



UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU

Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Wybrane aspekty pojazdów autonomicznych

Nawigacja

Rafał Szczepański

e-mail: szczepi@umk.pl

www.umk.pl/~szczepi

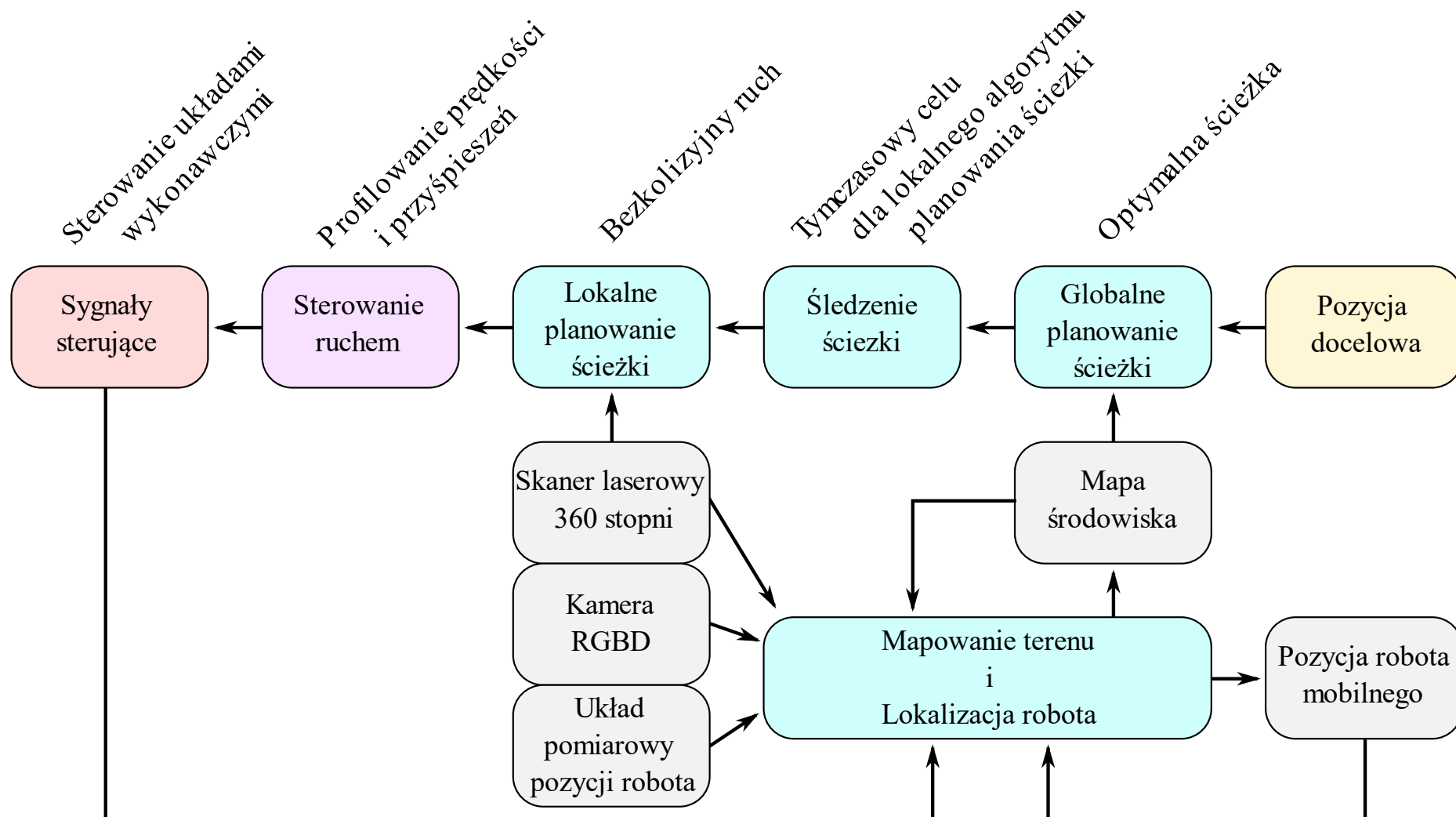
03.06.2024



Plan prezentacji

- Czujniki dotykowe
- Pomiar położenia
- Czujniki bezwładnościowe
- Pomiar odległości
- Analiza obrazu

Nawigacja





Lokalizacja

Autonomiczna nawigacja wymaga informacji o położeniu i orientacji w pewnym, przyjętym układzie współrzędnych.

Rozwiązanie problemu lokalizacji pojazdu autonomicznego wymaga połączenia kilku metod wyznaczania zmiennych stanu układu umożliwiających określenia położenia i orientacji.

Rozważane są następujące układy współrzędnych:

- **absolutny** – korzysta z informacji pochodzących z globalnych punktów orientacyjnych (m.in. GPS) wyznaczenie położenia absolutnego,
- **względny** – w każdej kolejnej dyskretniej chwili czasu, akumulowana informacja o prędkości liniowej i kątowej, siłach działających na pojazd, analiza cech wizualnych, pozwala na aktualizowanie informacji o położeniu względnym.



Odometria - Kinematyka prosta

- Kinematyka prosta rozwiązuje problem: jeżeli znam pozycje, prędkości i przyspieszenia kątowe kół to możemy obliczyć aktualną pozycję, prędkości oraz przyspieszenia robota względem początkowej pozycji (tj. prędkość liniową oraz kątową).
- Kinematyk prosta jest zdefiniowana funkcją f przechodzącą z przestrzeni kół do przestrzeni do przestrzeni roboczej:

$$f: \mathbb{R}^n \rightarrow \mathbb{R}^m$$

$$x = f(q) \quad x \in \mathbb{R}^n, q \in \mathbb{R}^m$$



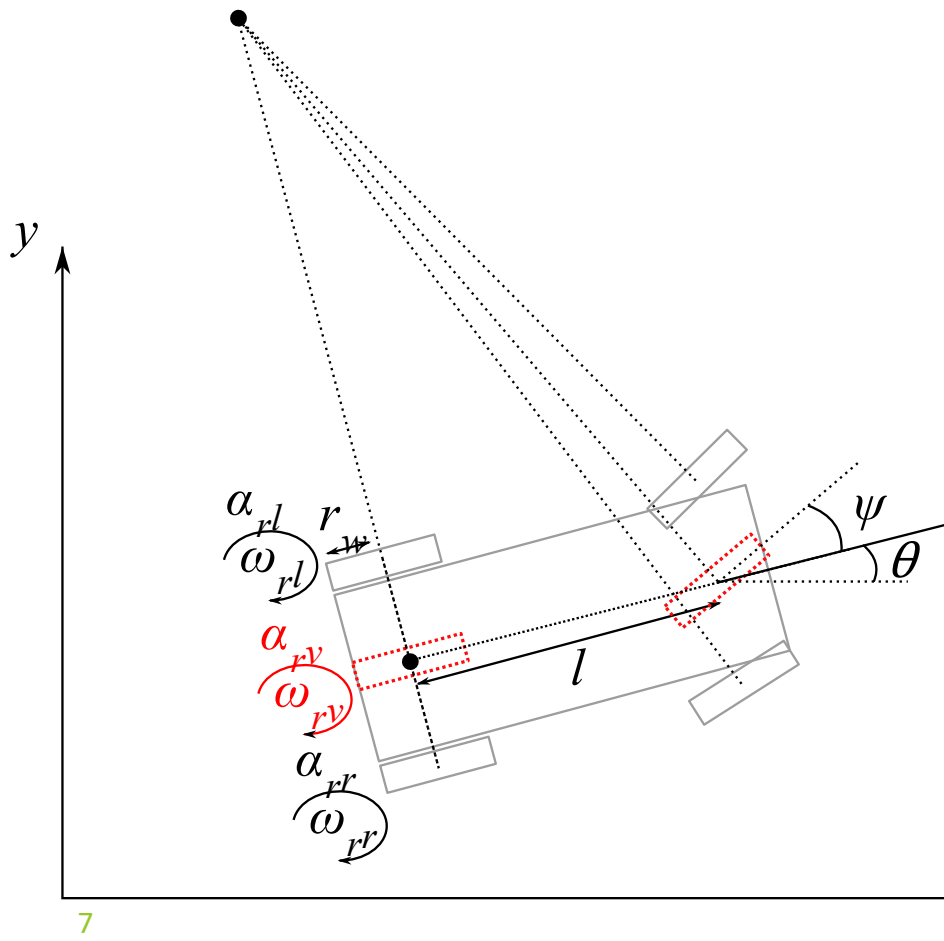
Odometria - Kinematyka odwrotna

- Kinematyka odwrotna rozwiązuje problem: jeżeli znam pozycje, prędkości i przyspieszenia robota w przestrzeni roboczej to możemy obliczyć prędkości i przyspieszenia kątowe kół.
- Kinematyk prosta jest zdefiniowana funkcją g przechodzącą z przestrzeni roboczej do przestrzeni do przestrzeni kół:

$$g: \mathbb{R}^m \rightarrow \mathbb{R}^n$$

$$q = g(x) = f^{-1}(x) \quad x \in \mathbb{R}^n, q \in \mathbb{R}^m$$

Odometria - Model kinematyczny robota czterokołowego z więzami nieholonomicznymi (ang. *car-like*)



- W robocie występują dwa napędy: napęd kół tylnych oraz skręt kół przednich.
- Tworzymy wirtualne koła będące pośrodku kół fizycznych tylnych oraz przednich.
- Prędkości i pozycje kół wirtualnych wynoszą odpowiednio:

$$\omega_{rv} = \frac{\omega_{rr} + \omega_{rl}}{2}$$

$$\alpha_{rv} = \frac{\alpha_{rr} + \alpha_{rl}}{2}$$

$$\psi = \frac{\psi_{fr} + \psi_{fl}}{2}$$

- Sygnałami sterującymi jest prędkość tylnego koła wirtualnego (ω_{rv}) oraz pozycja kątowa układu sterowniczego (ϕ)

Odometria - Model kinematyczny robota czterokołowego z więzami nieholonomicznymi (ang. *car-like*)

Dzięki wprowadzeniu wirtualnych kół otrzymujemy stosunkowo prosty problem. Koła tylne powodują ruch wzdłuż aktualnej orientacji robota, natomiast skręt kół przednich powoduje zmianę orientacji robota.

Rozwiązanie kinematyki prostej względem układu współrzędnych związanego z robotem jest następujące:

$$v = r_w \cdot \omega_{rv}$$
$$\dot{\theta} = \frac{\operatorname{tg} \psi}{l} \cdot r_w \cdot \omega_{rv}$$

Rozwiązanie kinematyki odwrotnej względem układu współrzędnych związanego z robotem jest następujące:

$$\omega_{rv} = \frac{v}{r_w}$$
$$\phi = \operatorname{arctg} \left(\frac{l \dot{\theta}}{v} \right)$$



Odometria - Model kinematyczny robota czterokołowego z więzami nieholonomicznymi (ang. *car-like*)

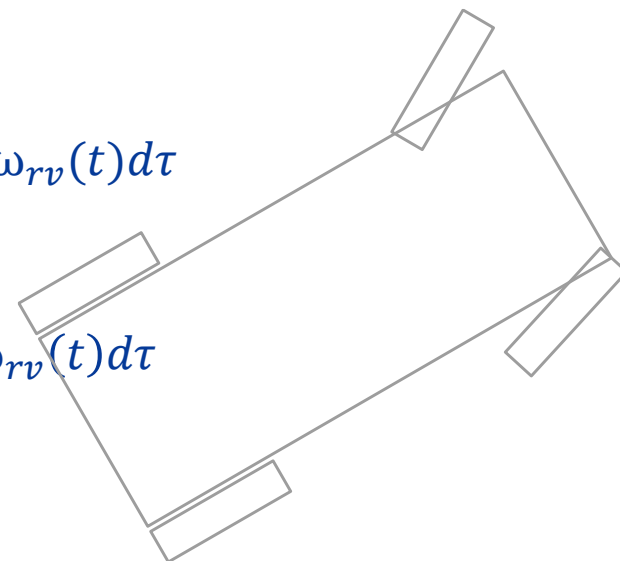
Odometria – wyznaczenie trajektorii pojazdu autonomicznego poprzez sumowanie prędkości obrotowej koła;

Rozwiązanie odometrii jest następujące:

$$x_r(t) = x_r(t - \Delta t) + \cos(\theta(t)) r_w \int_{t-\Delta t}^t \omega_{rv}(\tau) d\tau$$

$$y_r(t) = y_r(t - \Delta t) + \sin(\theta(t)) r_w \int_{t-\Delta t}^t \omega_{rv}(\tau) d\tau$$

$$\theta(t) = \theta(t - \Delta t) + \operatorname{tg}(\psi(t)) \frac{r_w}{l} \int_{t-\Delta t}^t \omega_{rv}(\tau) d\tau$$



Odometria - Model kinematyczny robota czterokołowego z więzami nieholonomicznymi (ang. *car-like*)

Sekwencja danych
wejściowych do kinematyki
odwrotnej:

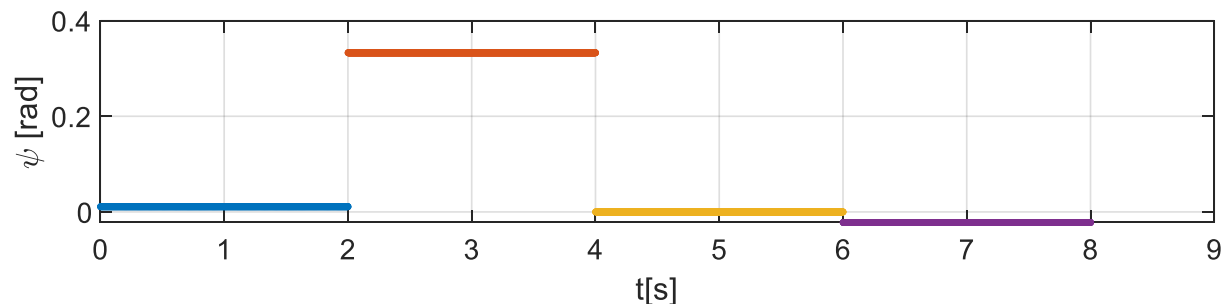
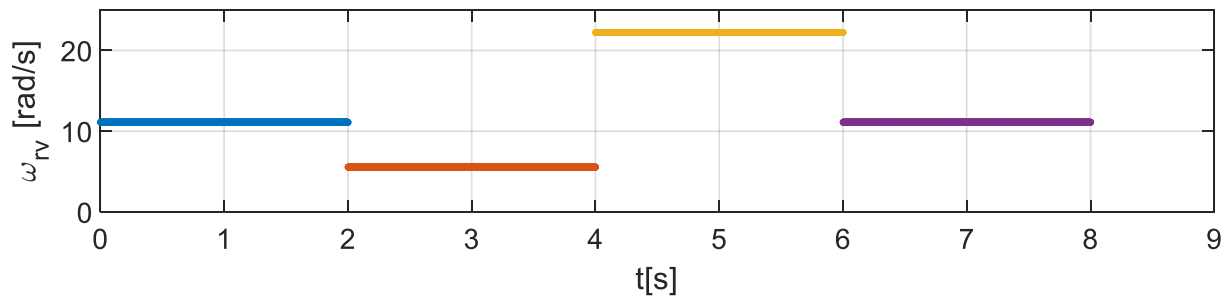
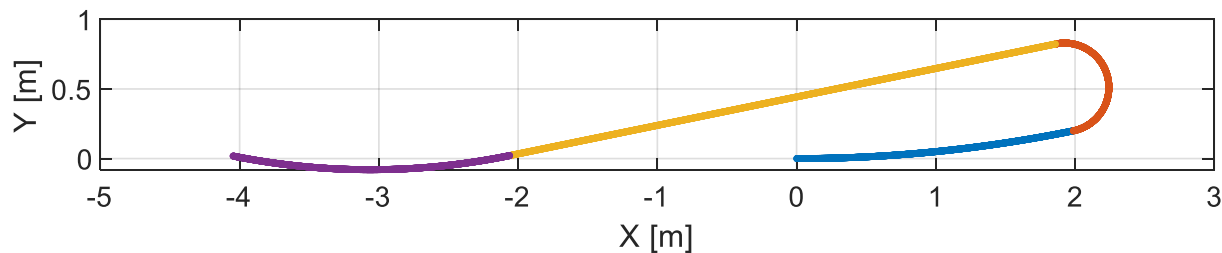
$$v = [1, 0.5, 2, 1]; \text{ m/2}$$

$$\dot{\theta} = [0.1, \pi/2, 0, -0.2]; \text{ rad/s}$$

Wyniki kinematyki odwrotnej:

$$\omega_{rv} = [11. (1), 5. (5), \\ 22. (2), 11. (1)] \text{ rad/s}$$
$$\psi = [0.011, 0.33, \\ 0, -0.022] \text{ rad}$$

Wyniki kinematyki odwrotnej i
odometrii na wykresie.



Wyznaczanie położenia za pomocą systemu bezwładnościowego

Każda zmiana kierunku ruchu przejawia się w mierzalnej sile przyspieszenia.

Zmiana kierunku i / lub prędkości pojazdu wymaga jego przyspieszenia.

Pomiary przyspieszenia pozwalają wnioskować o zmianach stanu oraz interpretować pojawiającą się zmianę położenia, jeżeli znana jest wiedza o początkowym, pozbawionym przyspieszenia stanie pojazdu, włączając w to ewentualną stałą wartość prędkości.

Pomiary przyspieszenia zapewniają odporność na błędy związane np. z poślizgiem.

