

# PORÓWNANIE WYBRANYCH 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 (projekt, implementacja, testy)

---

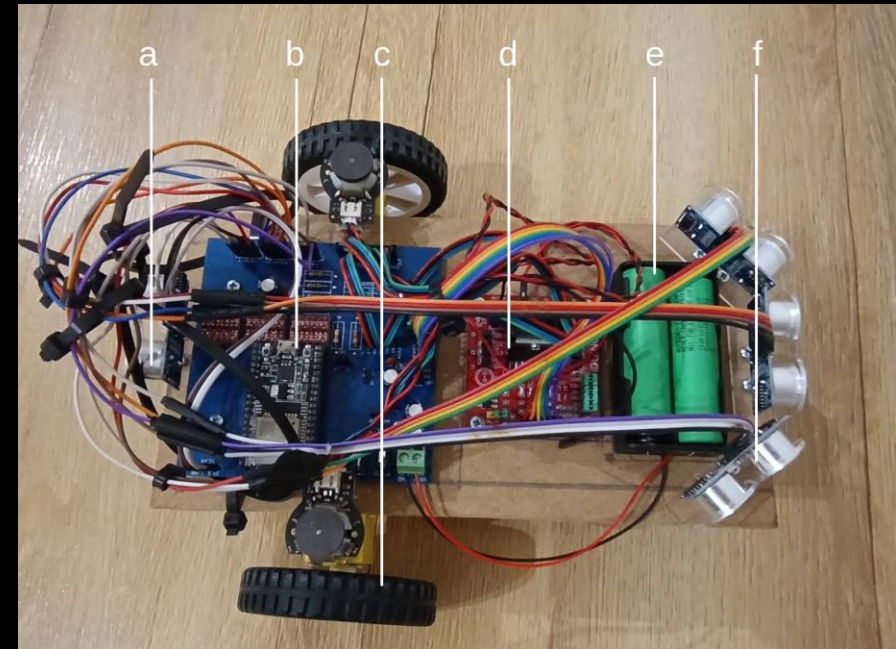
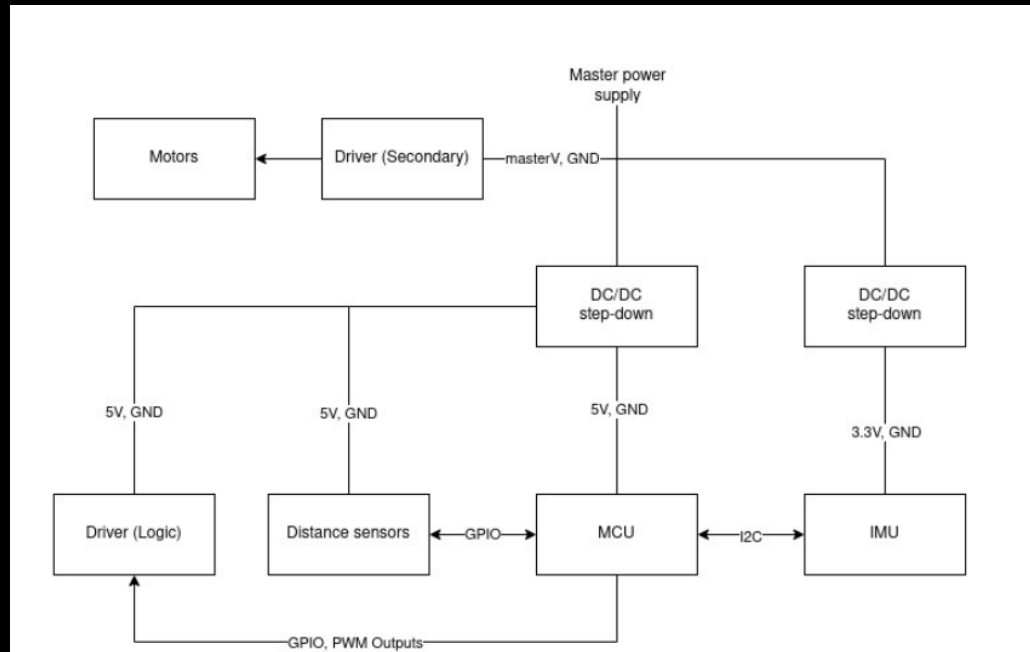
Implementacja algorytmu genetycznego  
wyznaczającego nastawy PID, porównanie  
otrzymanych rezultatów z metodami  
klasycznymi

