 Podpis rysunku po rysunkiem. . . . . . . . . . . . . . . . . . . . . . . . . . 16

# Spis tabel