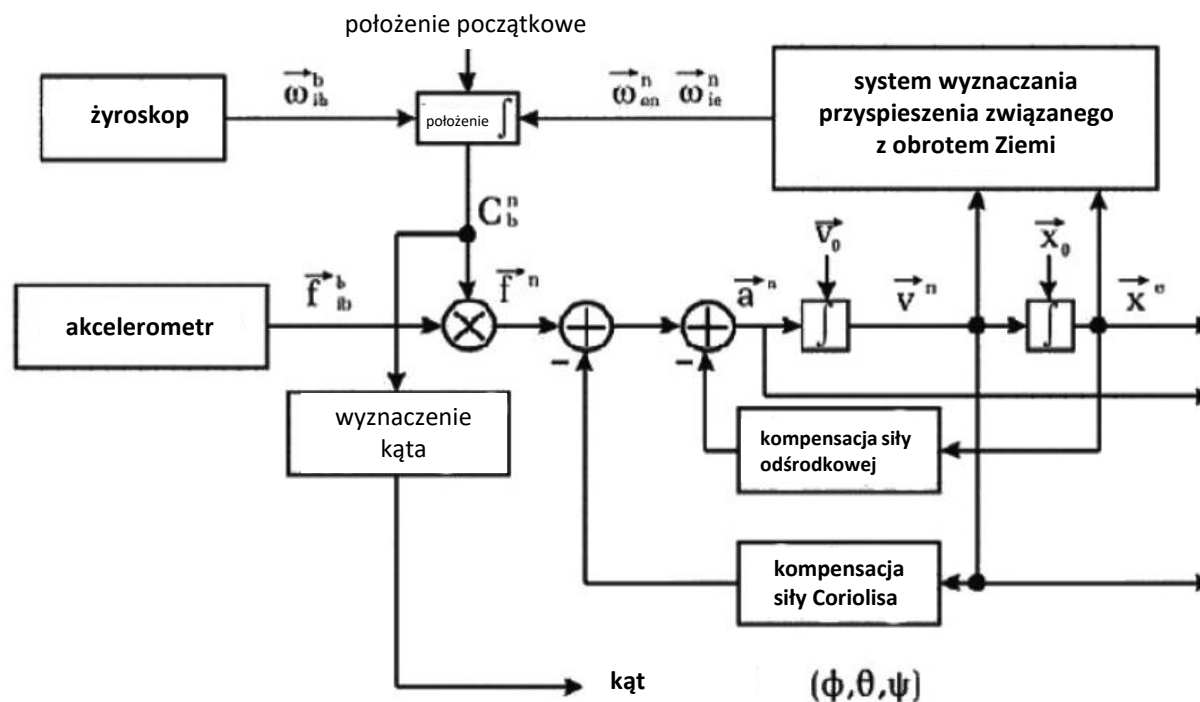
Ciekawe wykorzystanie w urządzeniu pomiarowym bazujące na tej zasadzie działania:

<https://www.moasure.com>

Wyznaczanie położenia za pomocą systemu bezwładnościowego

Przykładowe zakłócenia związane z zastosowaniem systemu bezwładnościowego:

- obracanie się Ziemi,
- grawitacja,
- inkrementalny charakter wyznaczania położenia z przyspieszeń.





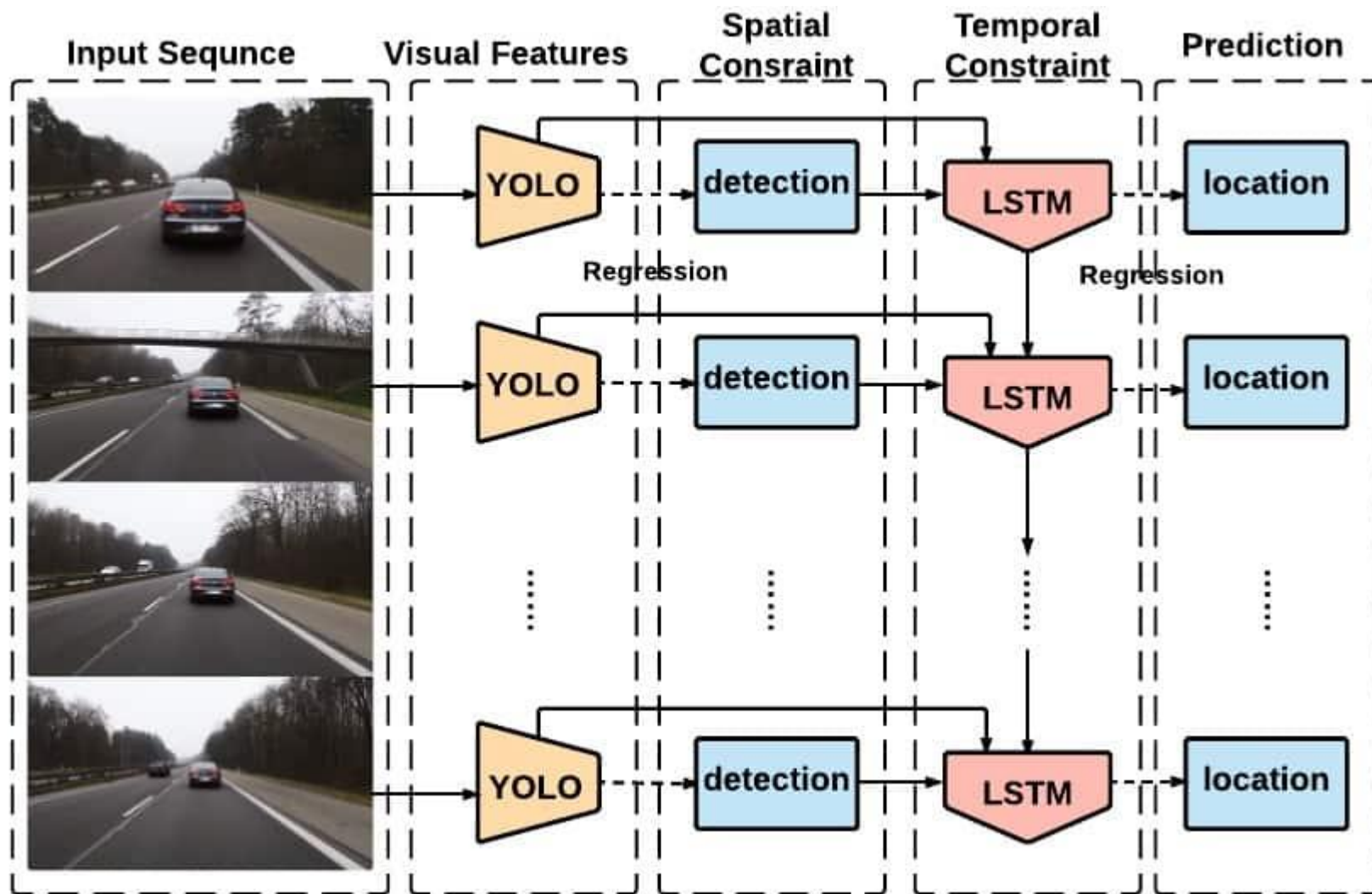
Wyznaczanie położenia za pomocą systemu optycznego

Wizualna odometria bazuje na jedno- lub wielookularowym systemie kamer. Omawiane podejście zostało zastosowane m.in. w robotach marsjańskich;

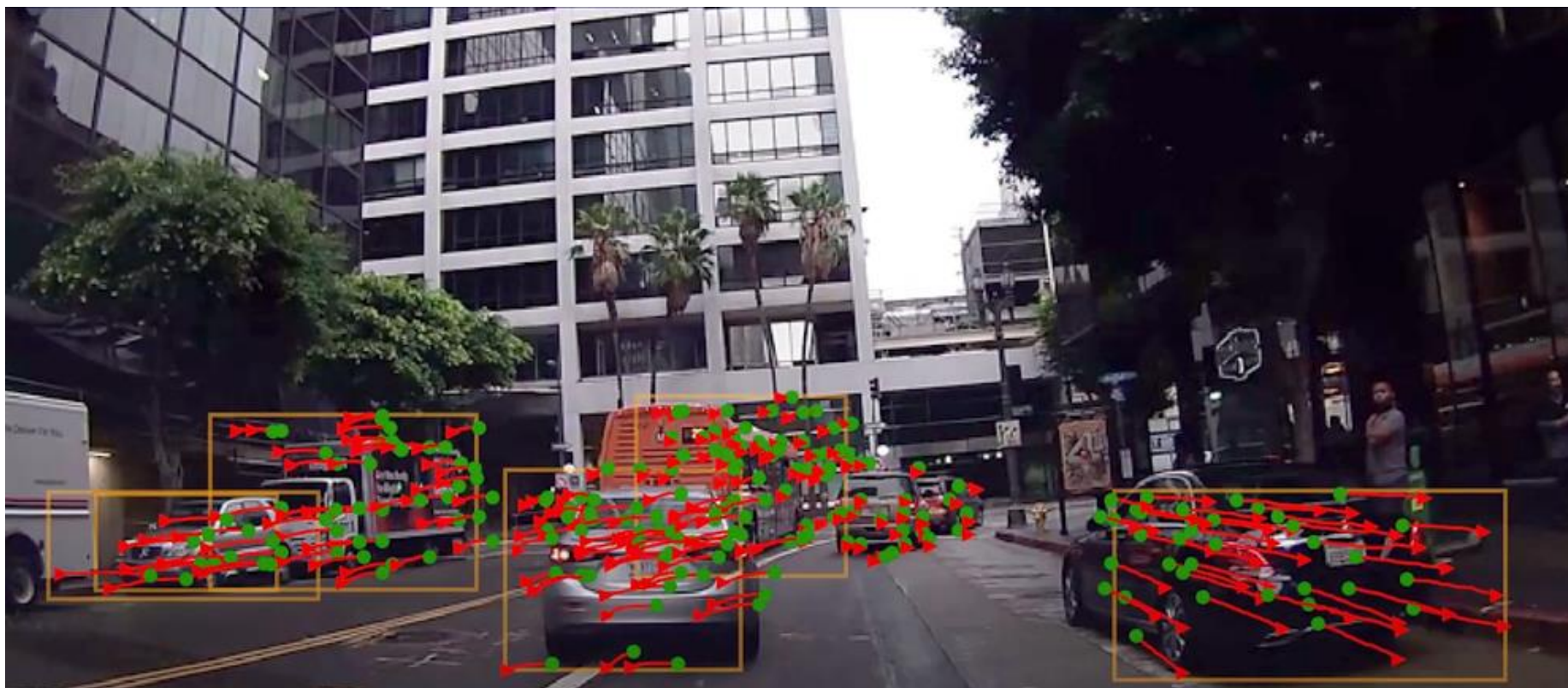
Zasada działania:

- system kamer przechwytuje obraz,
- na podstawie analizy obrazu wybierany jest zestaw cech odpowiednich do śledzenia,
- po przechwyceniu kolejnego obrazu, analizowane (śledzone) jest położenie wybranych cech,
- wektory między punktami zawierają informacje o zmianie orientacji i położenia pojazdu między dwiema kolejnymi ramkami,
- ostatni krok polega na oszacowaniu przemieszczenia robota w odniesieniu do punktu początkowego;

Wyznaczanie położenia za pomocą systemu optycznego



Wyznaczanie położenia za pomocą systemu optycznego



Wyznaczanie położenia za pomocą systemu optycznego

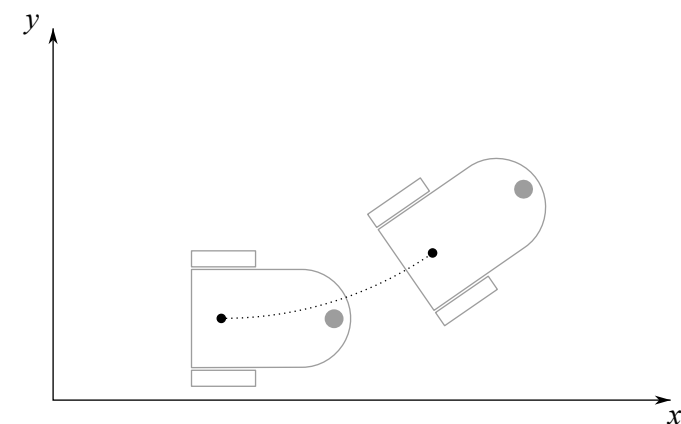


Wychwytywanie cech na podstawie danych pochodzących z LiDAR

Mapa środowiska

Zaszumiony odczyt skanera laserowego

Pozycja i orientacja



Zaszumiony pomiar odometryczny

Fuzja
Filtrowanie



Punkty orientacyjne (znaczniki)

Punkty orientacyjne (wskaźniki) – łatwe do wykrycia cechy zarejestrowanego obrazu umożliwiające ustalenie pozycji i orientacji pojazdu.

Klasyfikacja punktów orientacyjnych:

- naturalne – łatwo rozpoznawalne, nieruchome przedmioty, krawędzie przedmiotów,
- sztuczne – kody kreskowe,
- aktywne – emitują energię (zazwyczaj w bliskiej podczerwieni); wiązka światła zawiera zakodowaną informację o współrzędnych,
- pasywne;

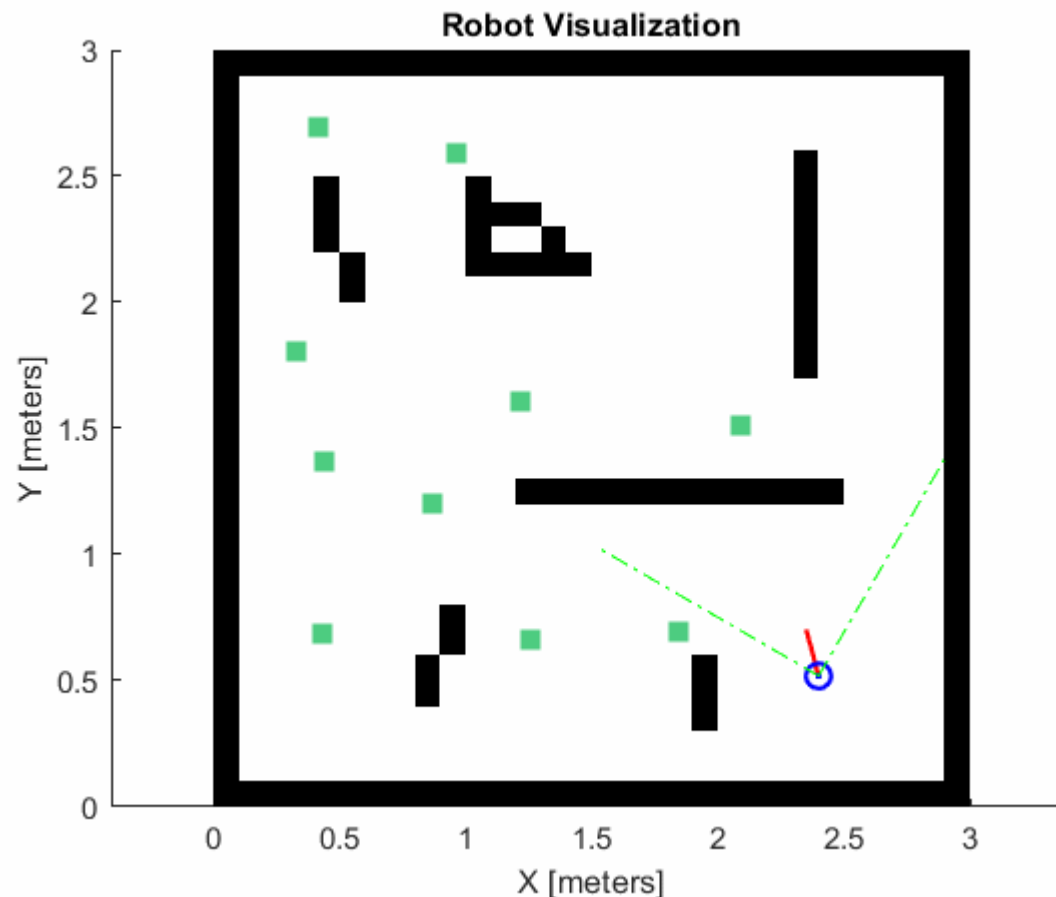
Rozszerzony filtr Kalmana

Robot bazując na pozycji
wyznaczonej na bazie EKF
podąża za wyznaczoną
ścieżką przez algorytm
globalny algorytmem czysty
pościg (ang. *PurePursuit*).