---

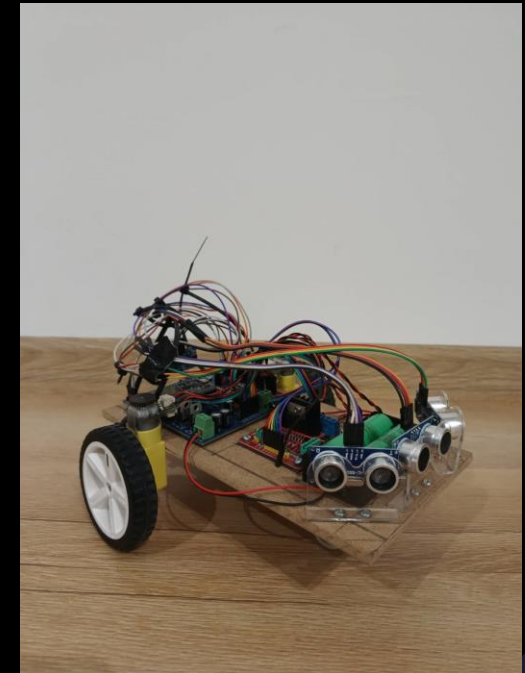
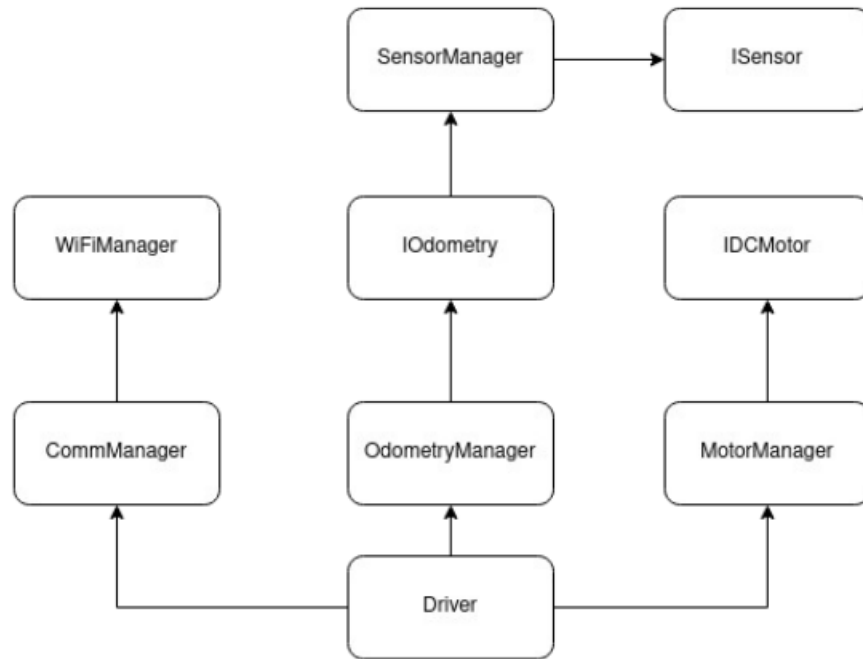
Implementacja oprogramowania  
testującego nawigacyjne algorytmy statyczne i  
dynamiczne, porównanie wyników pod kątem  
wydajności obliczeniowej i długości  
wyznaczanej trasy

