

PORÓWNANIE ALGORYTMÓW NAWIGACYJNYCH

Michał Bazan

Opiekun pracy:

dr inż. Dariusz Rzońca

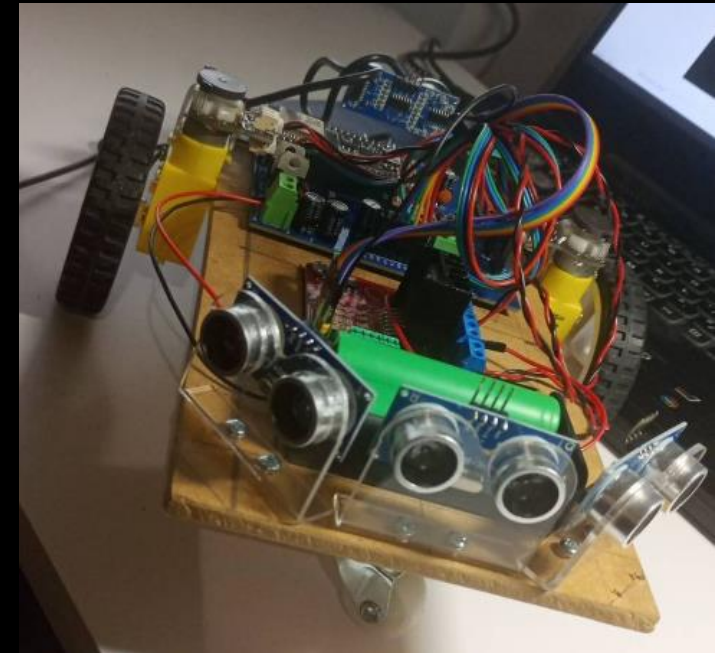
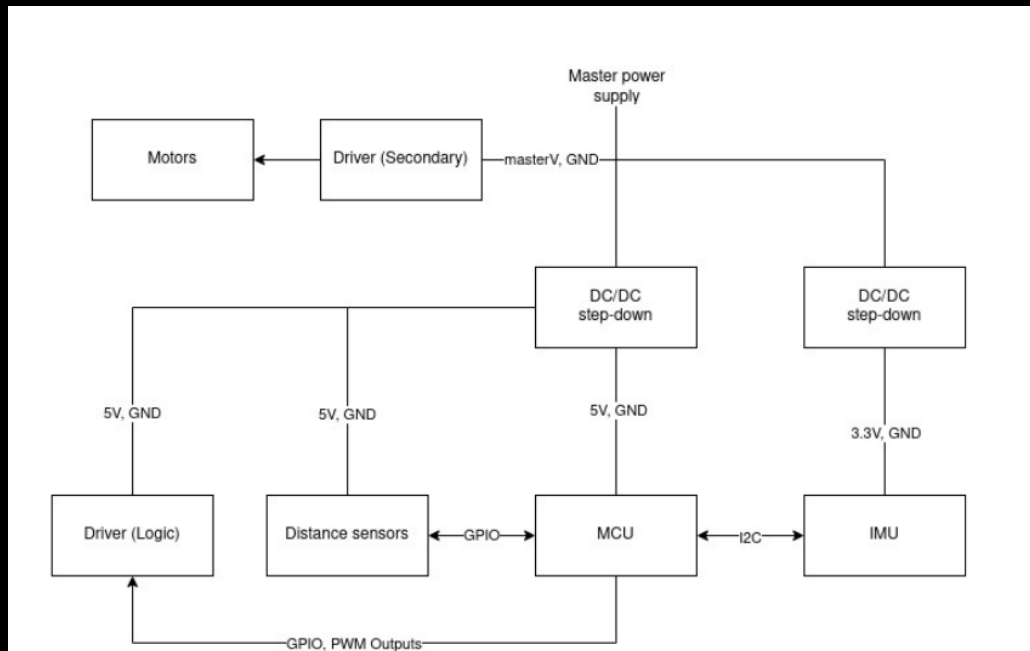
Cel pracy

- Badanie heurystycznych metod optymalizacji nastaw PID i porównanie z metodami klasycznymi
 - Badanie i porównanie statycznych algorytmów pod kątem wydajności obliczeniowej i optymalizacji długości trasy
 - Badanie i porównanie dynamicznych algorytmów pod kątem wydajności obliczeniowej i optymalizacji długości trasy
-