Punkty orientacyjne

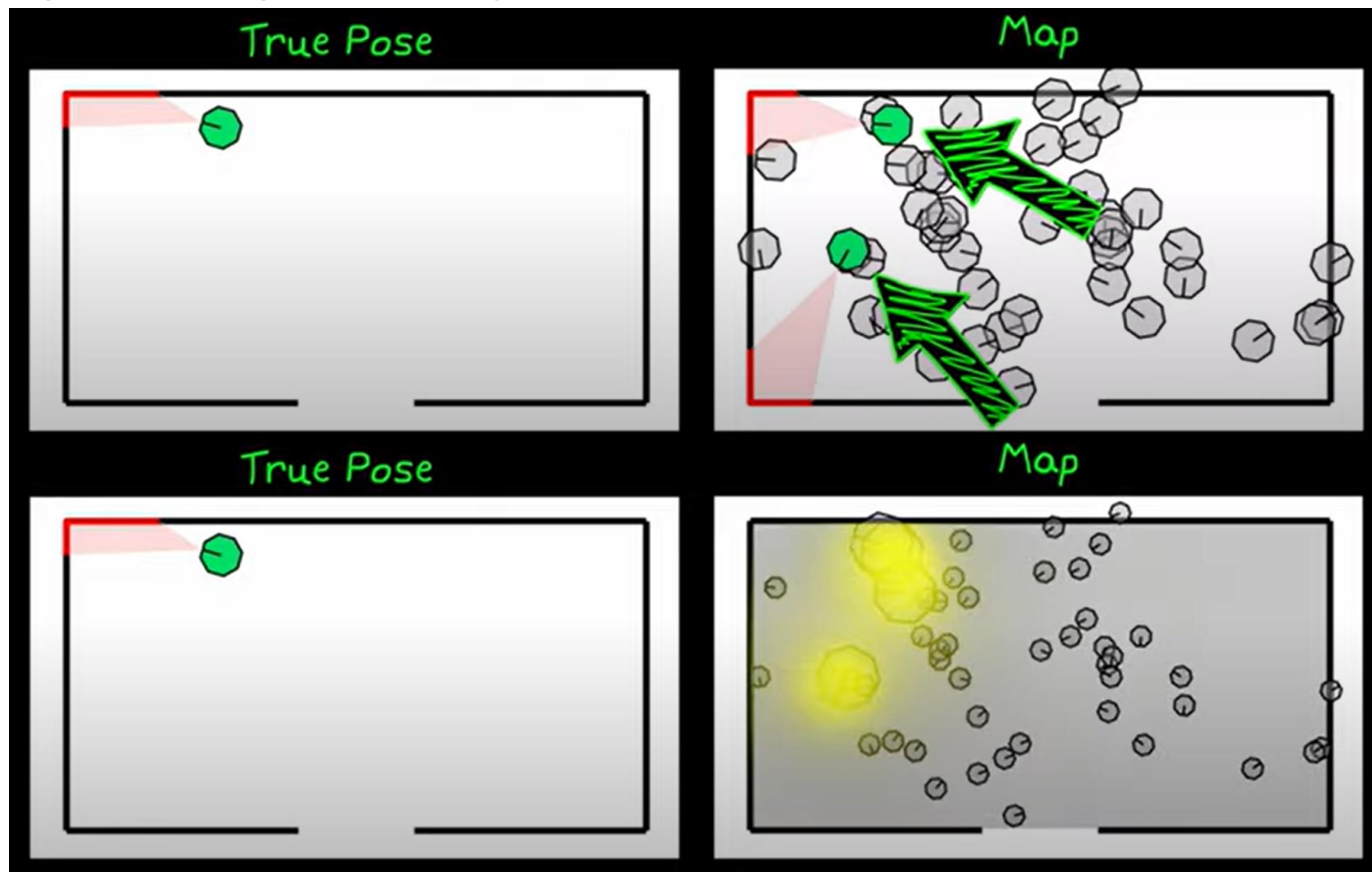
Ścieżka globalna

Przebyta trasa przez robota



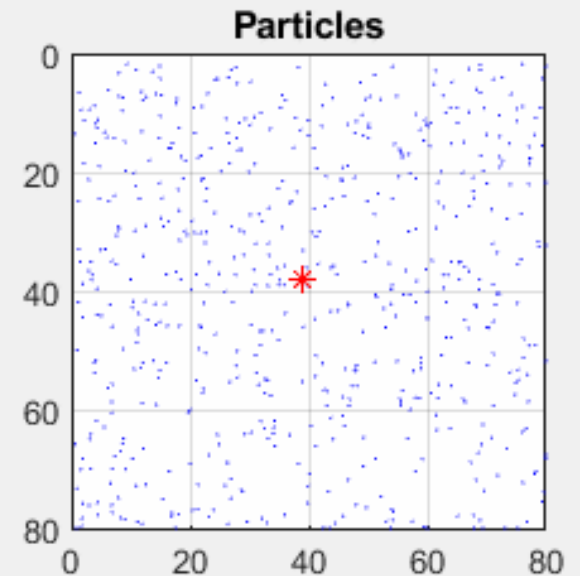
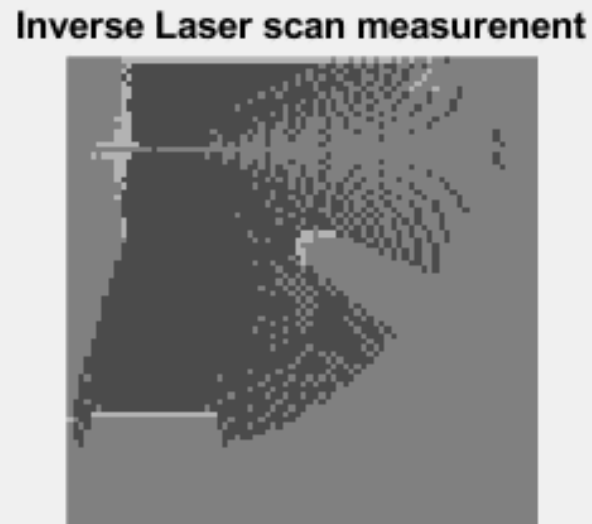
Sekwencyjny algorytm Monte Carlo

Inaczej: Filtr cząsteczkowy



Sekwencyjny algorytm Monte Carlo

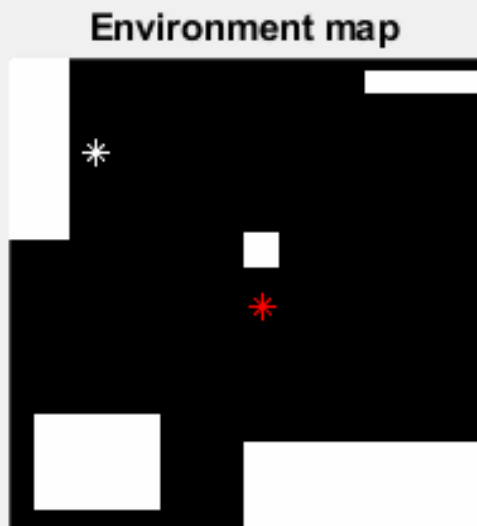
Inaczej: Filtr cząsteczkowy – 1000 cząstek



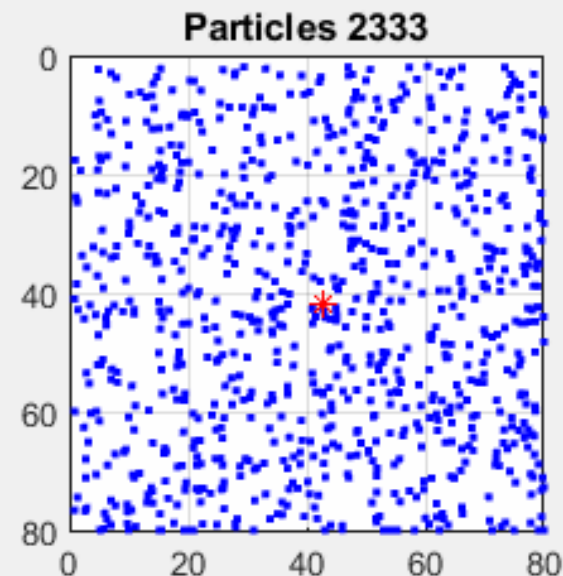
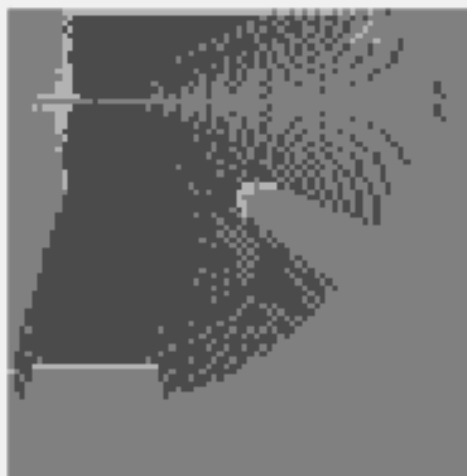
Adaptacyjny algorytm lokalizacji Monte Carlo (ang. *adaptive (or KLD-sampling) Monte Carlo localization, AMCL*)

Wariacja szumu procesu $V = 0.3$

Wariacja szumu pomiaru $W = 0.1$



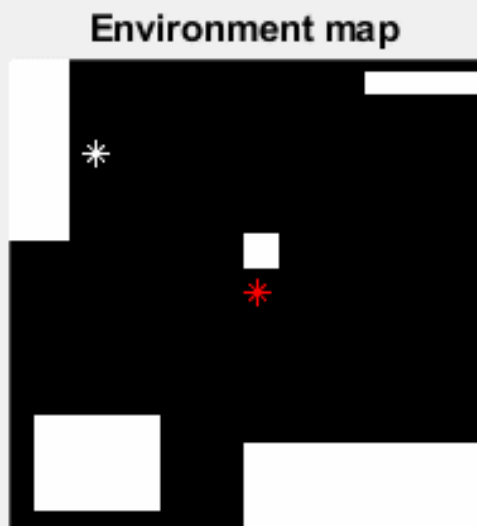
Inverse Laser scan measurement



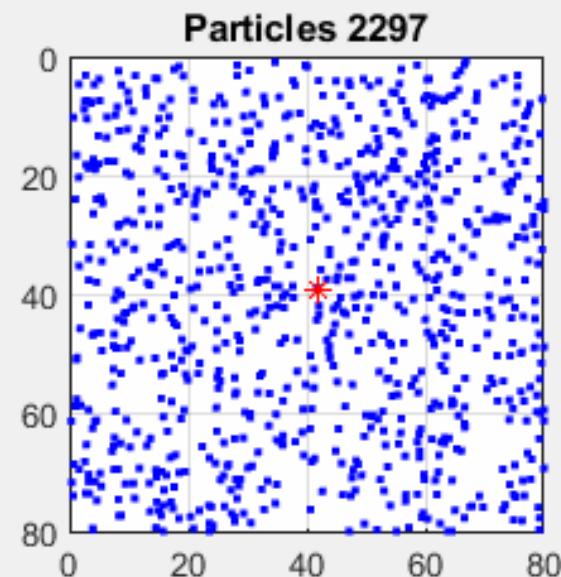
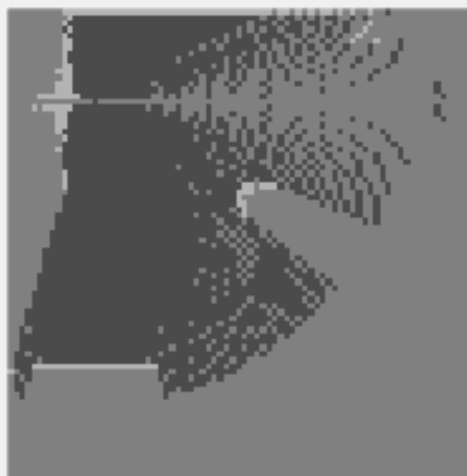
Adaptacyjny algorytm lokalizacji Monte Carlo (ang. *adaptive (or KLD-sampling) Monte Carlo localization, AMCL*)

Wariacja szumu procesu $V = 0.6$

Wariacja szumu pomiaru $W = 0.2$



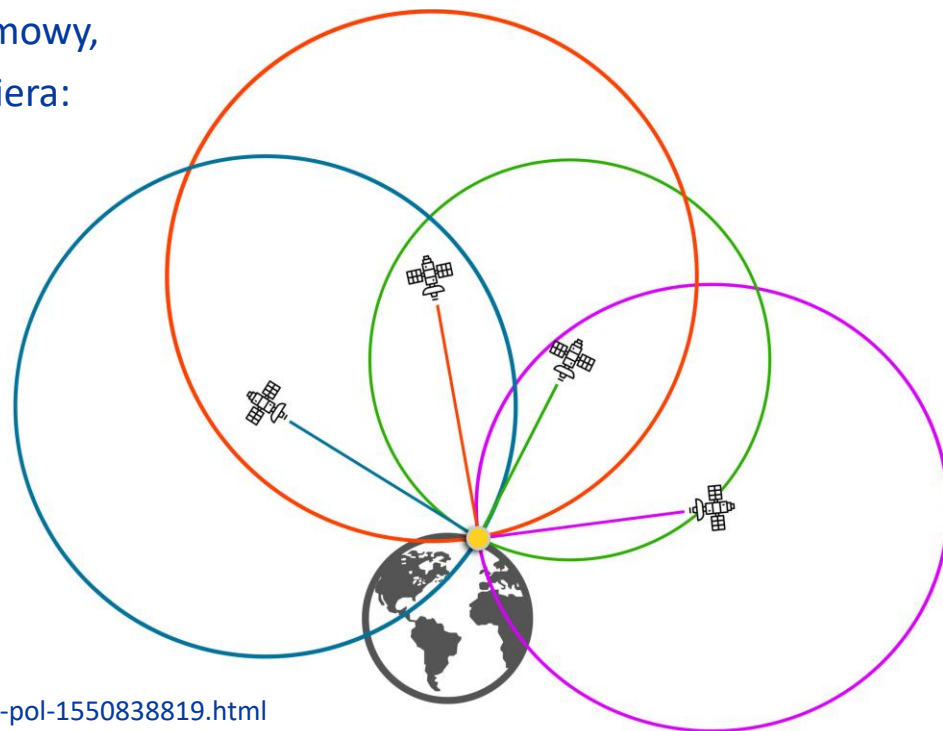
Inverse Laser scan measurement



Globalny system pozycjonowania (GPS)

Podstawowe własności systemu:

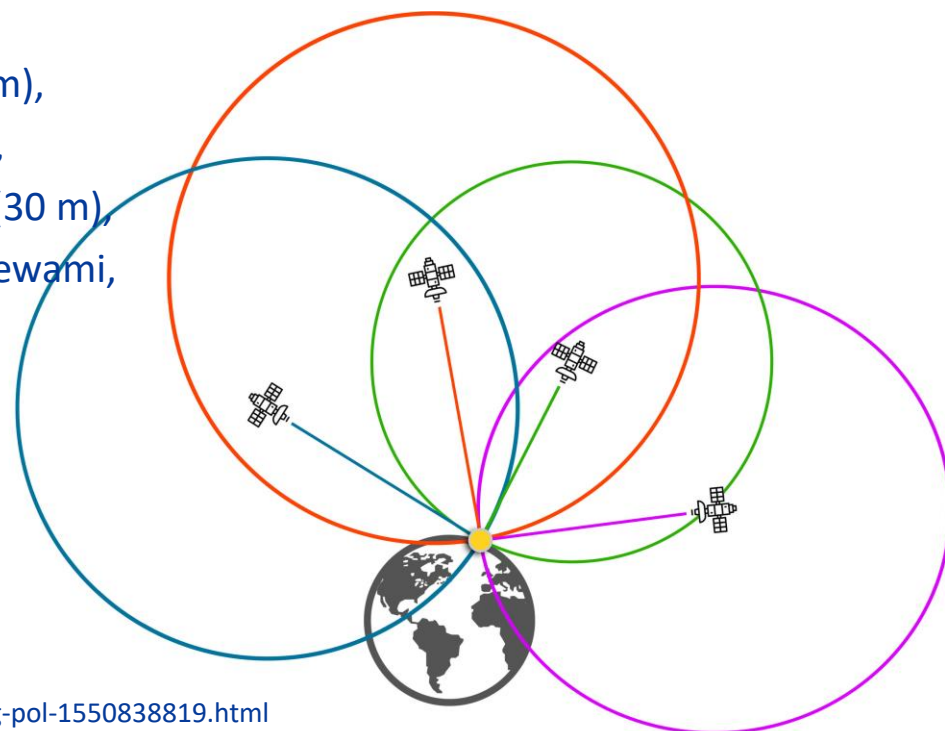
- opracowany w latach 70-tych przez Departament Obrony USA do celów militarnych,
- wykorzystuje 24 satelity umieszczone na 6-ciu orbitach,
- każdy satelita pokrywa 1/6 powierzchni Ziemi,
- w dowolnym momencie, z każdego punktu na Ziemi można zobaczyć przynajmniej 4 satelity,
- każdy satelita jest wyposażony w zegar atomowy,
- informacja wysyłana z każdego satelity zawiera:
 - numer satelity,
 - pozycję satelity,
 - stempel czasowy.



Globalny system pozycjonowania (GPS)

Podstawowe własności systemu (cd.):

- wyznaczenie pozycji odbiornika realizowane jest na podstawie pomiarów różnic w czasach odbioru sygnałów pochodzących z 4-ech satelitów,
- potencjalnymi źródłami błędów pomiarowych są:
 - pozycja satelity (3 m),
 - refrakcja jonosfery (5 m),
 - załamywanie fal w troposferze (2 m),
 - wielokrotne odbicie sygnału (5 m),
 - selektywna dostępność satelitów (30 m),
 - zanik sygnałów spowodowany drzewami, wysokimi budynkami, tunelami;





Globalny system pozycjonowania (GPS)



GPS



GNSS



Lokalizacja i planowanie trajektorii - mapy

Metody nawigowania pojazdu autonomicznego poza jego horyzontem sensorycznym:

- strategia poszukiwań lokalnych (np. bug algorithm) – może skutkować dużymi niedokładnościami,
- modelowanie świata zewnętrznego – przechowywanie wskazówek dla nawigacji:
 - dostarczanie map a priori,
 - budowanie map online – zastosowanie algorytmu mapowania.

Rodzaje mapowania:

- świato-centriczne – informują o położeniu wszystkich obiektów (również pojazdu) w ustalonym układzie współrzędnych,
- robo-centriczne – stosowane w sterowaniu np. do unikania kolizji.

Transformacja układów współrzędnych umożliwia płynne przechodzenie między rodzajami mapowania.