---

# Implementacja robota mobilnego

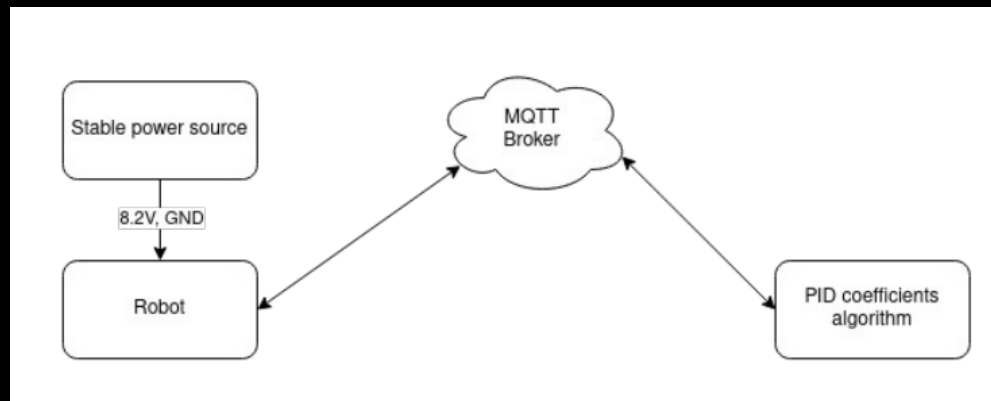


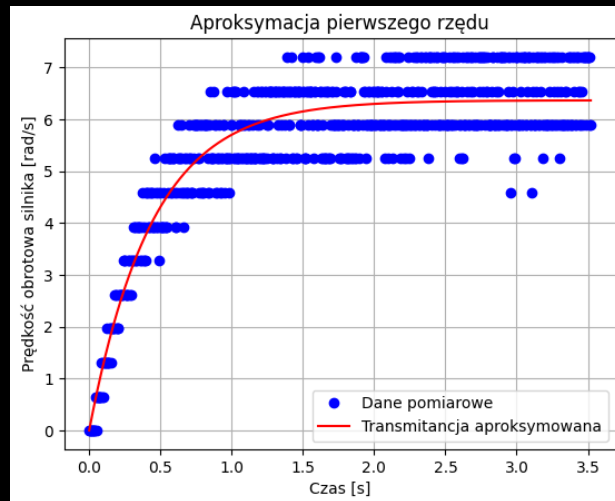
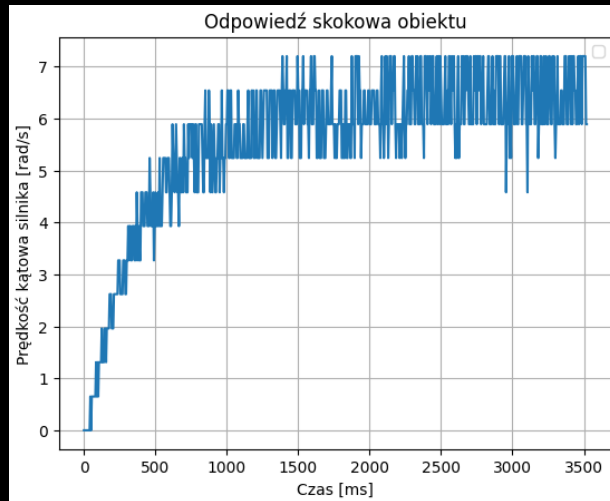
# Implementacja oprogramowania