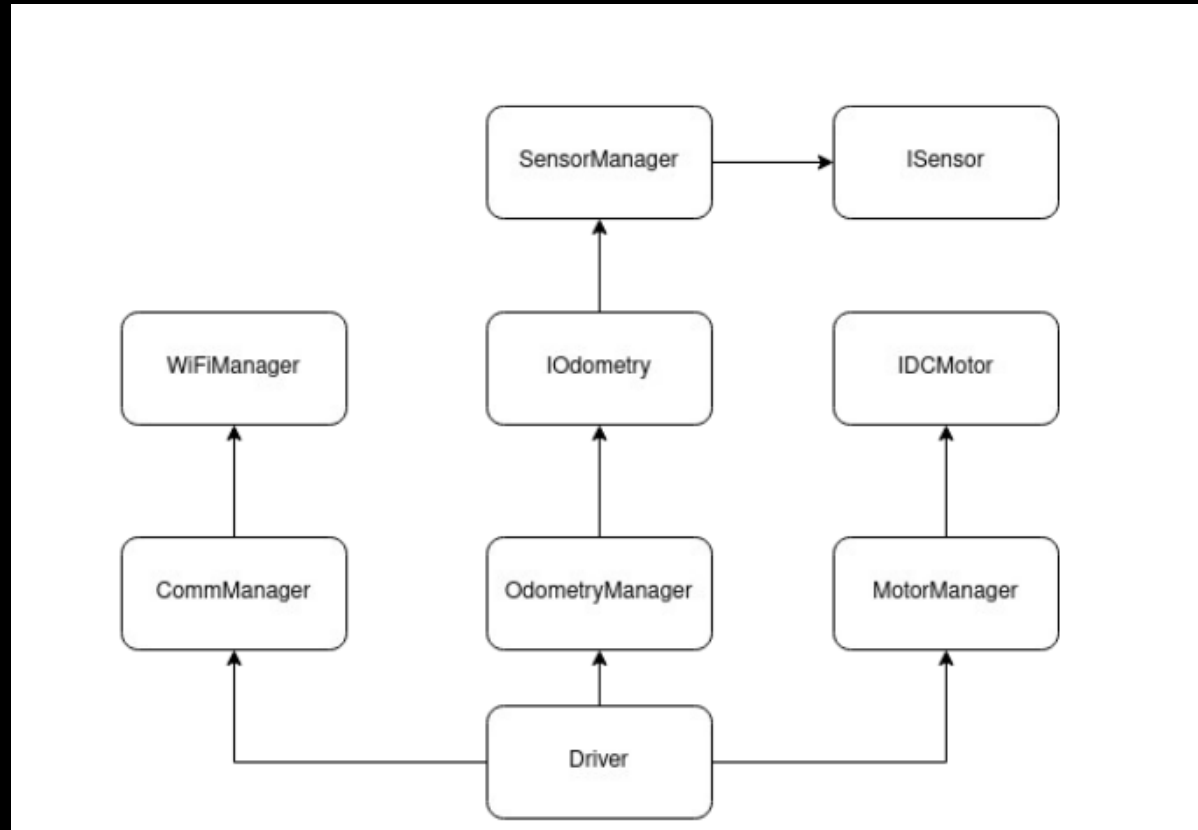
Zakres pracy

- Projekt i implementacja robota mobilnego zgodnie z procedurami ASPICE
 - Implementacja algorytmu genetycznego wyznaczającego nastawy PID, porównanie otrzymanych rezultatów z metodami klasycznymi
 - Implementacja oprogramowania testującego algorytmy statyczne i dynamiczne do wyznaczania trasy, porównanie pod kątem wydajności obliczeniowej i dokładności nawigacji
-