Mapowanie świato-centryczne

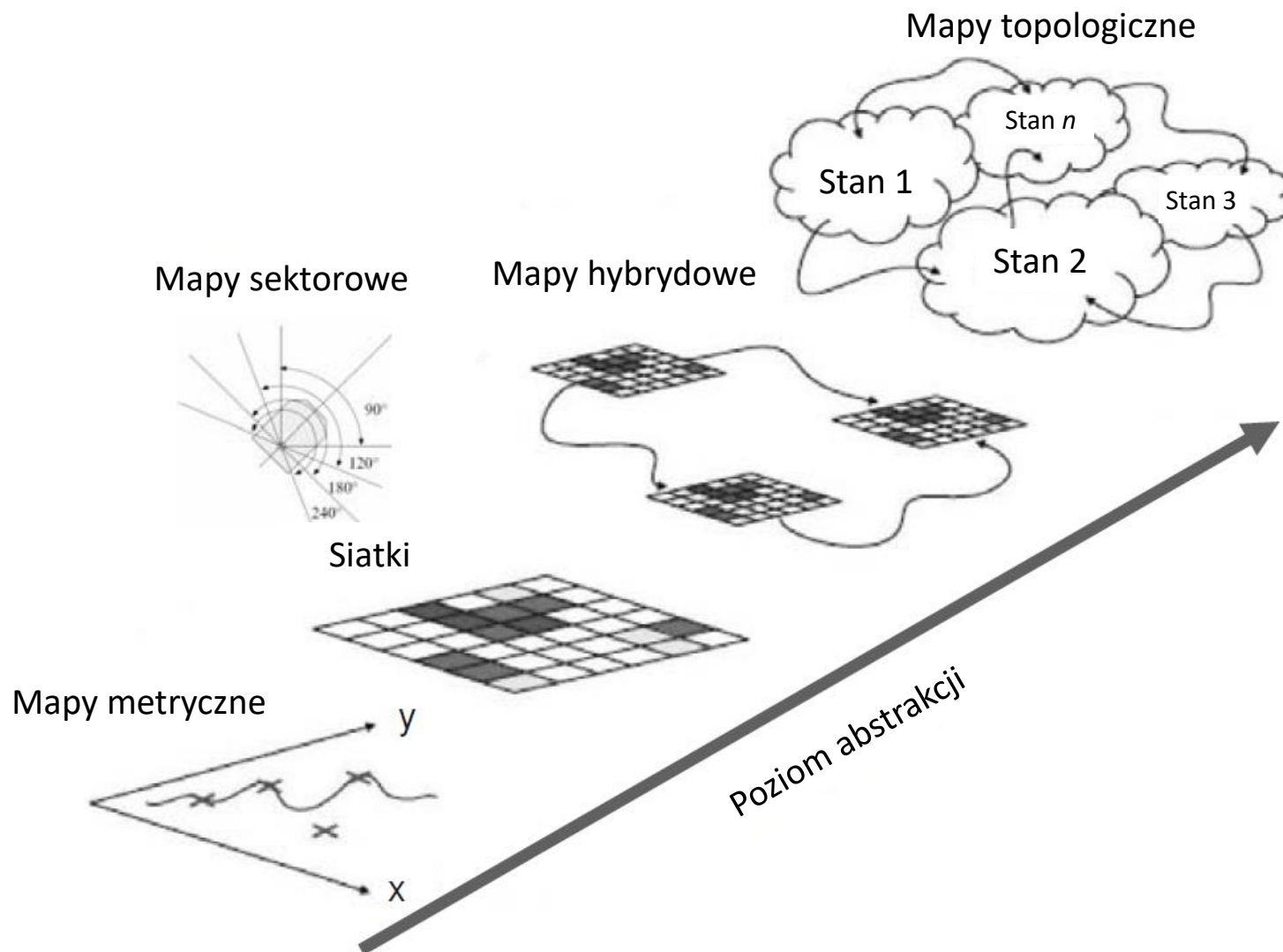
Wykorzystywane do nawigacji lub planowania trasy:

- wewnątrz obiektów – korzysta z narożnika pomieszczenia lub niezmiennej powierzchni,
- środowisko zewnętrzne – korzysta z jednego z dostępnych globalnych systemów pozycjonujących (np. GPS, GLONASS);

Niedokładności systemów pomiarowych stosowanych do lokalizacji pojazdu i mierzenia jego odległości od innych obiektów powoduje:

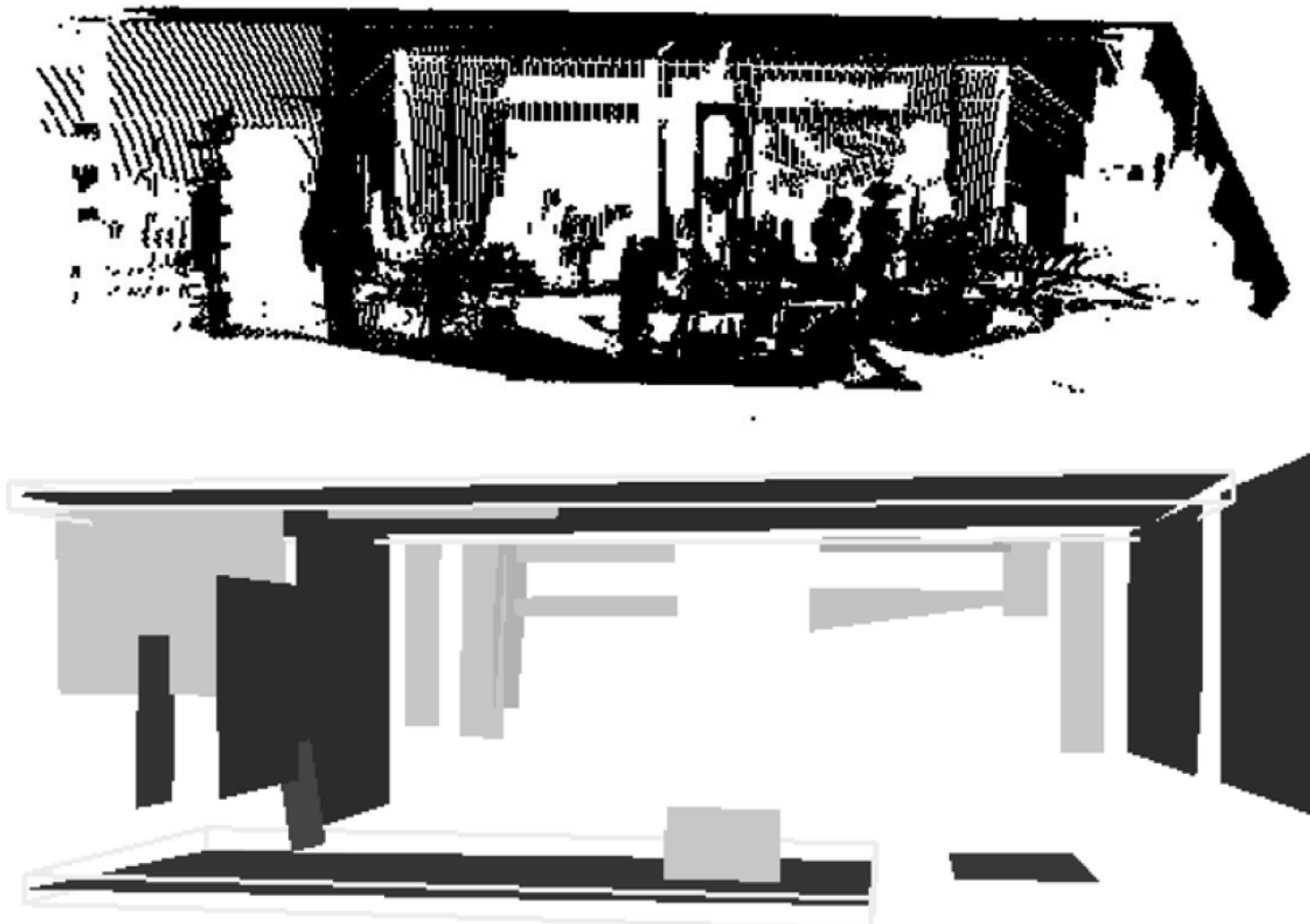
- trudności podczas generowania map globalnych na podstawie danych lokalnych,
- problemy podczas aktualizowania istniejących map na podstawie lokalizowania obiektów naniesionych wcześniej na mapę;

Lokalizacja i planowanie trajektorii - mapy





Mapy metryczne – wyodrębnianie powierzchni z zarejestrowanego zbioru punktów 3D

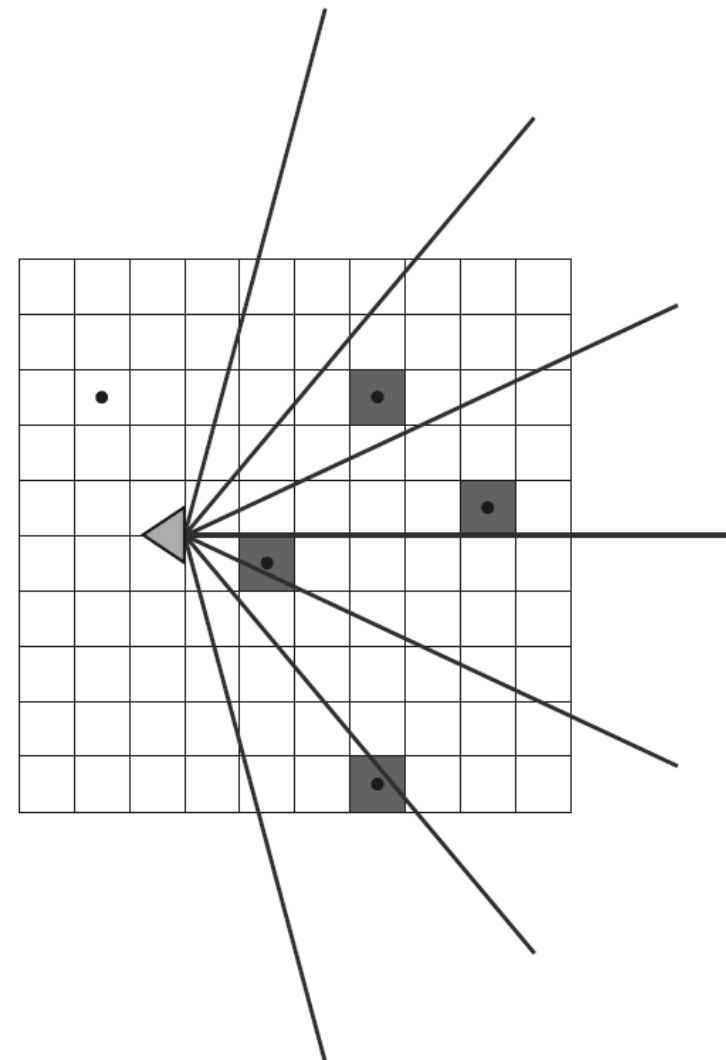
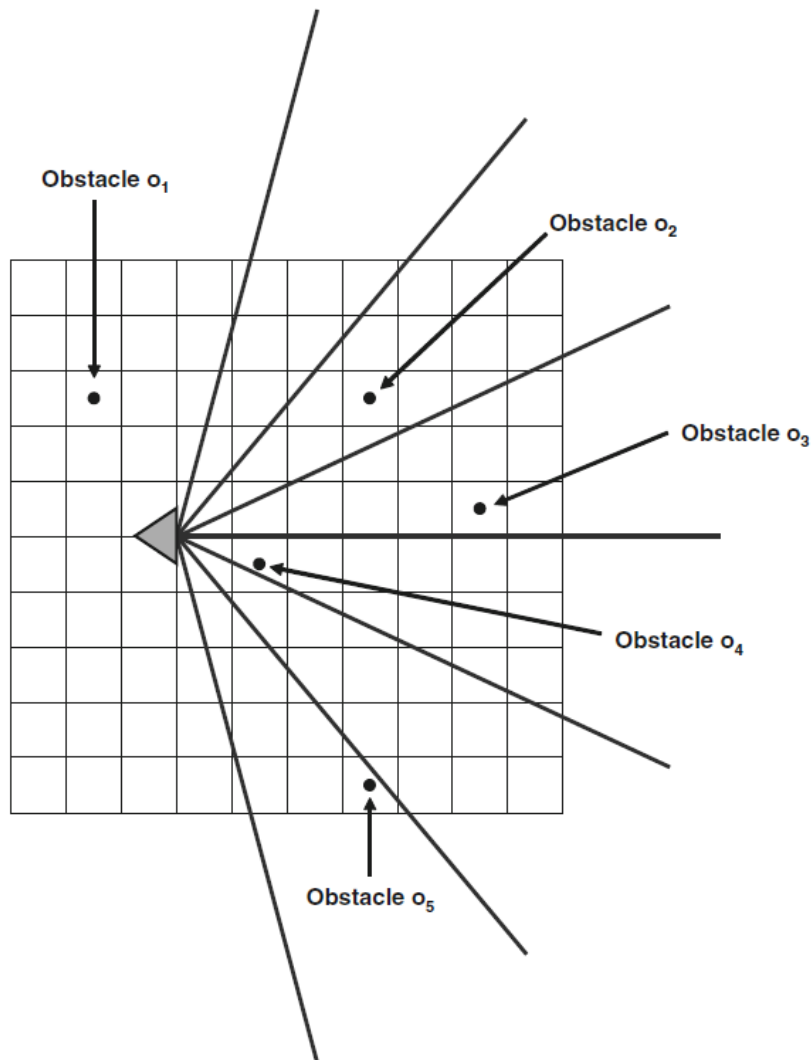




Siatki zajętości

- Często wykorzystywane w nawigacji robotów mobilnych.
- Popularne ze względu na prostotę oraz intuicyjną reprezentację.
- Podczas tworzenia siatki, przestrzeń dzielona jest na obszary o jednakowych rozmiarach metrycznych.
- Obszary wypełnione są informacjami uzyskanymi z czujników (sonarów, laserów).
- Przyjmuje się, że brak informacji z czujnika najprawdopodobniej oznacza wolną przestrzeń.
- Poprawne mapowanie wymaga znajomości położenia robota w przestrzeni (może bazować na estymacji).
- Zalety: odporność na zakłócenia, łatwa implementacja algorytmów.
- Wady: wymagają informacji o dokładnej pozycji pojazdu.

Siatki zajętości





Mapy topologiczne

Abstrakcyjne opisy wielkoskalowych struktur przestrzennych.

Reprezentowane jako grafy składające się z:

- węzłów – charakteryzują miejsca istotne dla nawigacji,
- krawędzi – definiują połączenia między poszczególnymi węzłami.

Korzystają z informacji metrycznych: współrzędnych, długości metryczne krawędzi.

Przetwarzanie map topologicznych:

- polega na generowaniu i odnajdywaniu ścieżek,
- mniej złożone obliczeniowo w porównaniu do przetwarzania map metrycznych.



Mapy topologiczne

Przykłady:

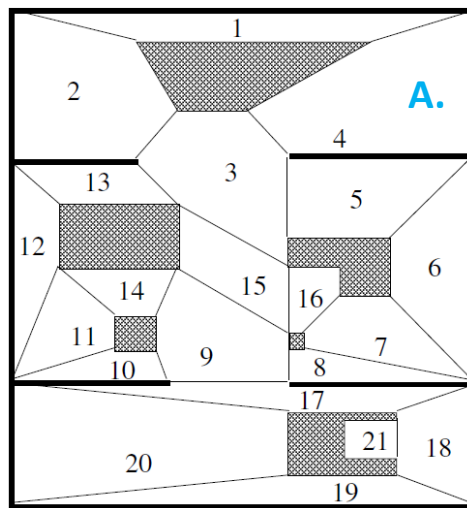
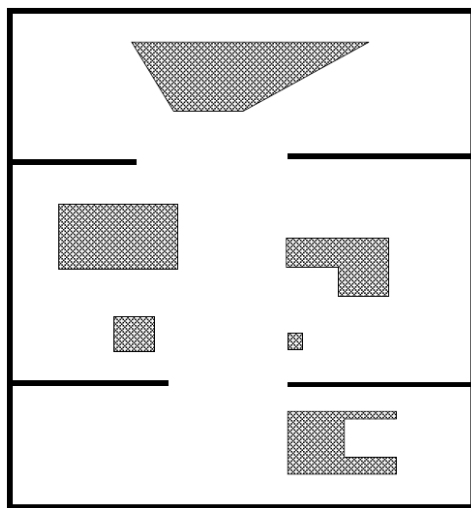
- linie autobusowe i przystanki autobusowe w mieście,
- sieć autostrad w danym kraju,
- sieć stacji i linii kolejowych dla metra lub systemu kolejowego,
- sieć linii wysokiego napięcia w danym kraju,
- kanalizacja miejska.

Typowe pytania, na które można odpowiedzieć za pomocą mapy topologicznej:

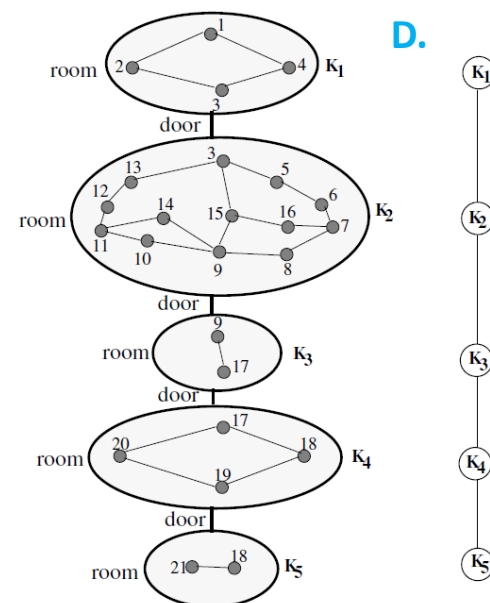
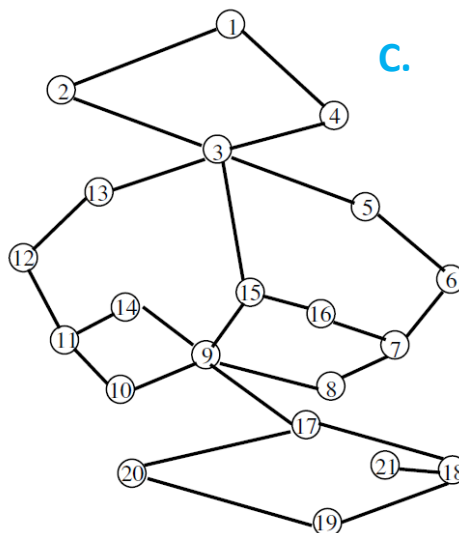
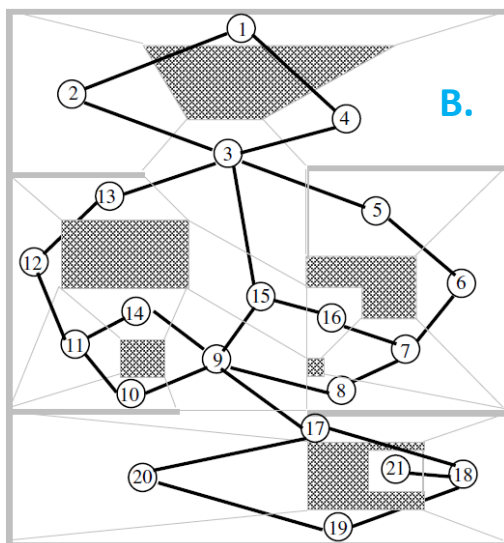
- Gdzie mogę zmienić autobusy między stacjami A i B?
- Czy mogę dotrzeć z A do B przez C lub przez D?
- Ile przystanków jest w drodze z punktu A do punktu B?
- Czy istnieją linie obejścia jeżeli jedna linia transmisyjna ulegnie awarii?
- Gdzie jest wejście do głównej linii kanalizacyjnej?