# Badanie metod heurystycznych do optymalizacji nastaw PID - stanowisko pomiarowe





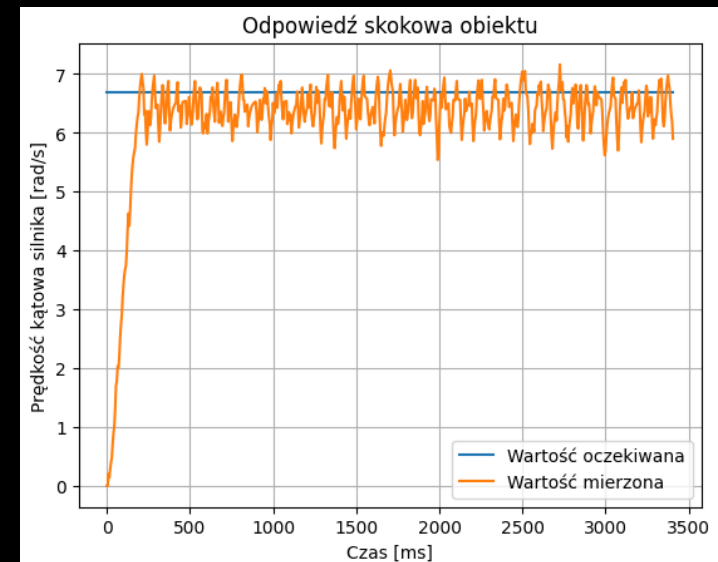
# Identyfikacja obiektu

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

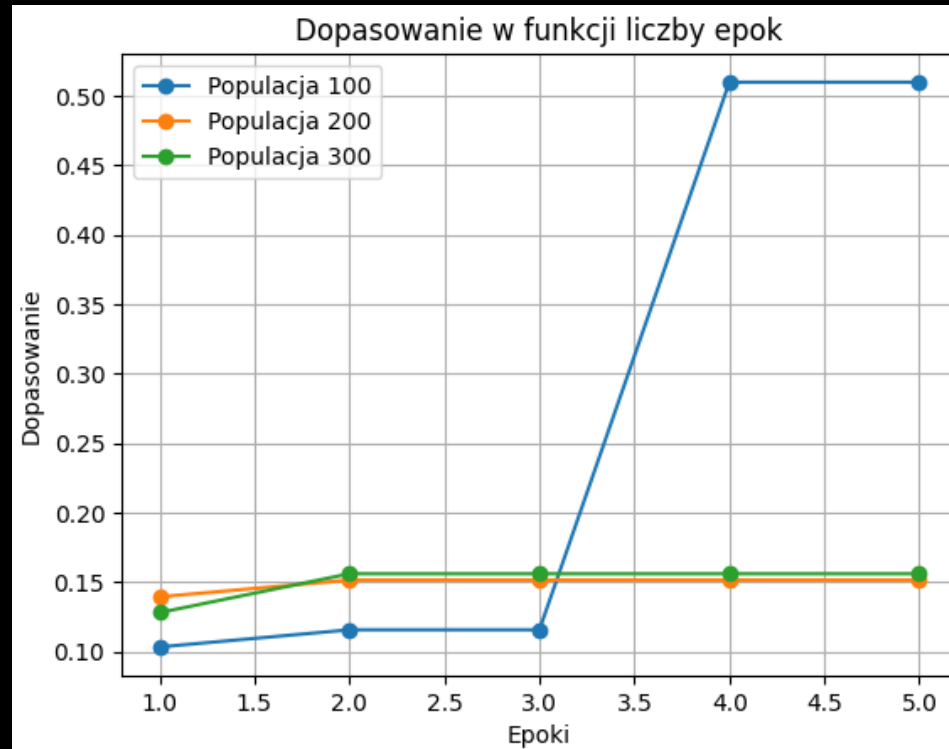
Kryterium stopu:

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

---

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

Średni błąd względny regulacji: 1.96%

---

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

Średni błąd względny regulacji: 6,60%

---

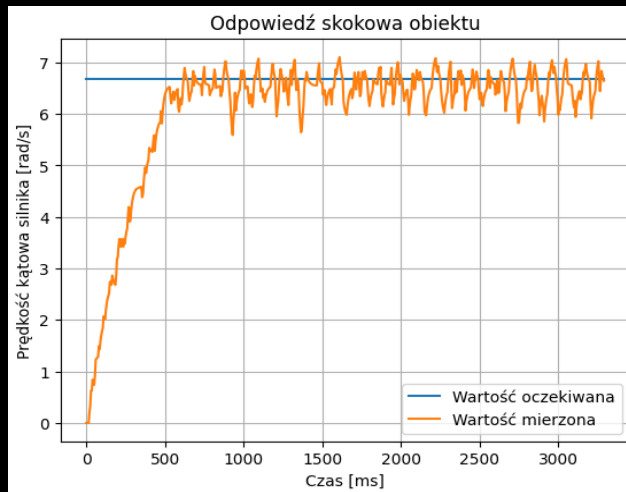
$K_p = 11.98, T_i = 1.59, T_d = 1.34$

Średni błąd względny regulacji: 6,41%

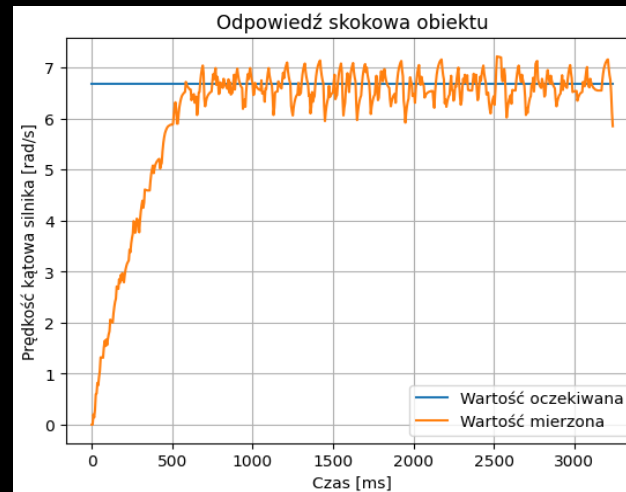
---

# Wyniki działania algorytmu

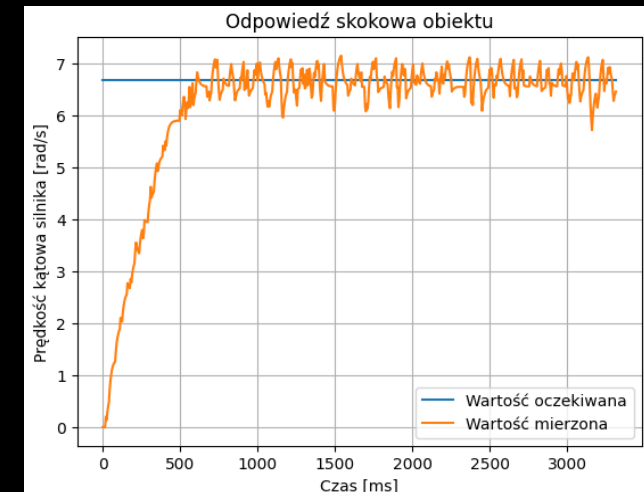
$K_p = 6.48, T_i = 2.93, T_d = 2.34$



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



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



# 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.
-

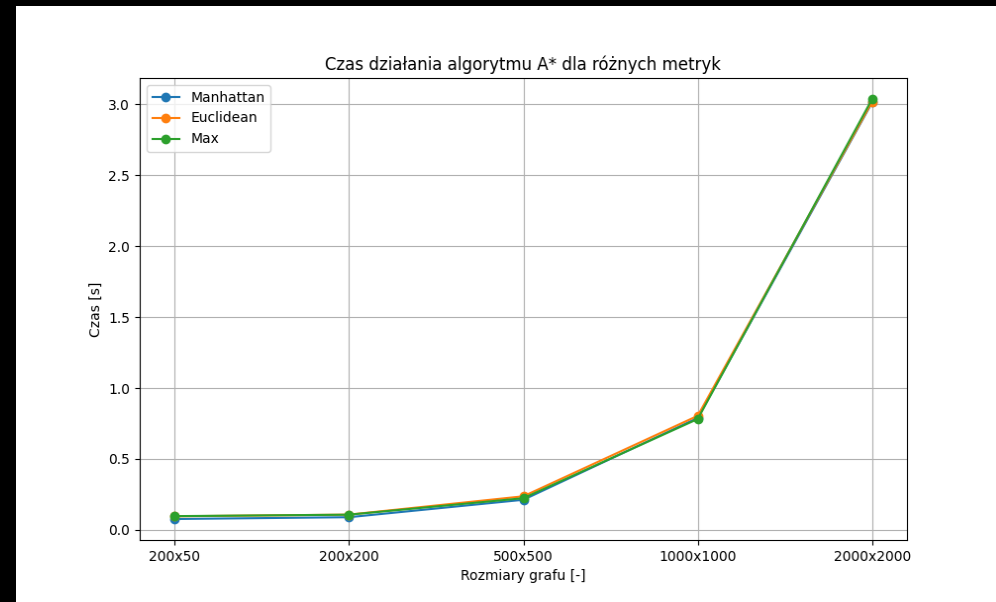
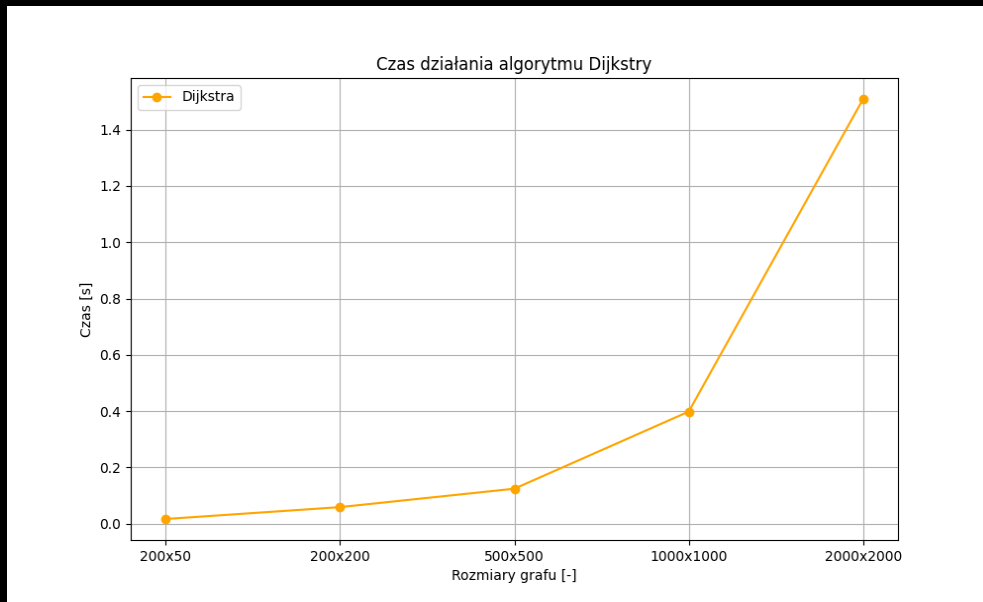