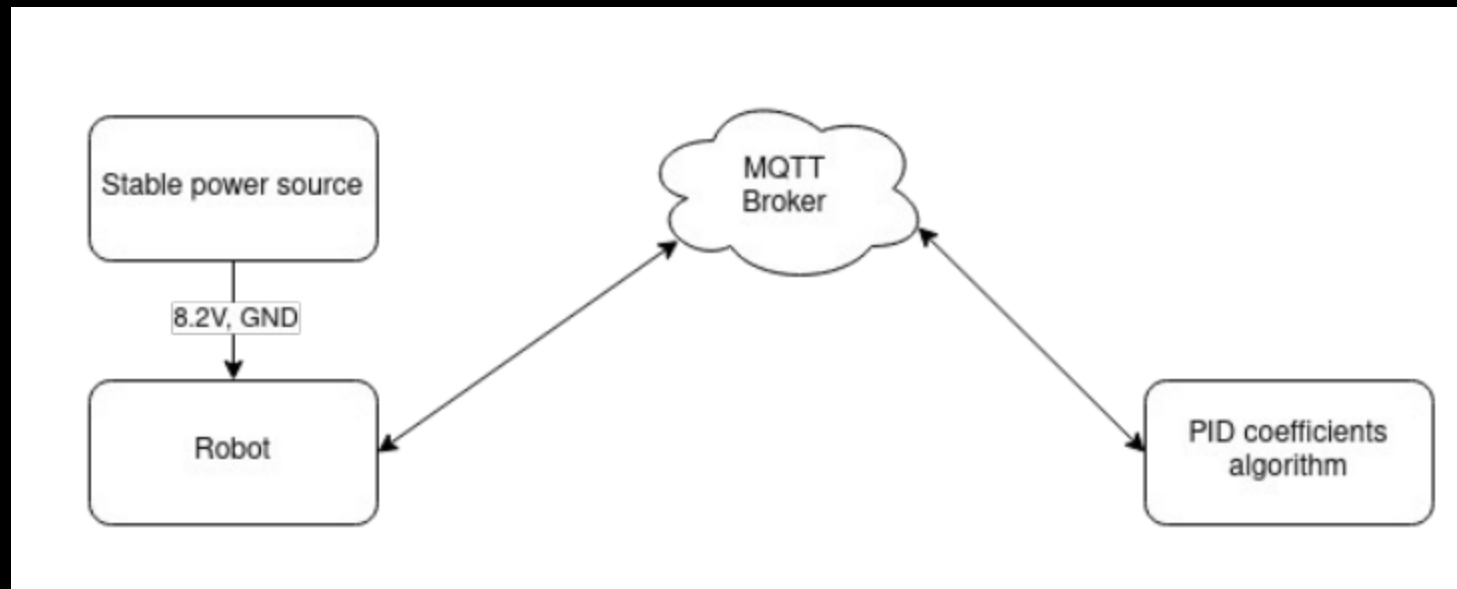
Implementacja robota mobilnego



Implementacja oprogramowania



Badanie metod heurystycznych do optymalizacji nastaw PID

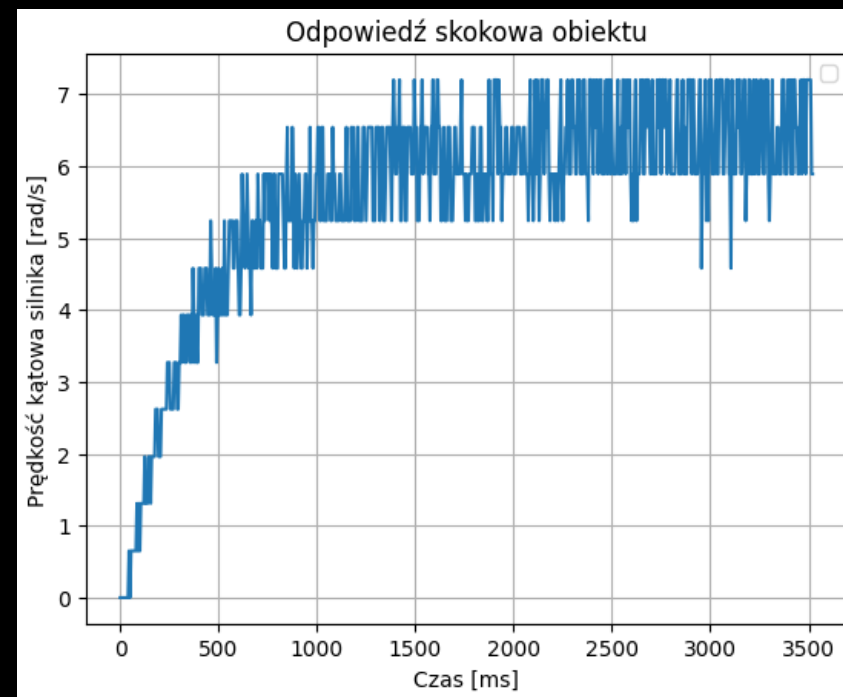


Stanowisko pomiarowe



Identyfikacja obiektu

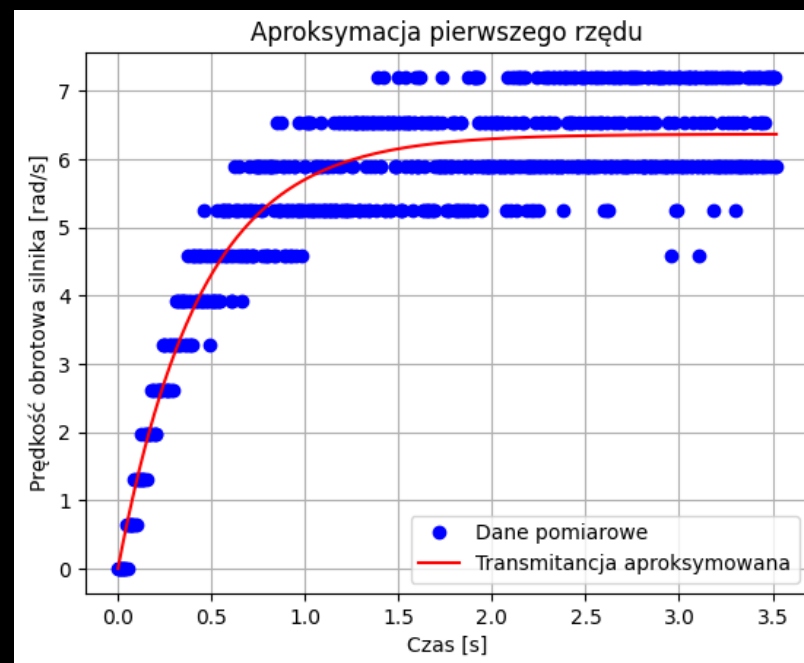
- Obiekt – silnik szczotkowy DC
- Sygnał wejściowy - PWM o wypełnieniu 50%



Identyfikacja obiektu

Aproksymacja pierwszego rzędu:

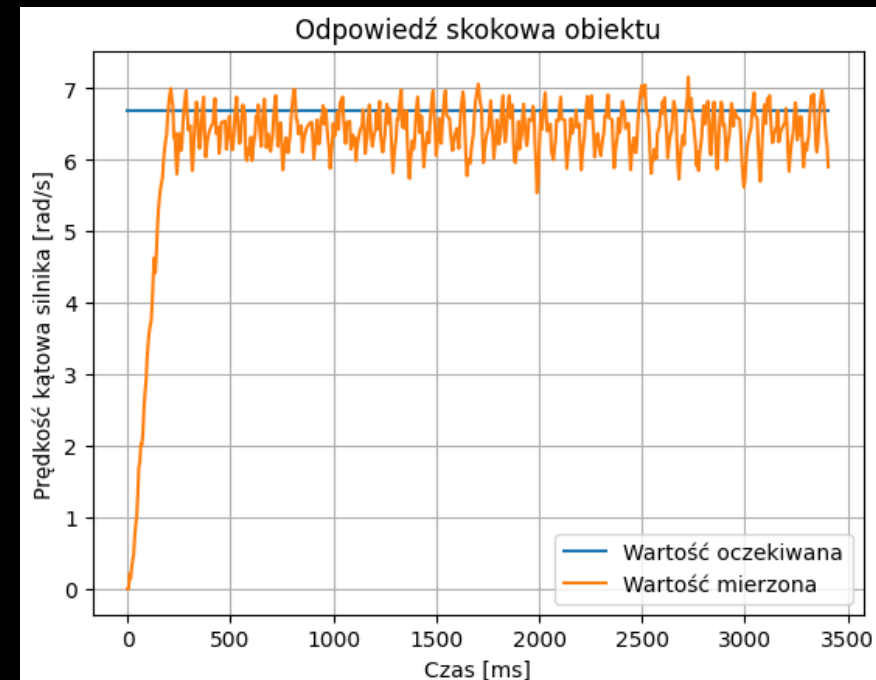
- Wzmocnienie (K): 6.36
- Stała czasowa (T): 0.44