Uzyskanie odpowiedzi wyłącznie za pomocą map geometrycznych jest trudne;

Przekształcenie mapy geometrycznej na topologiczną



- A.** dekompozycja mapy na obszary wolne od obiektów poprzez połączenie narożników z krawędziami nie przecinającymi obiektów;
B. otrzymane obszary można przedstawić w postaci grafu; krawędzie oznaczają przejścia między obszarami;
C. graf można przekształcić w drzewo dekompozycji zawierające podgrafy reprezentujące kolejne pomieszczenia;
D. możliwa interpretacja drzewa dekompozycji;

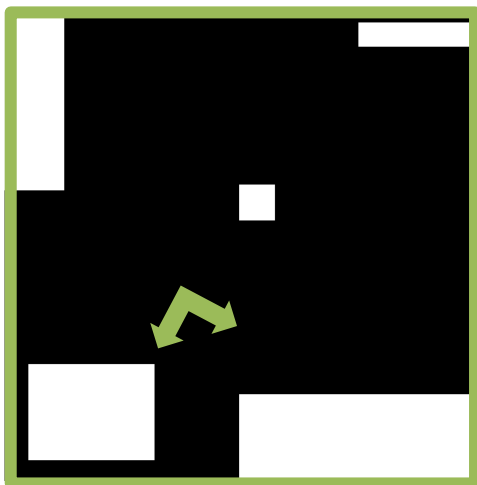


Problem mapowania

Odczyt skanera laserowego

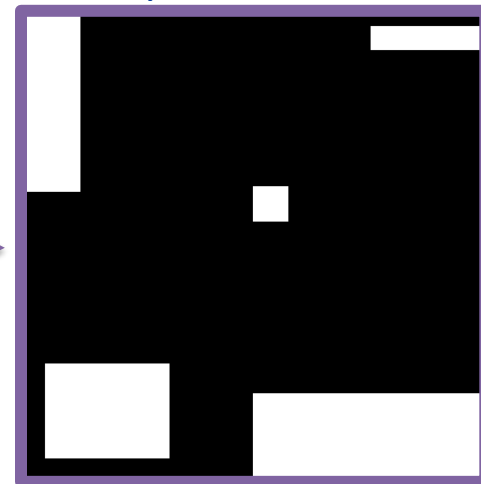


Pozycja i orientacja



Mapowanie

Mapa środowiska

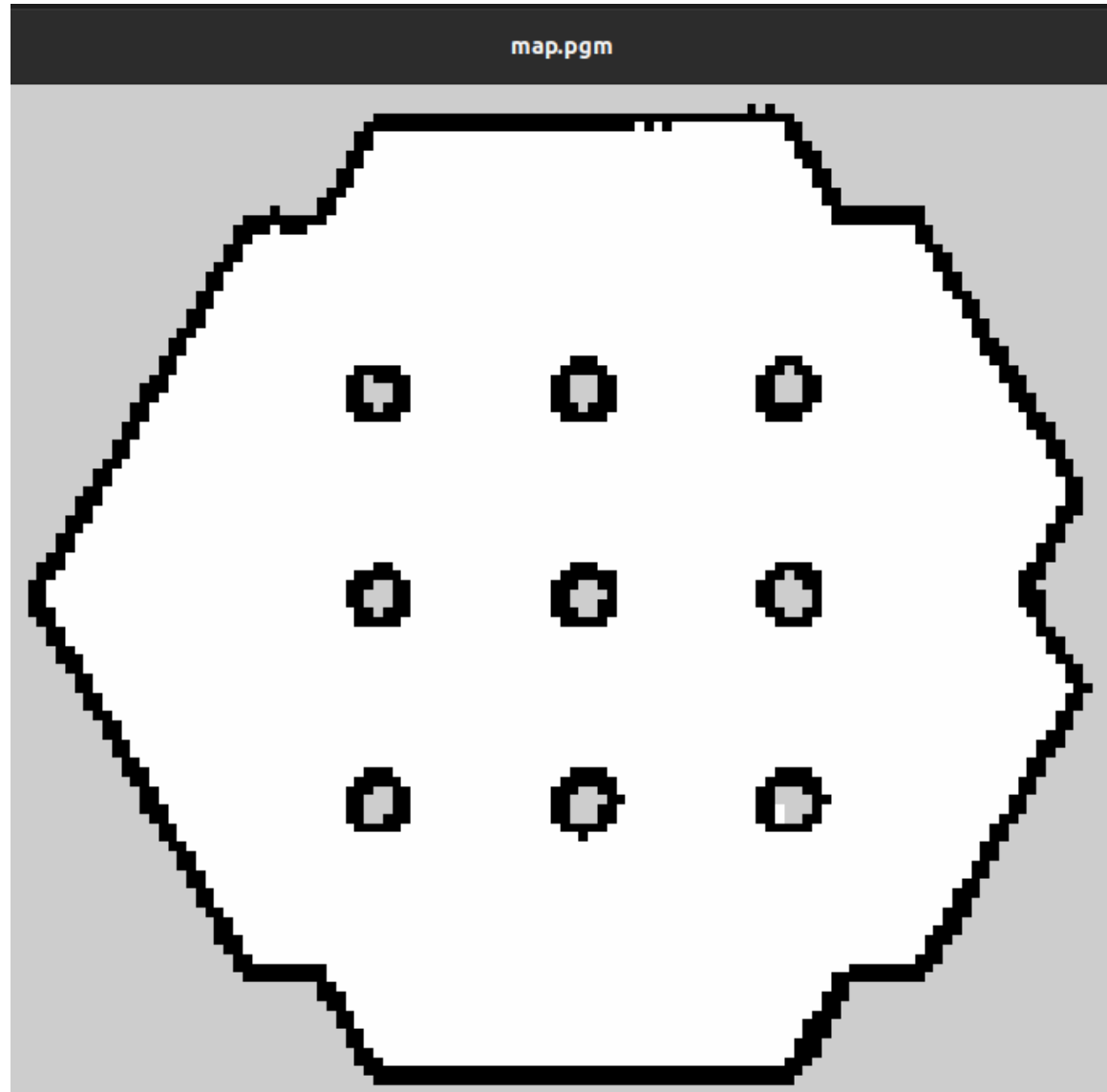


Siatka zajętości (ang. *occupancy grid*)

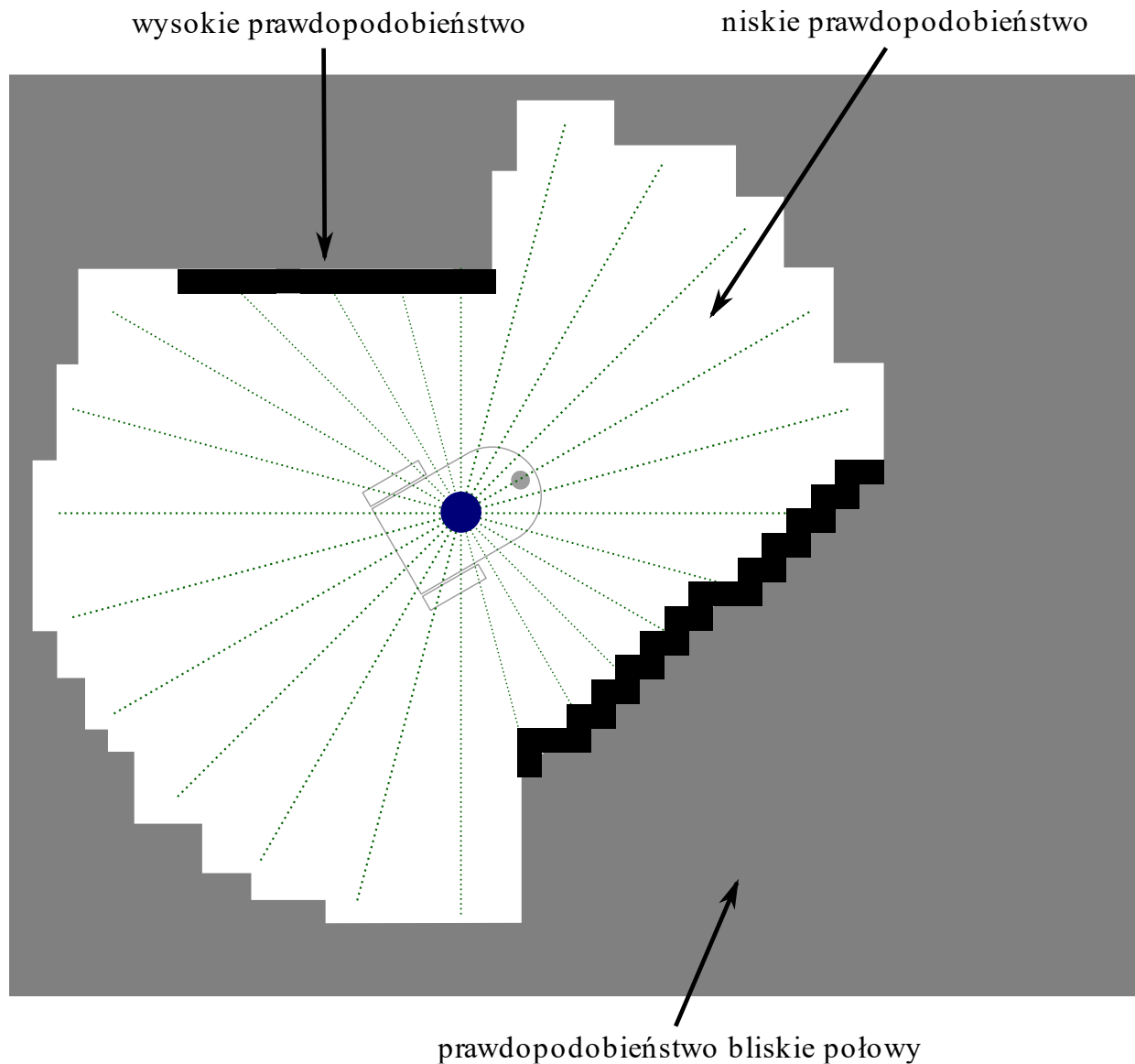
- Zdyskretyzowana siatka terenu 2D lub 3D,
- Zajętość (ang. *occupancy*) określamy obiektami statycznymi, obiekty dynamiczne nie powinny być uwzględniane na mapie,
- Każda komórka, tj. *pixel*, przyjmuje wartość $m^i \in \{0,1\}$

$m^i = 0$ (teraz wolny)

$m^i = 1$ (teraz zajęty)

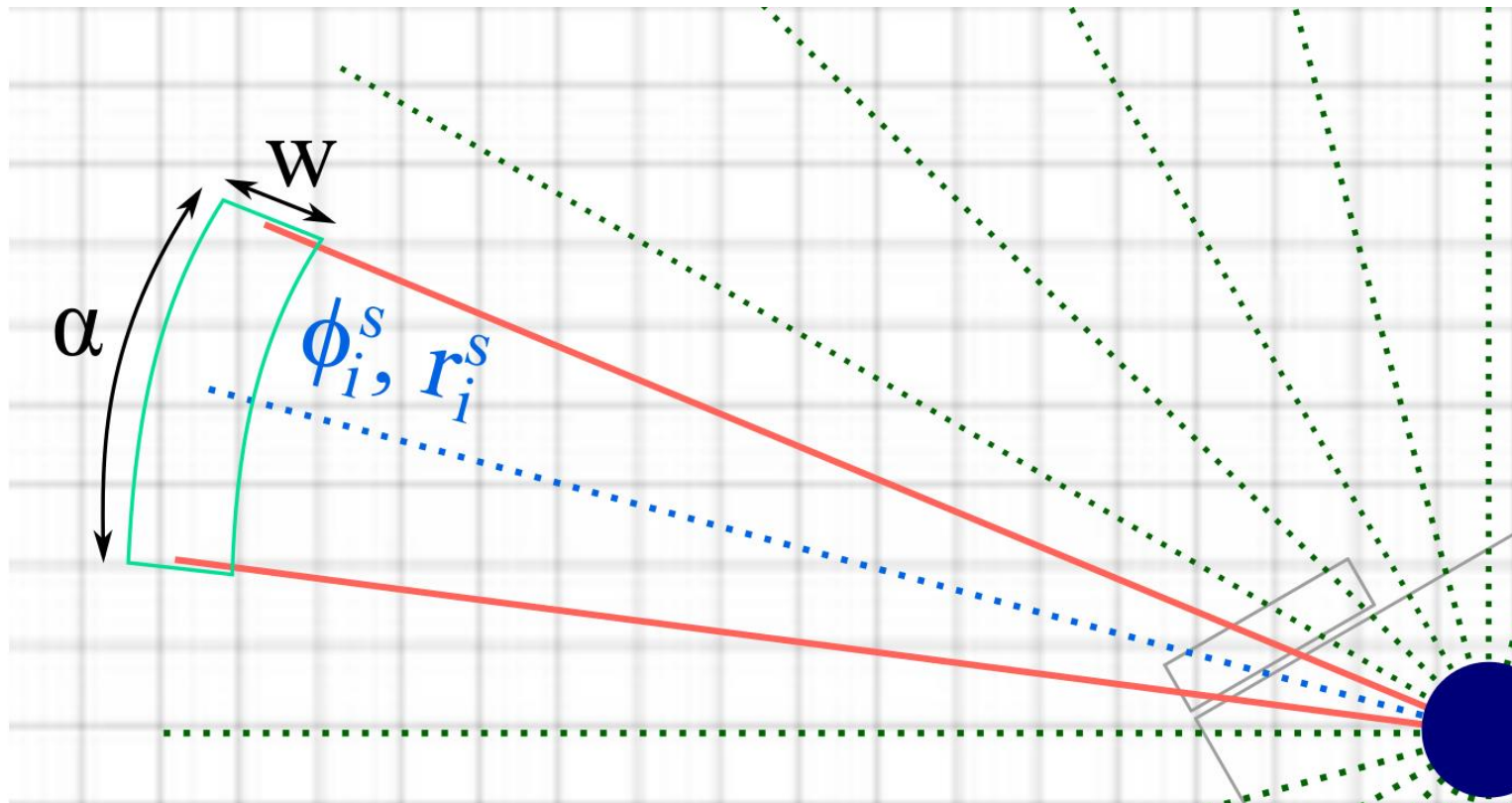


Odwrotny model pomiaru dla skanera laserowego LiDAR



Odwrotny model pomiaru dla skanera laserowego LiDAR

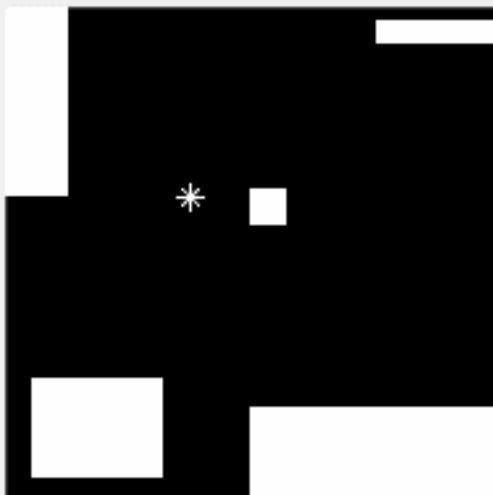
- α - definiuje kąt wpływu, dla którego są uwzględniane komórki mapy
- w - definiuje szerokość wpływu przy wykryciu przeszkody



Odwrotny model pomiaru dla skanera laserowego LiDAR

Problem! Bardzo duża złożoność obliczeniowa!

Environment map



Inverse Laser scan measurement

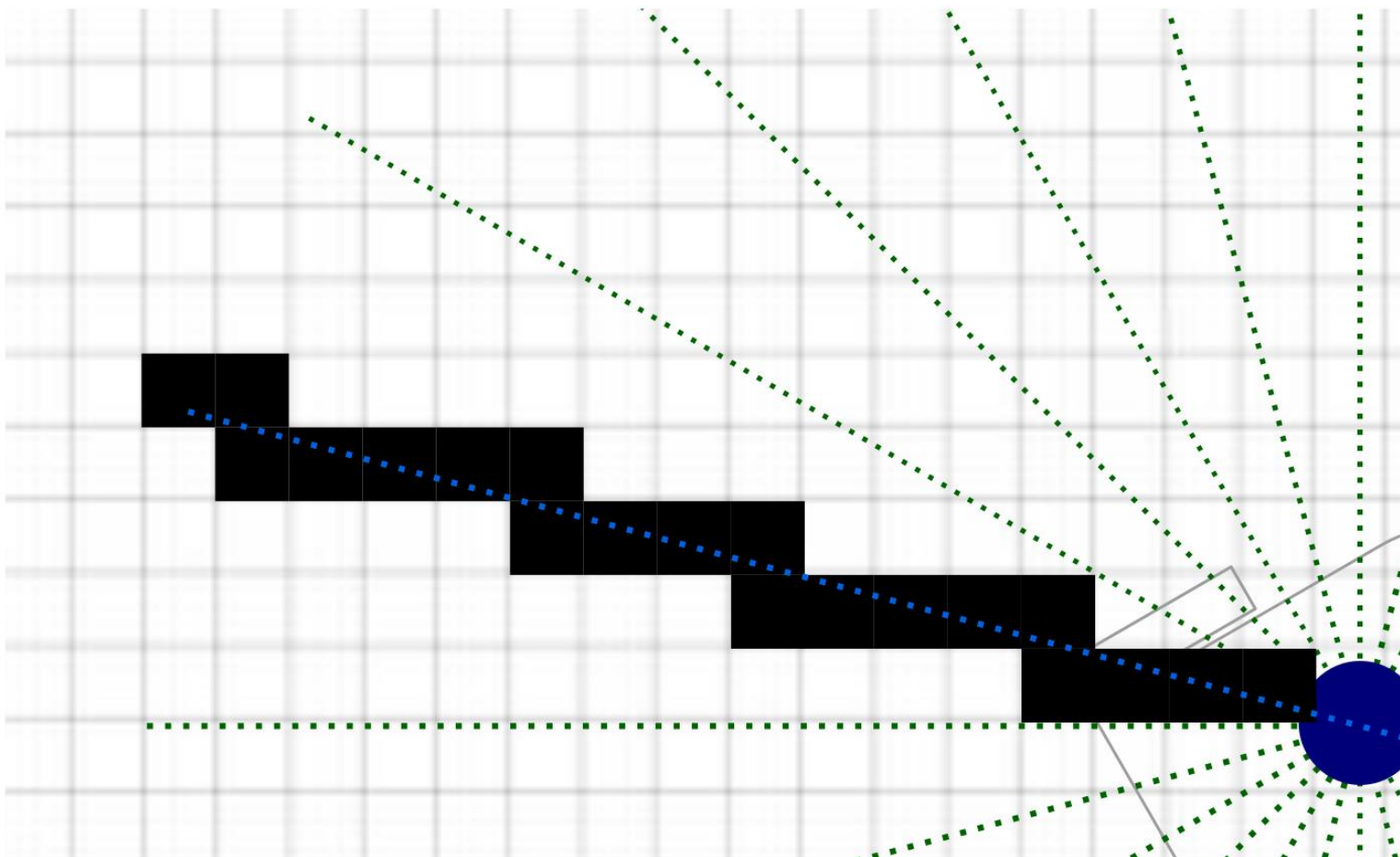


Builted map



Odwrotny model pomiaru dla skanera laserowego LiDAR

Algorytm śledzenie promieni

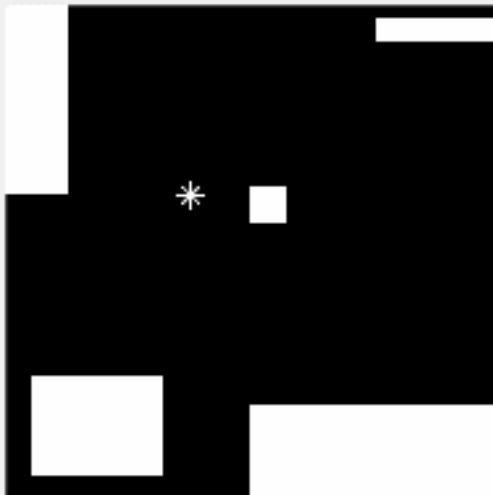


Odwrotny model pomiaru dla skanera laserowego LiDAR

Algorytm śledzenie promieni

Problem! Ograniczona dokładność!