# Porównanie statycznych algorytmów nawigacyjnych - opis eksperymentu

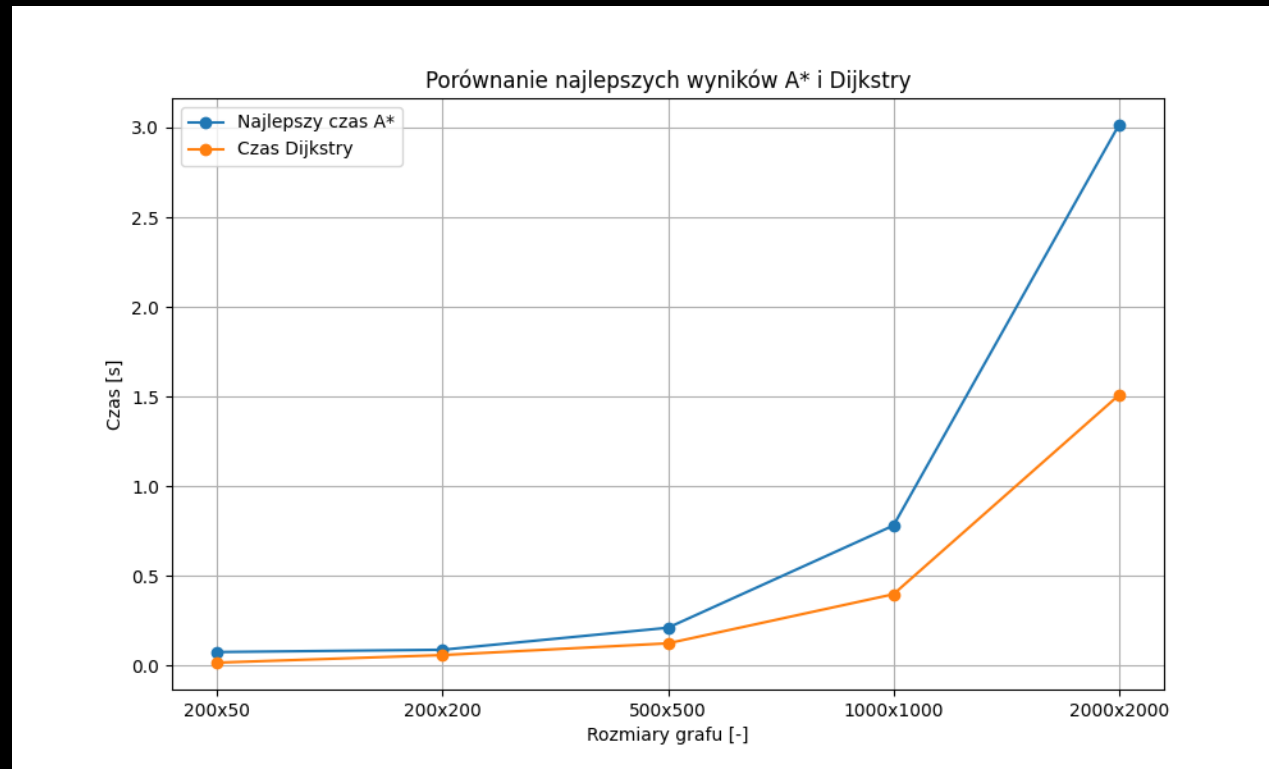
Eksperyment polegał na przetestowaniu algorytmu Dijkstry oraz A\* z wybranymi metrykami (Manhattan, Euclidean, Max) na mapach i porównaniu otrzymanych wyników:

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

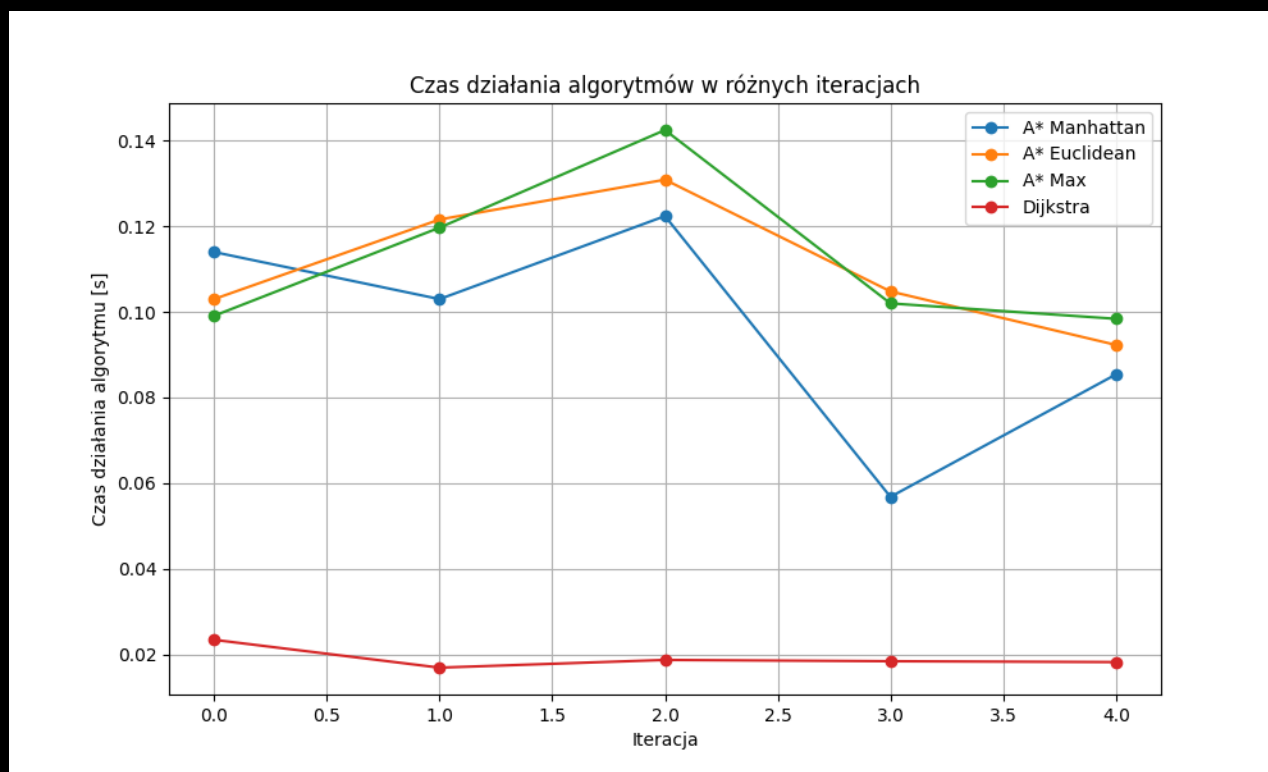
# Porównanie algorytmów statycznych na mapach bez przeszkód



# Porównanie algorytmów statycznych na mapach bez przeszkód



# Porównanie algorytmów statycznych na mapach z 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 Dijkstry oraz A\* z różnymi heurystykami uzyskały takie same wyniki pod względem długości trasy, jednak wymagały więcej czasu obliczeniowego w porównaniu do Dijkstry.
-