Wyznaczenie nastaw metodą Zieglera-Nicholsa

$K_p = 0.04$, $T_i = 0.695$, $T_d = 0.166$

Średni błąd względny regulacji: 7.43%



Opis eksperymentu – algorytm genetyczny

Algorytm został uruchomiony 3 razy z różnym zestawem parametrów:

- Populacja początkowa 100
- Populacja początkowa 200
- Populacja początkowa 300

Pozostałe parametry wspólne dla każdego eksperymentu:

- Maksymalna ilość generacji 25
- Prawdopodobieństwo krzyżowania 70%
- Prawdopodobieństwo mutacji 15%

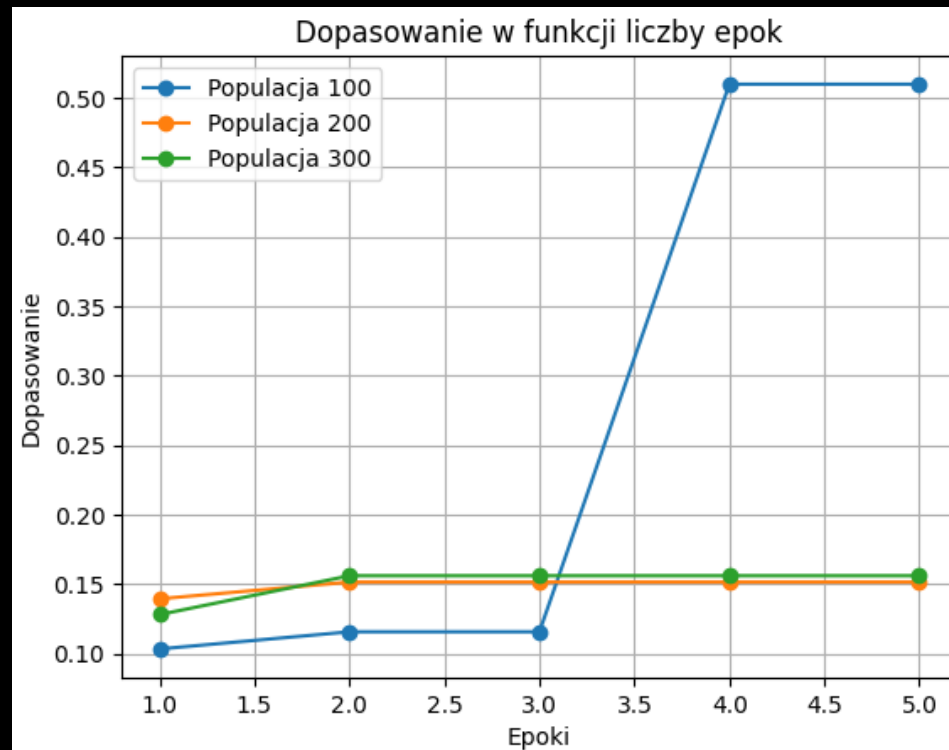
Kryterium stopu:

Średni błąd regulacji najbliższy 0.

Algorytm **maksymalizuje** funkcję celu, dlatego wzór opisujący fitness danego osobnika został określony przez:

$$\frac{1}{\text{mean relative error}}$$

Wyniki działania algorytmu



Wyniki działania algorytmu

a) $K_p = 6.48$, $T_i = 2.93$, $T_d = 2.34$

Średni błąd względny regulacji: 1.96%

b) $K_p = 29.50$, $T_i = 14.54$, $T_d = 1.37$

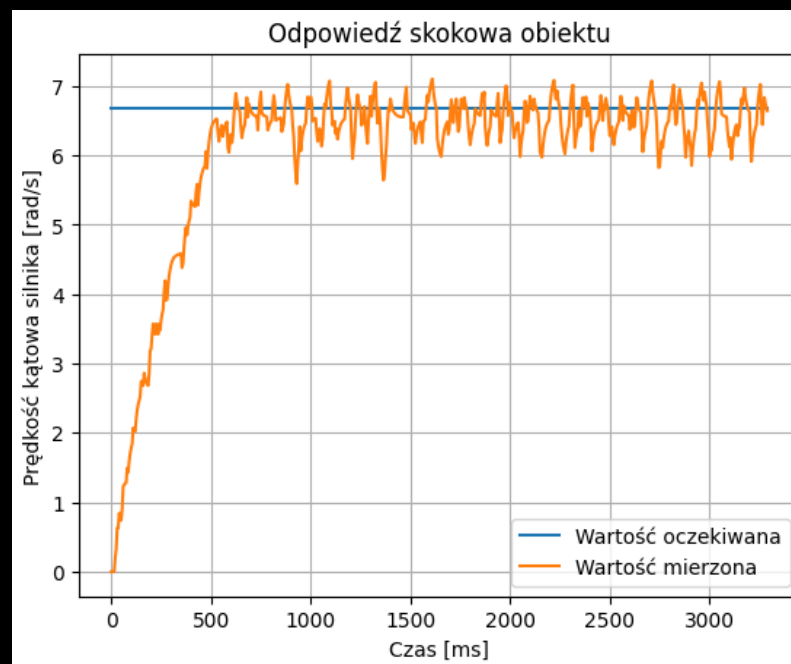
Średni błąd względny regulacji: 6,60%

c) $K_p = 11.98$, $T_i = 1.59$, $T_d = 1.34$

Średni błąd względny regulacji: 6,41%

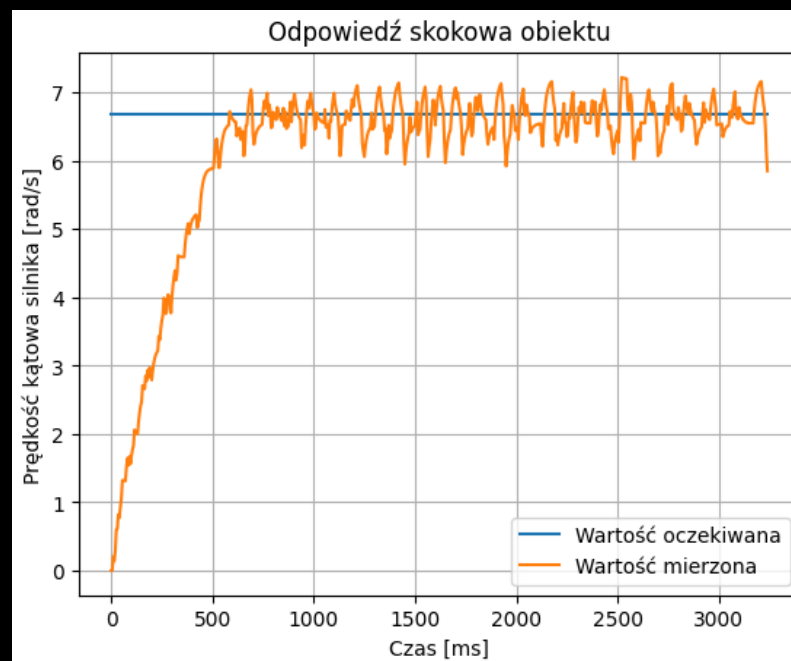
Wyniki działania algorytmu

$K_p = 6.48$, $T_i = 2.93$, $T_d = 2.34$



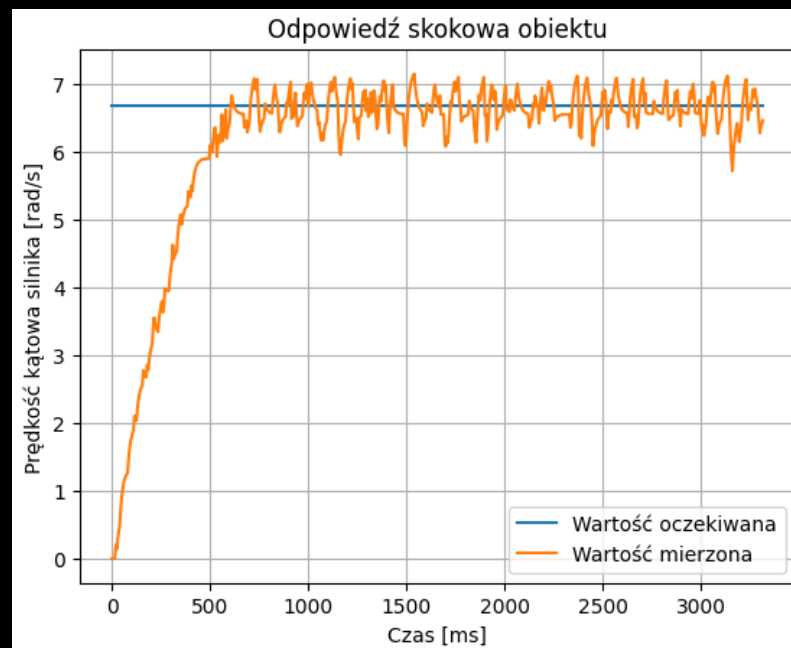
Wyniki działania algorytmu

$K_p = 29.50$, $T_i = 14.54$, $T_d = 1.37$



Wyniki działania algorytmu

$K_p = 11.98$, $T_i = 1.59$, $T_d = 1.34$



Porównanie algorytmu genetycznego z metodą klasyczną

- a) Metoda Zieglera-Nicholsa:
 - Czas regulacji ok 192ms
 - Niedoregulowanie, zbyt delikatny regulator
 - b) Algorytm genetyczny
 - Czasy regulacji ok 600ms
 - Satysfakcjonująca regulacja
-