Environment map



Inverse Laser scan measurement



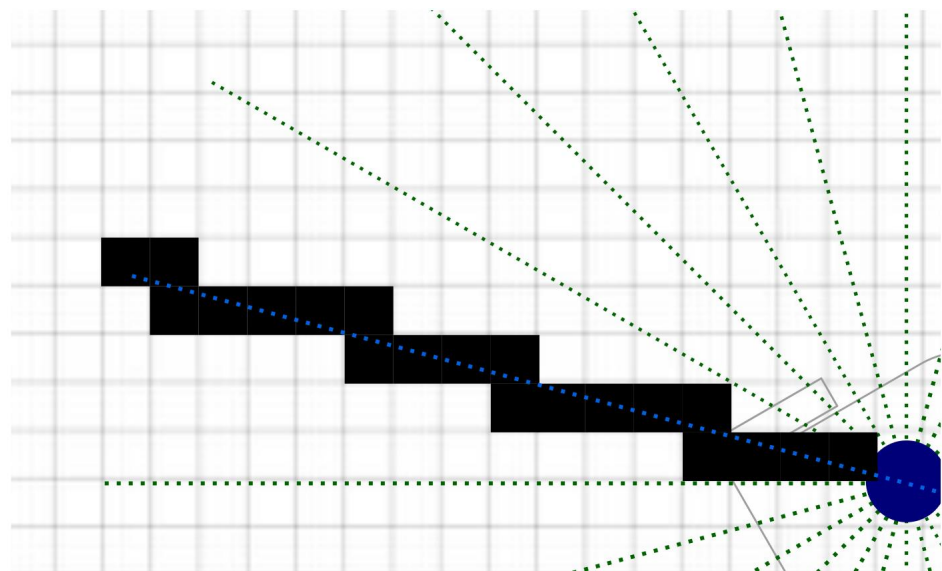
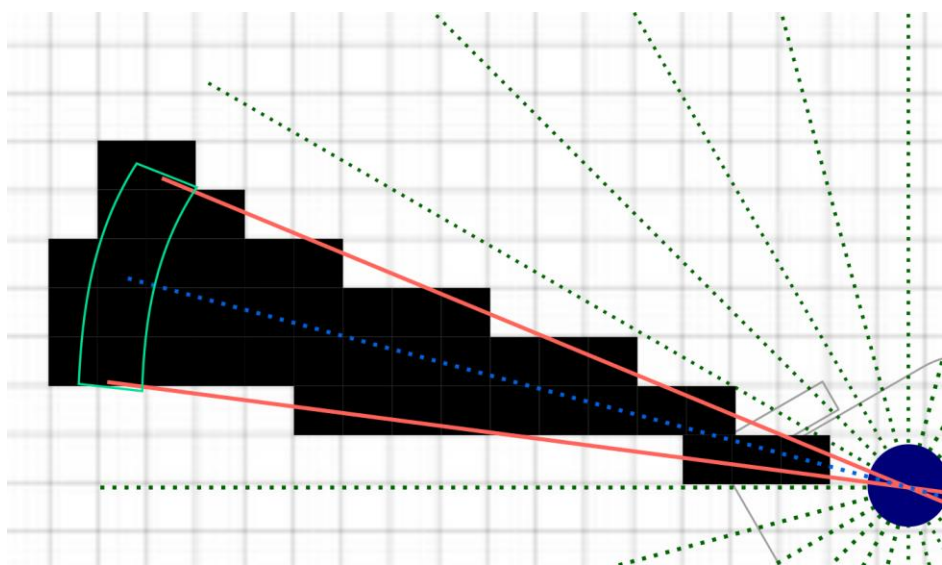
Builted map



Odwrotny model pomiaru dla skanera laserowego LiDAR

Algorytm śledzenie promieni

Alg. \ Wymiary mapy	80x80	160x160	800x800
Pełny odwrotny model	2.965804 s	11.159494 s	311.292021 s
Śledzenie promieni	0.144957 s	0.251876 s	4.624526 s

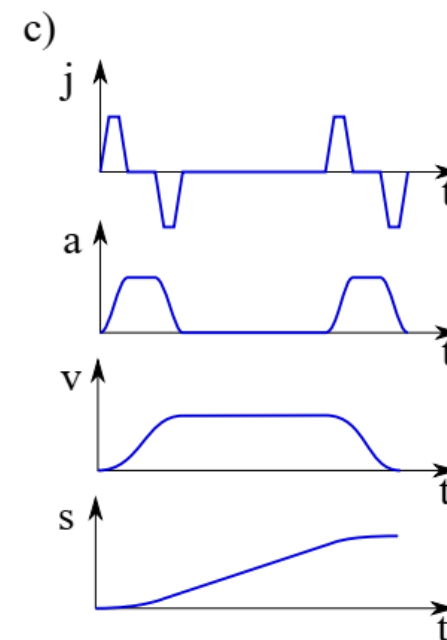
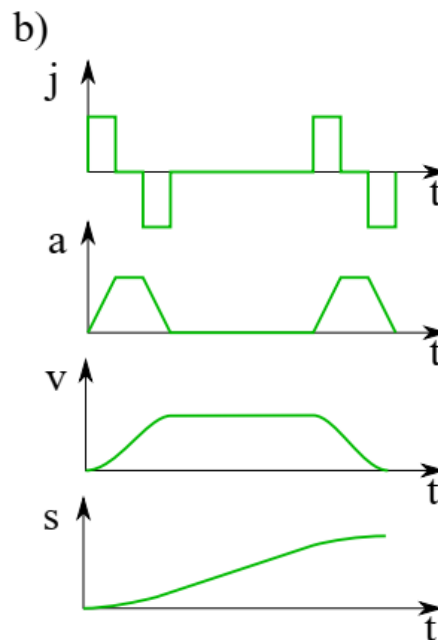
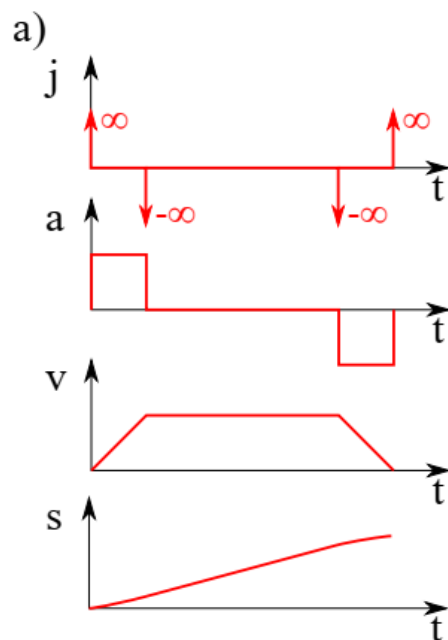


Planowanie trajektorii

Celem nawigowania jest prowadzenie pojazdu autonomicznego przez otoczenie;

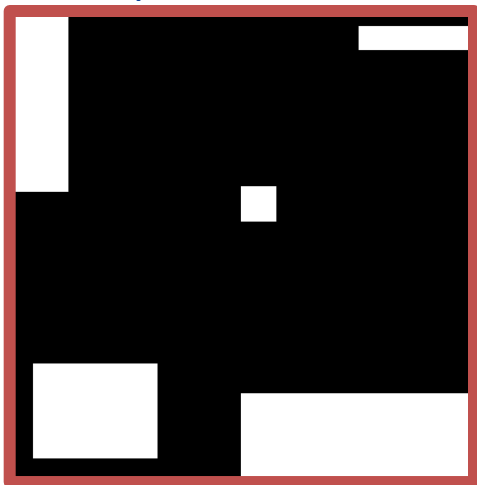
Zadanie nawigowania dzieli się na:

- **globalne planowanie trasy** – znalezienie właściwej trajektorii między punktem początkowym i końcowym przy użyciu danej reprezentacji środowiska,
- **lokalne planowanie trasy** – definiuje kolejne punkty trajektorii z uwzględnieniem wymiarów pojazdu oraz ograniczeń kinematycznych,
- **sterowanie** – generowanie właściwych sygnałów sterujących na podstawie pochodzących z lokalnego planowania trasy punktów referencyjnych;

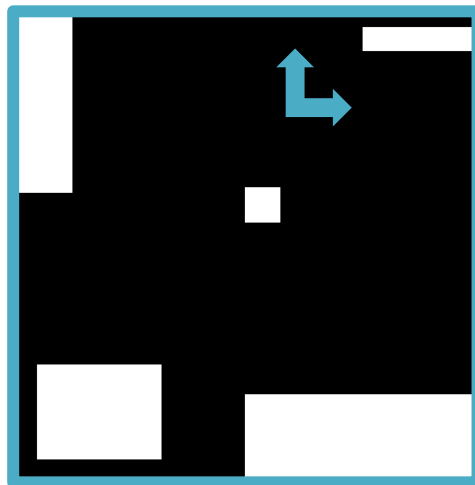


Problem planowania ścieżki globalnej

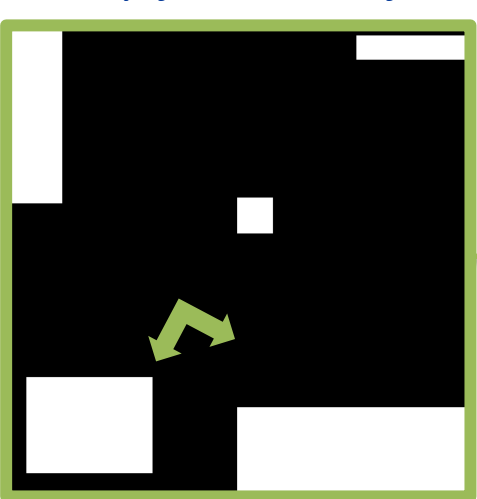
Mapa środowiska



Pozycja i orientacja zadana

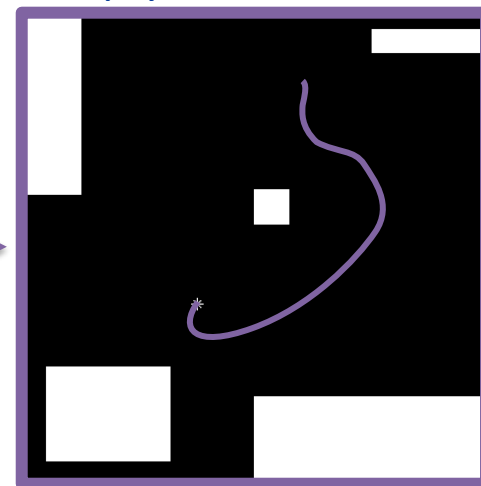


Pozycja i orientacja



Planowanie
ścieżki

Optymalna ścieżka



Problem planowania ścieżki globalnej

Algorytm eksplorowania drzew losowych (ang. Rapidly-Exploring Random Tree)

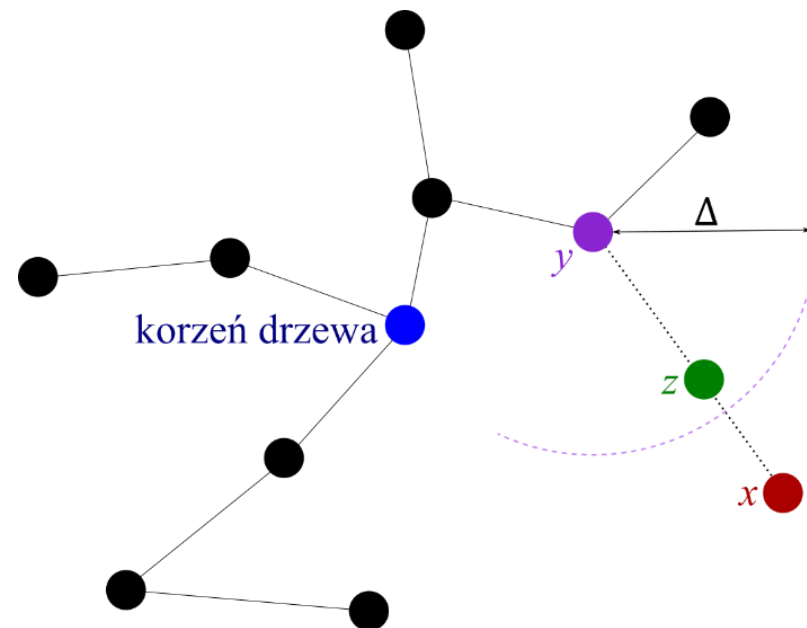
Algorytm eksplorowania drzew losowych (RRT) polega na konstrukcji grafu w sposób losowy. Algorytm RRT tworzy specjalny rodzaj grafu zwany drzewem. Bardzo często stosuje się dwa drzewa – jedno z korzeniem w aktualnej pozycji pojazdu, drugie z korzeniem w punkcie docelowym

Algorytm 6: Pseudokod algorytmu eksplorowania drzew losowych

Wejście: s (pozycja początkowa), d (pozycja docelowa)

Wyjście: ścieżka do celu

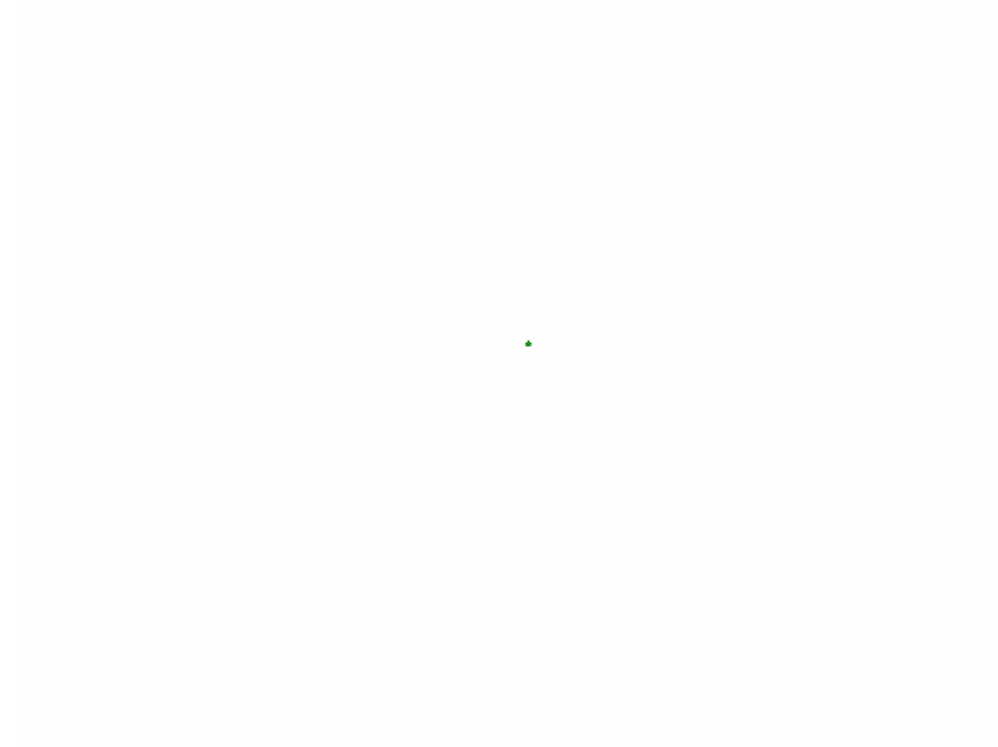
- 1 Stwórz nowe drzewo A i dodaj s jako jego korzeń
- 2 Stwórz nowe drzewo B i dodaj d jako jego korzeń
- 3 Dopóki drzewa A i B nie są połączone
- 4 Rozszerz drzewo A o pozycję x (patrz rysunek)
- 5 Znajdź najbliższego sąsiada y pozycji x w drzewie B
- 6 Jeżeli $LocalPlanner(x, y) = true$
 - 7 Zwróć ścieżkę łączącą pozycje wierzchołków:
 - od punktu x do korzenia w drzewie A
 - od punktu y do korzenia w drzewie B
 - 8 Koniec jeżeli
 - 9 Zamień drzewa A i B
 - 10 Koniec dopóki