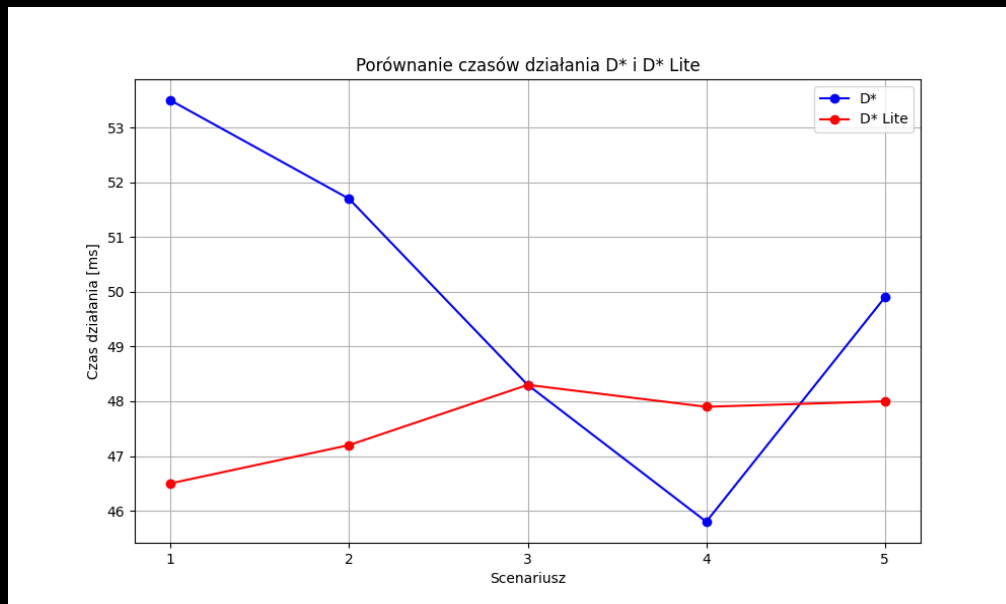
# Porównanie dynamicznych algorytmów nawigacyjnych - opis eksperymentu

Eksperyment polegał na przetestowaniu algorytmu D\* oraz D\*-Lite na rzeczywistej mapie i porównaniu wyników w celu wybrania optymalnego algorytmu dla zaimplementowanego robota. Eksperyment został przeprowadzony w 5 iteracjach. W każdej z nich:

- Wykorzystano mapę o rozmiarach 100x100 cm z wyznaczonymi punktami startu i stopu,
  - Zmieniano ilość i rozmieszczenie przeszkód na mapie.
-



# Porównanie dynamicznych algorytmów nawigacyjnych - wyniki



| Iteracja | Ilość przeszkód | Czas działania dstar [ms], Pokonana odległość | Czas działania dstar_lite [ms], Pokonana odległość |
|----------|-----------------|---|--|
| 1        | 4               | 53.5, 112                                     | 46.5, 112  |
| 2        | 4               | 51.7, 105                                     | 47.2, 105  |
| 3        | 0               | 48.3, 101                                     | 48.3, 101  |
| 4        | 3               | 45.8, 106                                     | 47.9, 107  |
| 5        | 2               | 49.9, 115                                     | 48.0, 115  |

# Porównanie dynamicznych algorytmów nawigacyjnych - wnioski

- Algorytm D\* Lite wykazał się minimalnie lepszym czasem działania w porównaniu do D\*. Taki wynik może być spowodowany tym, że w badanych scenariuszach występowała stosunkowo mała ilość napotkanych przeszkód oraz mapa była małych rozmiarów, co ograniczało liczbę operacji wymaganych do znalezienia optymalnej ścieżki,
  - Robot pokonał praktycznie tę samą odległość w każdym z pięciu przeprowadzonych scenariuszy testowych. Z uwagi na minimalnie lepszy czas działania, jaki uzyskał algorytm D\* Lite, warto zdecydować się na jego wybór, szczególnie w kontekście optymalizacji czasu obliczeń w bardziej wymagających środowiskach.
-

# Wkład własny

Za wkład własny autor uważa:

- Implementację robota mobilnego
  - Implementację systemu wbudowanego robota
  - Implementację skryptów,  
przeprowadzenie eksperymentów i analizę wyników
-