Porównanie algorytmu genetycznego z metodą klasyczną - wnioski

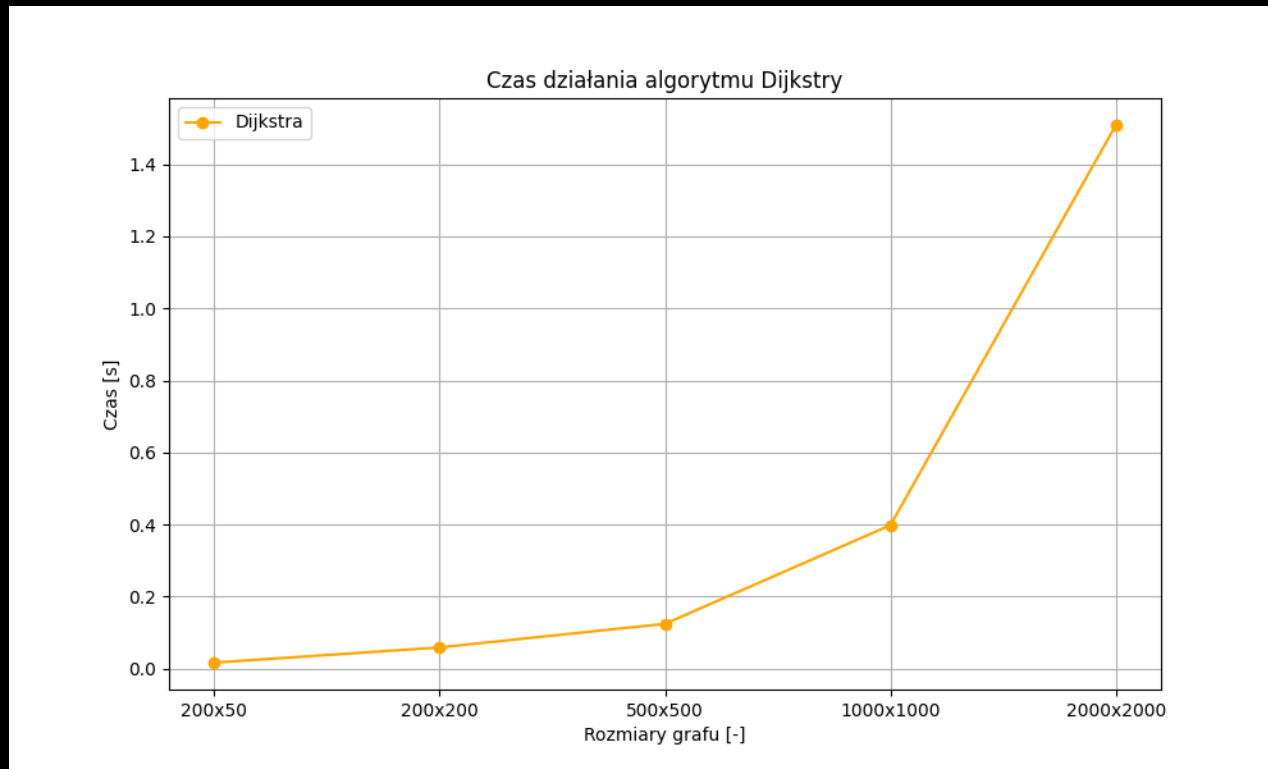
- Ze względu na prostotę obiektu jakim jest silnik DC, różne nastawy regulatora są w stanie usatysfakcjonować stawiane przed nim wymagania,
 - Nastawy wyznaczone metodą klasyczną nie satysfakcjonują potrzeb, ponieważ występuje niedoregulowanie a ponadto jeden z silników nie reaguje na wymuszenie,
 - Nastawy wyznaczone algorytmem genetycznym oferują stabilną regulację kosztem dłuższego czasu ustalania obiektu,
 - Algorytm genetyczny po kilku epokach wyznacza lokalne optimum popadając w stagnację, dlatego przy bardziej złożonych obiektach warto rozważyć modyfikację parametrów algorytmu lub ponowną inicjalizację populacji jeśli najlepsze rozwiązanie jest daleko od kryterium stopu i nie zmienia się w kolejnych generacjach.
-