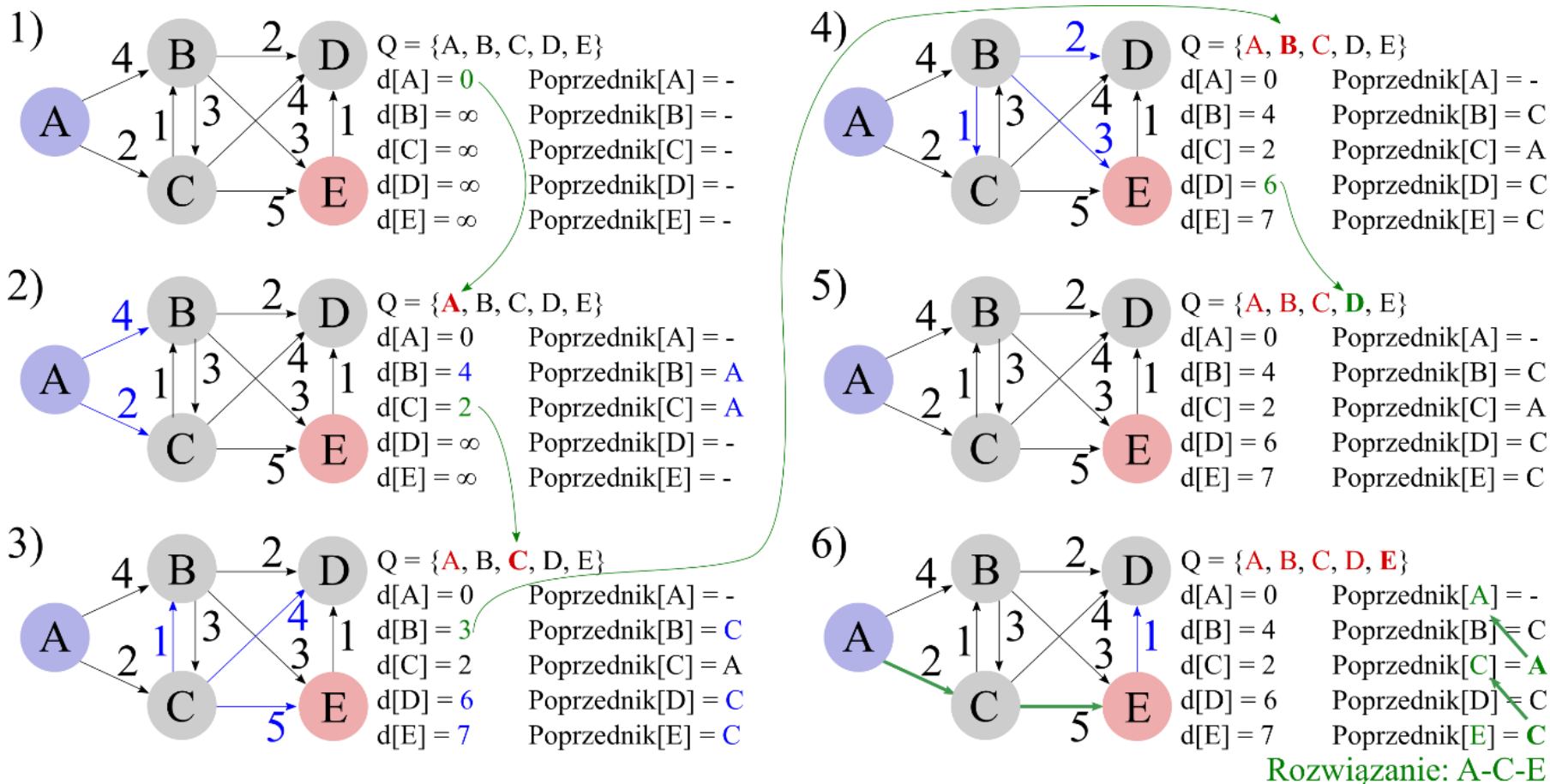
Problem planowania ścieżki globalnej

Algorytm eksplorowania drzew losowych (ang. Rapidly-Exploring Random Tree)



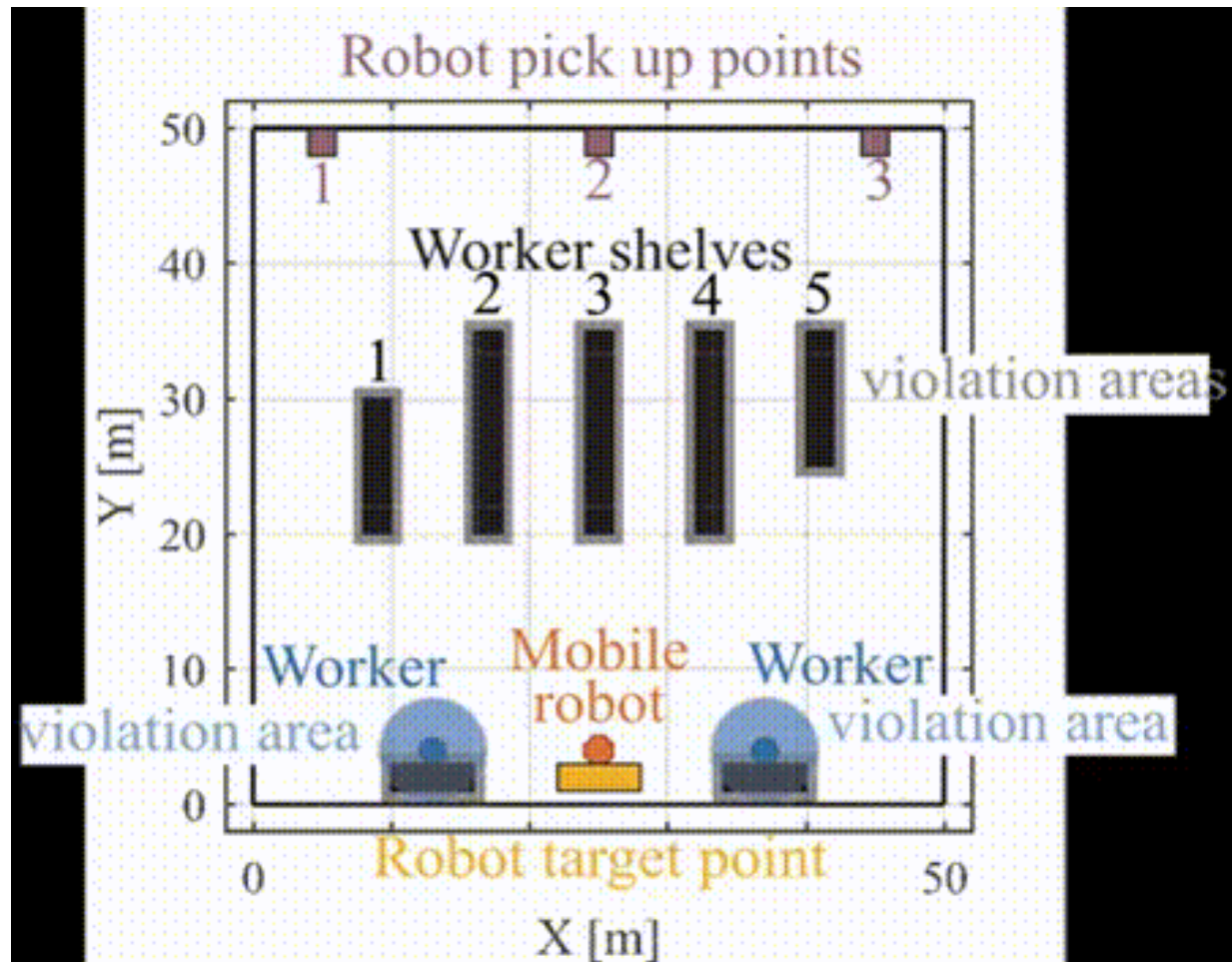
Problem planowania ścieżki globalnej

Algorytm Dijkstry



Problem planowania ścieżki globalnej

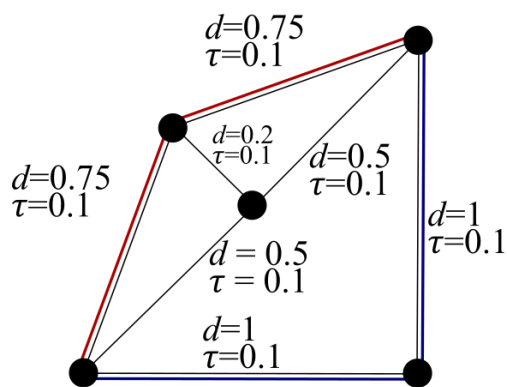
Algorytm Dijkstry



Problem planowania ścieżki globalnej

Algorytm mrówkowy

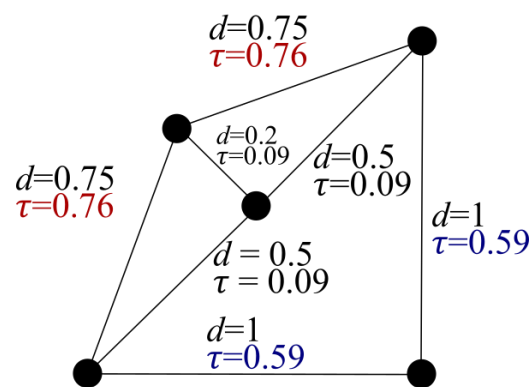
Trasy mrówek



 $L_1 = 0.75 + 0.75 = 1.5$
 $\Delta\tau^1 = 1/1.5 = 0.67$

 $L_2 = 1 + 1 = 2$
 $\Delta\tau^2 = 1/2 = 0.5$

Aktualizacja feromonu



$\rho = 0.1$
 $(1-0.1) * 0.1 + 0 = 0.09$
 $(1-0.1) * 0.1 + 0.67 = 0.76$
 $(1-0.1) * 0.1 + 0.5 = 0.59$

Algorytm 7: Pseudokod algorytmu mrówkowego

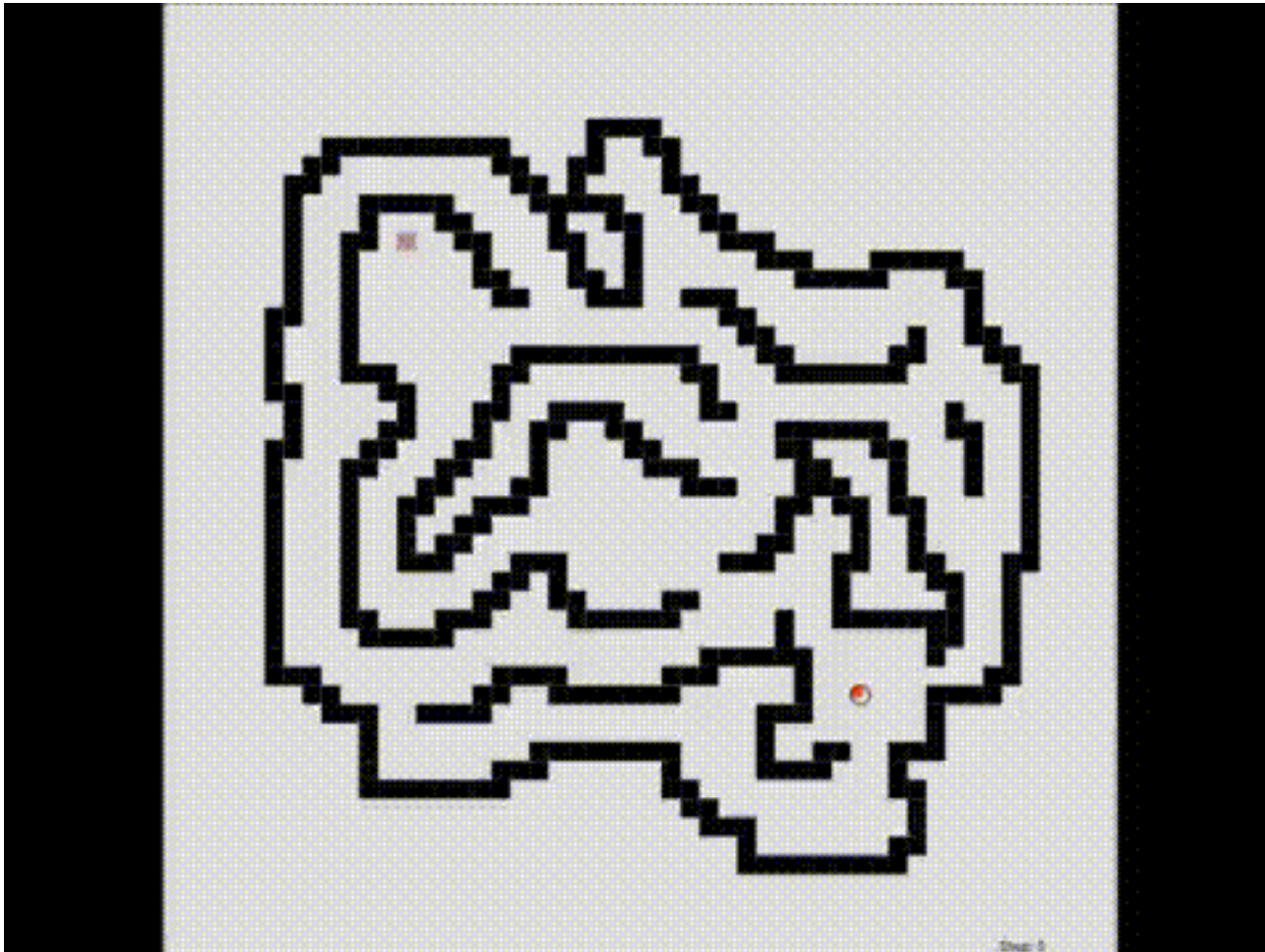
Wejście: s (pozycja początkowa), d (pozycja docelowa),
 N (populacja mrówek), M (liczba iteracji), Q , α ,
 β , ρ (parametry)

Wyjście: ścieżka do celu

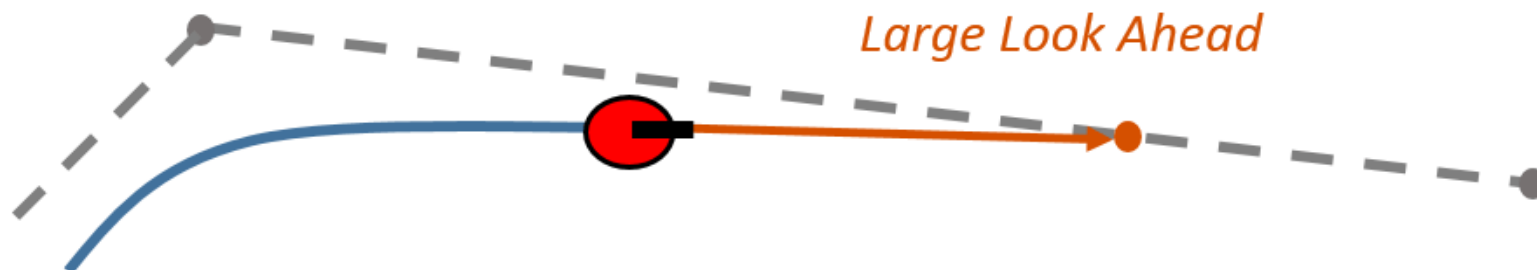
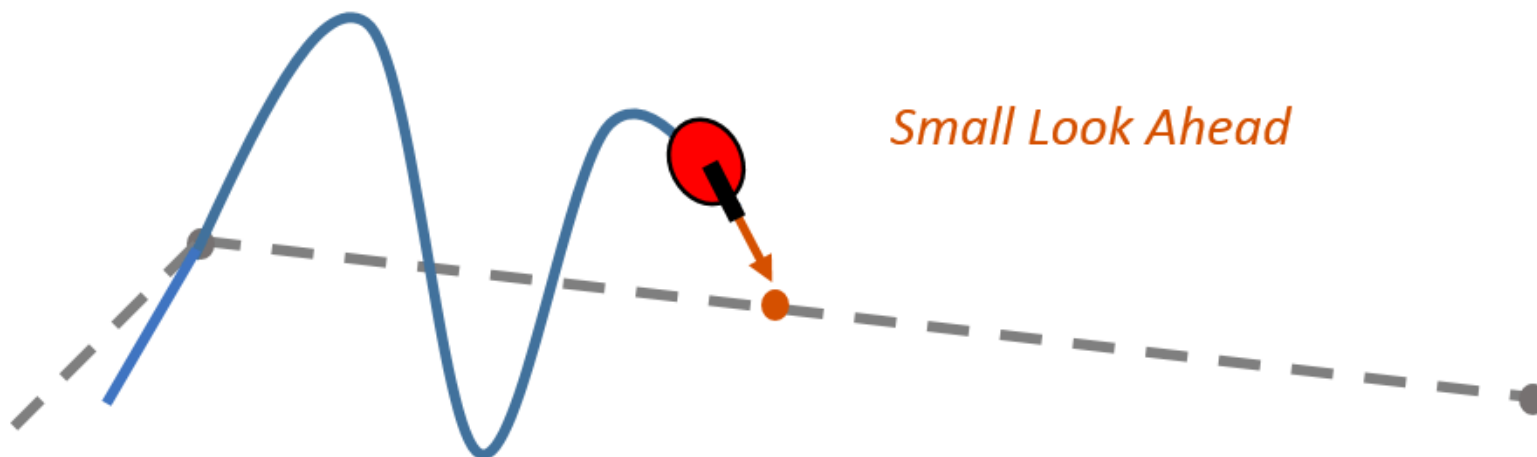
1	Inicjalizacja zmiennych
2	Dla $iter = 1 \dots M$ wykonaj
3	Dla $k = 1 \dots N$ wykonaj
4	Pozycja mrówki $\leftarrow s$
5	Dopóki mrówka nie osiągnęła punktu docelowego (d)
6	Oblicz prawdopodobieństwo dla każdego możliwego ruchu
7	Wybierz kolejną pozycję zgodnie z prawdopodobieństwem
8	Koniec dopóki
9	Aktualizuj ilość feromonów na ścieżkach
10	Koniec dla
11	Koniec dla
12	Zwróć ścieżkę wybierając trasę o największych prawdopodobieństwach

Problem planowania ścieżki globalnej

Algorytm mrówkowy

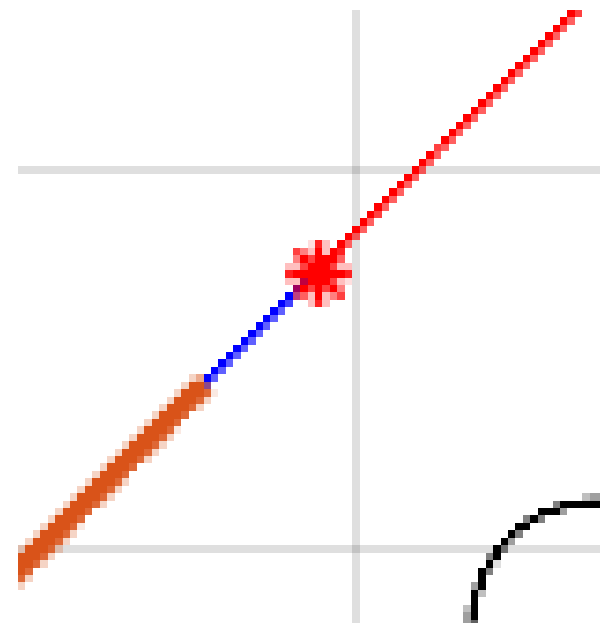


Śledzenie ścieżki globalnej (ang. *Waypoint tracking*)



Śledzenie ścieżki globalnej (ang. *Waypoint tracking*)

W celu bezkolizyjnej realizacji globalnej ścieżki należy zastosować algorytm planowania ścieżki lokalnej.