Porównanie statycznych algorytmów nawigacyjnych - opis eksperymentu

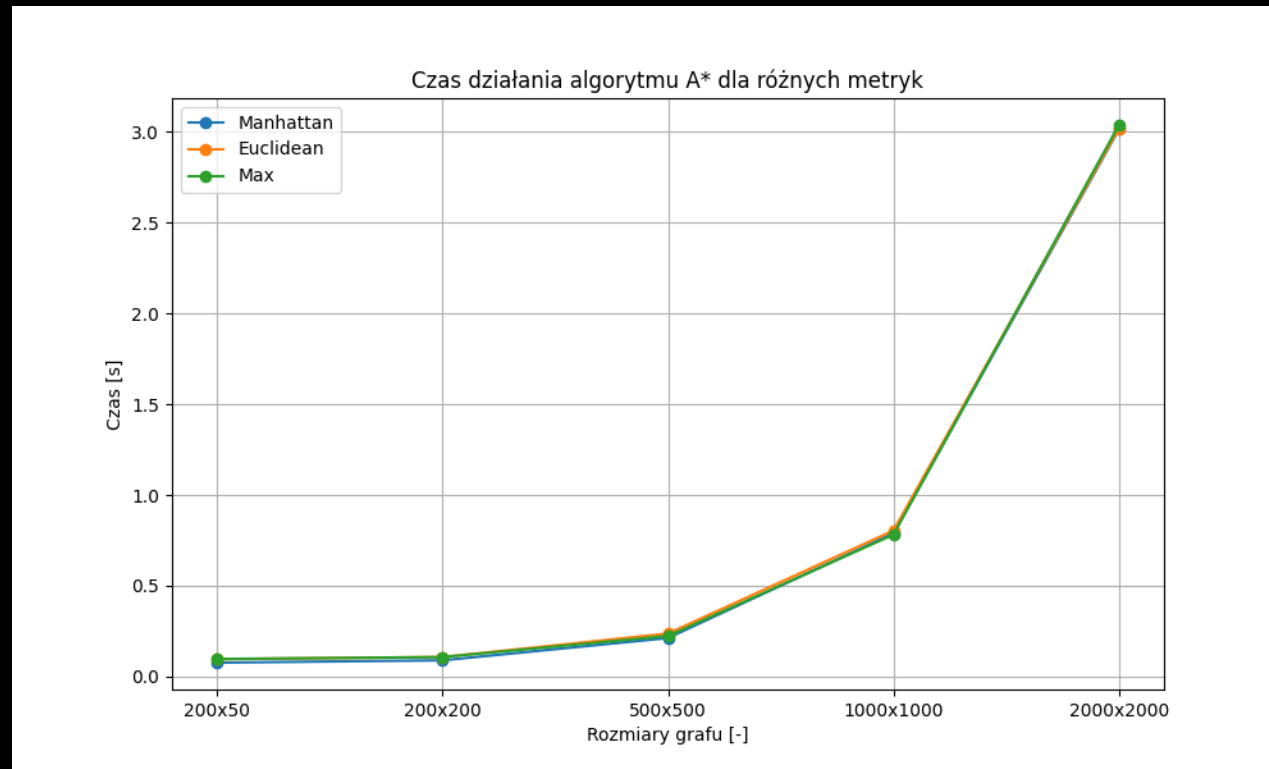
Eksperyment polegał na przetestowaniu algorytmu Dijkstry oraz A* z wybranymi metrykami (Manhattan, Euclidean, Max) na mapach i porównaniu otrzymanych wyników:

- Bez przeszkód o rozmiarach
 - 200x50
 - 200x200
 - 500x500
 - 1000x1000
 - 2000x2000
 - Z losowo umieszczonymi przeszkodami na mapie o rozmiarach 200x50
-

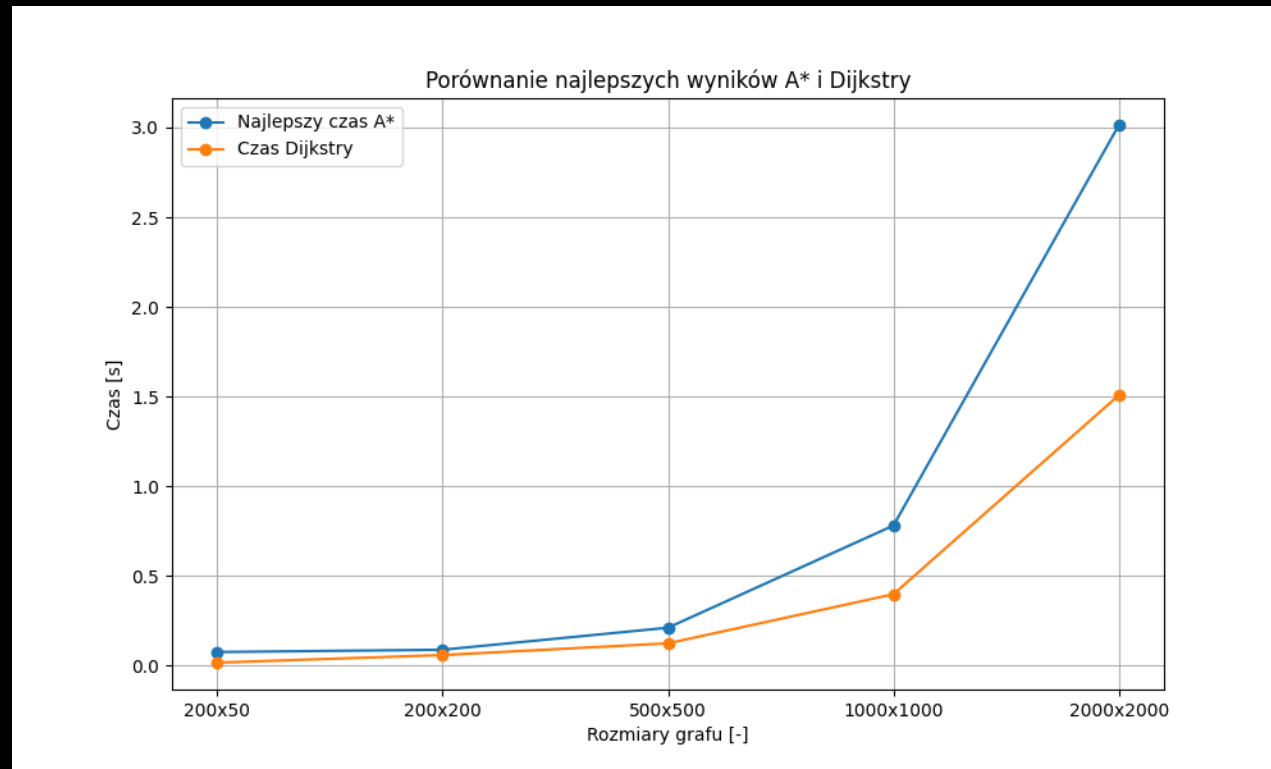
Porównanie algorytmów na mapach bez przeszkód - Dijkstra



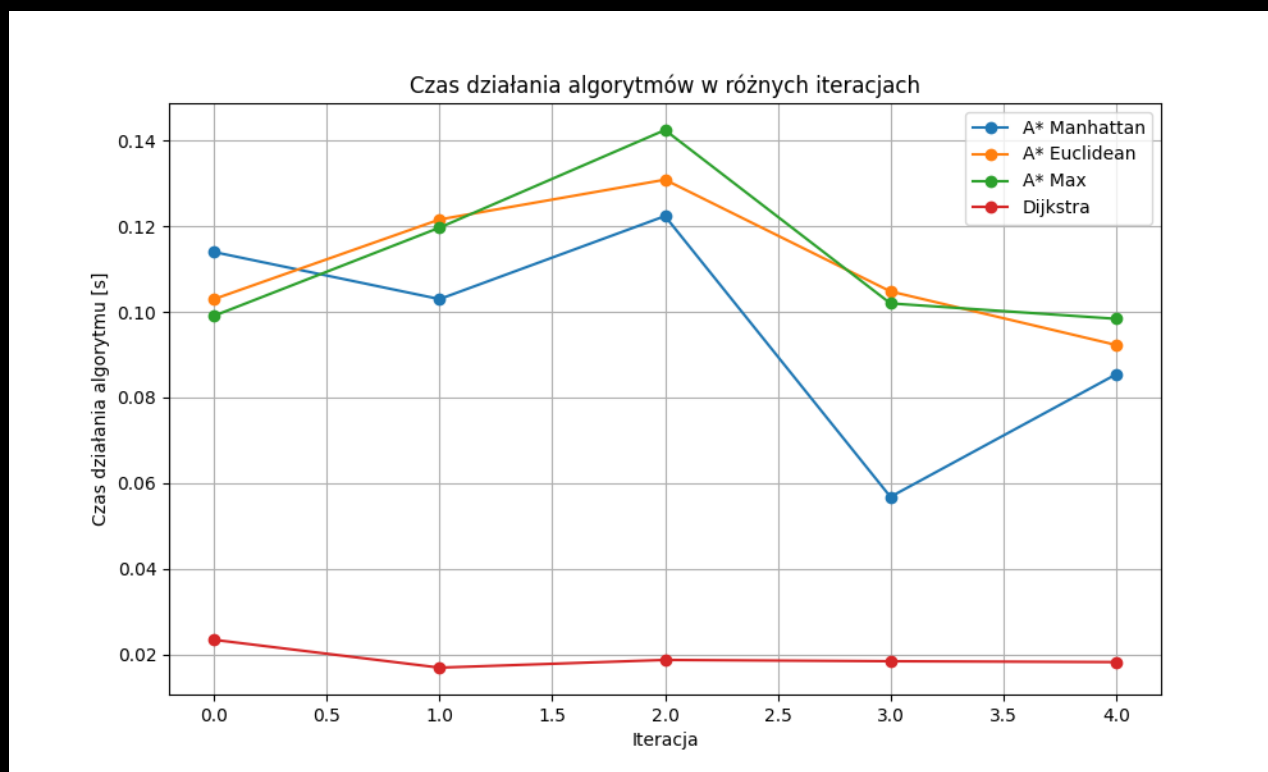
Porównanie algorytmów na mapach bez przeszkód - A*



Porównanie algorytmów na mapach bez przeszkód



Porównanie algorytmów na mapach z losowo umieszczonymi przeszkodami



Porównanie statycznych algorytmów nawigacyjnych - wnioski

- Algorytm Dijkstry wypadł lepiej pod względem czasu wykonania w porównaniu do algorytmu A*. Zwykle A* powinien działać szybciej, ponieważ używa heurystyki do prowadzenia wyszukiwania, co teoretycznie powinno ograniczyć liczbę odwiedzanych węzłów
 - Heurystyka Manhattan osiągała najkrótsze czasy wykonania dla większości badanych przypadków. Odległość manhattańska jest najprostszą heurystyką spośród używanych, ponieważ jest to suma różnic w poziomych i pionowych odległościach między punktami. To prowadzi do szybszych obliczeń w porównaniu z bardziej złożonymi heurystykami jak Euclidean czy Max.
 - Na mapach z przeszkodami Algorytmy A* z różnymi heurystykami uzyskały takie same wyniki pod względem długości trasy, jednak wymagały więcej czasu obliczeniowego w porównaniu do Dijkstry.
-

Porównanie dynamicznych algorytmów nawigacyjnych - opis eksperymentu

Eksperyment polegał na przetestowaniu algorytmu D* oraz D*-Lite na rzeczywistej mapie i porównaniu wyników w celu wybrania optymalnego algorytmu dla zaimplementowanego robota.

- Rozmiar mapy: 200x50cm
 - Wykorzystane algorytmy statyczne do wyznaczenia pierwotnej trasy: Dijkstra oraz A* z metryką Manhattan.
-

Porównanie dynamicznych algorytmów nawigacyjnych - Dijkstra jako algorytm bazowy

TODO



Porównanie dynamicznych algorytmów nawigacyjnych - A* jako algorytm bazowy

TODO

Porównanie dynamicznych algorytmów nawigacyjnych - wnioski

TODO



Wkład własny

Za wkład własny autor uważa:

- Implementację robota mobilnego
 - Implementację systemu wbudowanego robota
 - Implementację eksperymentów
-