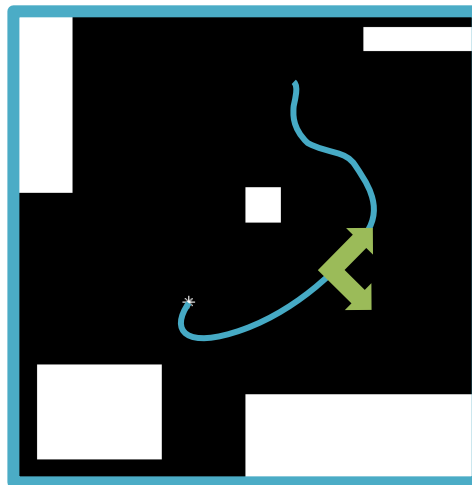


Problem planowania ścieżki lokalnej

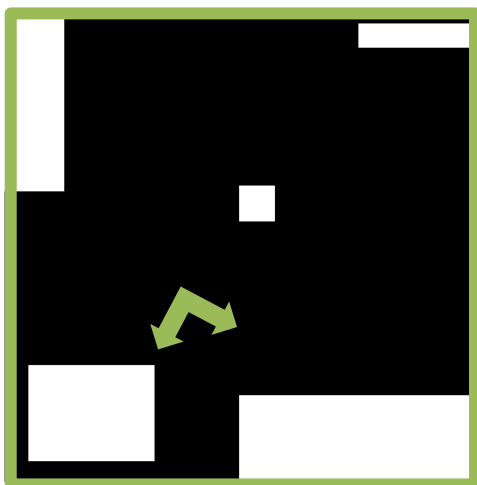
Odczyt skanera laserowego



Tymczasowa pozycja i orientacja zadana



Pozycja i orientacja



Omijanie
przeszkód

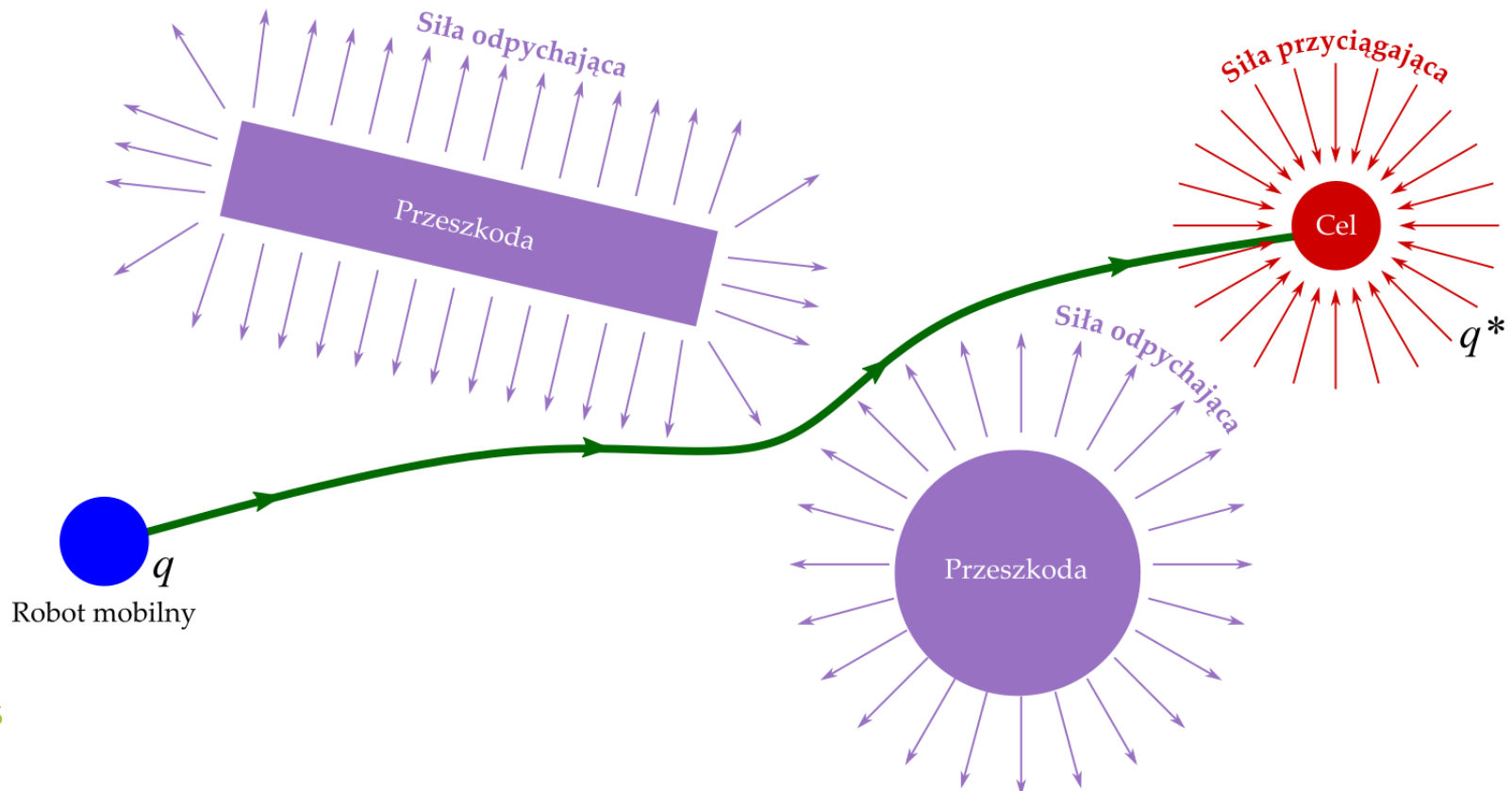
Bezkolizyjny ruch



Algorytm sztucznych pól potencjałowych

(ang. *Artificial Potential Field* algorithm, APF)

Algorytm inspirowany oddziaływaniem elektrostatycznym. Zakładając różne ładunki robota oraz punktu docelowego generowana jest siła przyciągająca, a takie same ładunki robota i przeszkód generują siłę odpychającą.





Algorytm sztucznych pól potencjałowych

Podstawowym równaniem algorytmu jest funkcja potencjału $U : R^N \rightarrow R$, wyrażona następującym równaniem:

$$U(q) = U_{att}(q) + U_{rep}(q)$$

gdzie:

q to pozycja robota,

$U_{att}(q)$ jest potencjałem przyciągającym (ang. *attractive*),

$U_{rep}(q)$ jest potencjałem odpychającym (ang. *repulsive*).

Wartość funkcji potencjału może być interpretowana jako energia, a wtedy gradient potencjału jest siłą. Gradient funkcji potencjału jest następujący:

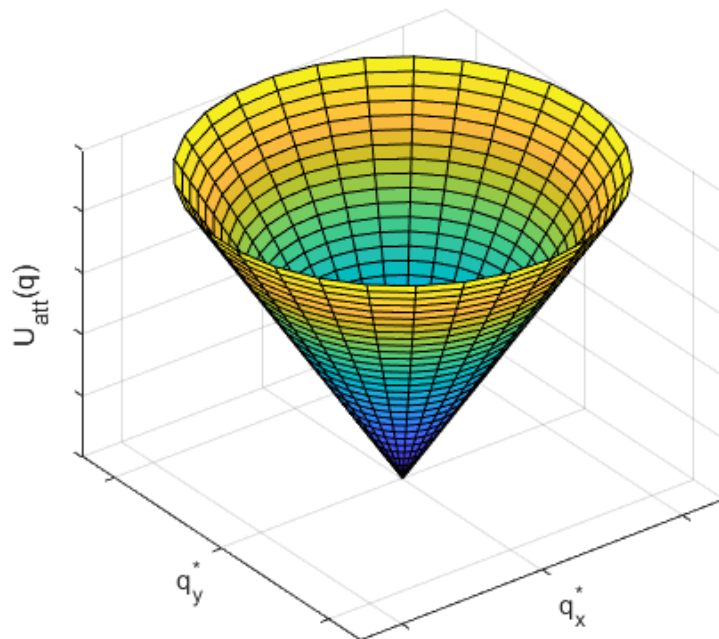
$$\nabla U(q) = \nabla U_{att}(q) + \nabla U_{rep}(q)$$

APF – potencjał przyciągający (ang. *attractive potential*)

Celem tego potencjału jest przyciąganie robota (o pozycji q) do punktu docelowego (q^*), czyli minimalizacja błędu $e(q) = q - q^*$. Rozważmy dwie funkcje: stożkową oraz paraboliczną.

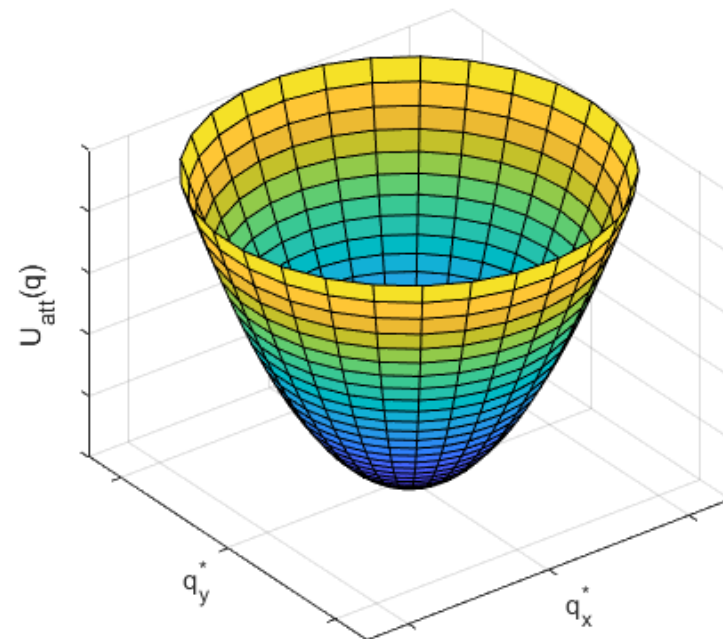
$$U_{att}(q) = \zeta ||e(q)||$$

$$\nabla U_{att}(q) = \zeta \frac{e(q)}{||e(q)||}$$



$$U_{att}(q) = \frac{1}{2} \zeta ||e(q)||^2$$

$$\nabla U_{att}(q) = \zeta e(q)$$





APF – potencjał przyciągający (ang. *attractive potential*)

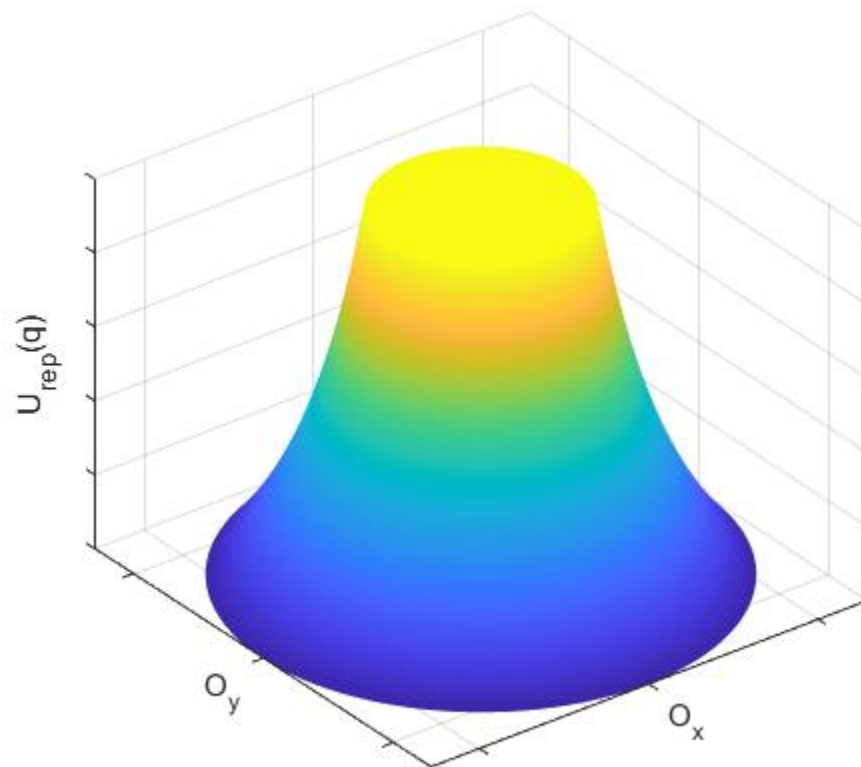
$$U_{att}(q) = \begin{cases} \frac{1}{2}\zeta||e(q)||^2, & \text{jeżeli } ||e(q)|| \leq d_g^* \\ d_g^*\zeta e(q) - \frac{1}{2}\zeta(d_g^*)^2, & \text{w pozostałych przypadkach} \end{cases}$$

gdzie: d_g^* jest wartością graniczną, tj. od tej odległości długość wektora gradientu maleje z razem z odległością od punktu docelowego pozwalając na płynne zatrzymanie się robota.

$$\nabla U_{att}(q) = \begin{cases} \zeta e(q), & \text{jeżeli } d_g \leq d_g^* \\ d_g^*\zeta \frac{e(q)}{||e(q)||}, & \text{w pozostałych przypadkach} \end{cases}$$

APF – potencjał odpychający (ang. *repulsive potential*)

Celem tego potencjału jest zapewnienie bezkolizyjnego poruszania się po nieznanym środowisku. Potencjał jest odwrotnie proporcjonalny do odległości od przeszkody. Dzięki temu jesteśmy w stanie zapewnić bezkolizyjny ruch, ponieważ przy odległości dążącej do zera wartość potencjału dąży do nieskończoności. W ten sposób minimalizujemy odległość od przeszkody (ang. *clearance*) $c_{oi}(q) = q - O_i$





APF – potencjał odpychający (ang. *repulsive potential*)

Warto podkreślić, że wprowadzony zostaje dodatkowy parametr Q^* , który odpowiada za zakres oddziaływania przeszkód na robota. Pozwala to na zignorowanie przeszkód w znacznej odległości od robota, dzięki czemu robot będzie się poruszał wprost do punktu docelowego. W celu zachowania płynności przejścia, musi on zostać uwzględniony w funkcji potencjału:

$$U_{rep}^{O_i}(q) = \begin{cases} \frac{1}{2} \eta \left(\frac{1}{\|c_{O_i}\|} - \frac{1}{Q^*} \right)^2, & \text{jeżeli } \|c_{O_i}\| \leq Q^* \\ 0, & \text{w pozostałych przypadkach} \end{cases}$$

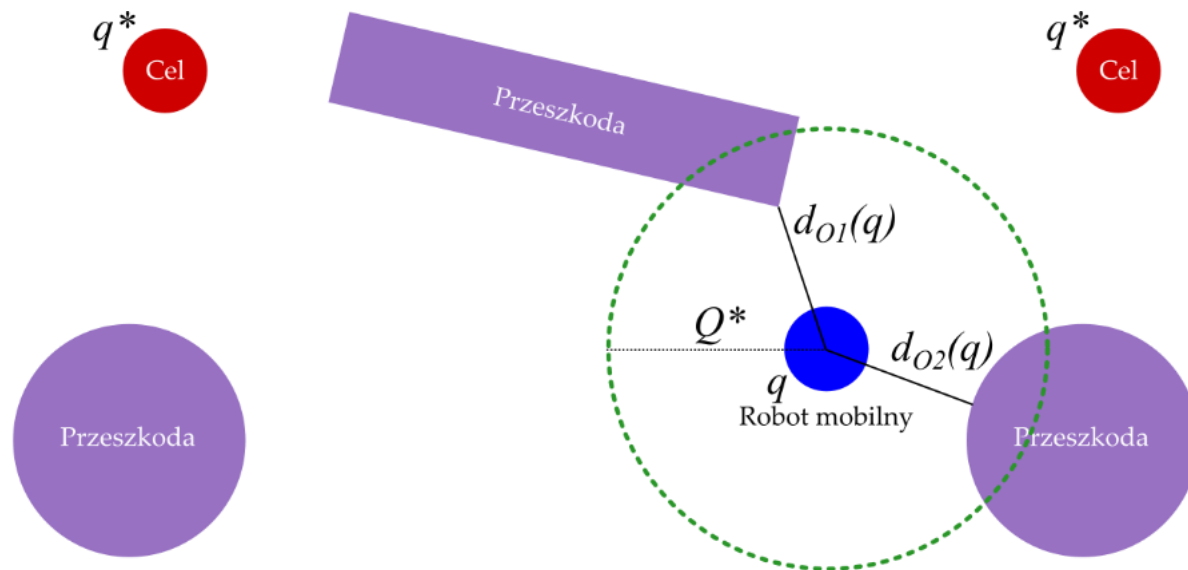
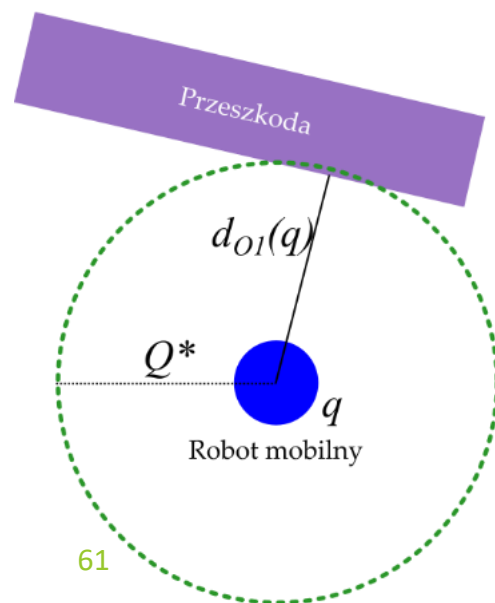
Powyższe wyrażenie prowadzi do następującego gradientu:

$$\nabla U_{rep}^{O_i}(q) = \begin{cases} \eta \left(\frac{1}{Q^*} - \frac{1}{\|c_{O_i}\|} \right) \frac{c_{O_i}}{\|c_{O_i}\|^2}, & \text{jeżeli } \|c_{O_i}\| \leq Q^* \\ 0, & \text{w pozostałych przypadkach} \end{cases}$$

APF – potencjał odpychający (ang. *repulsive potential*)

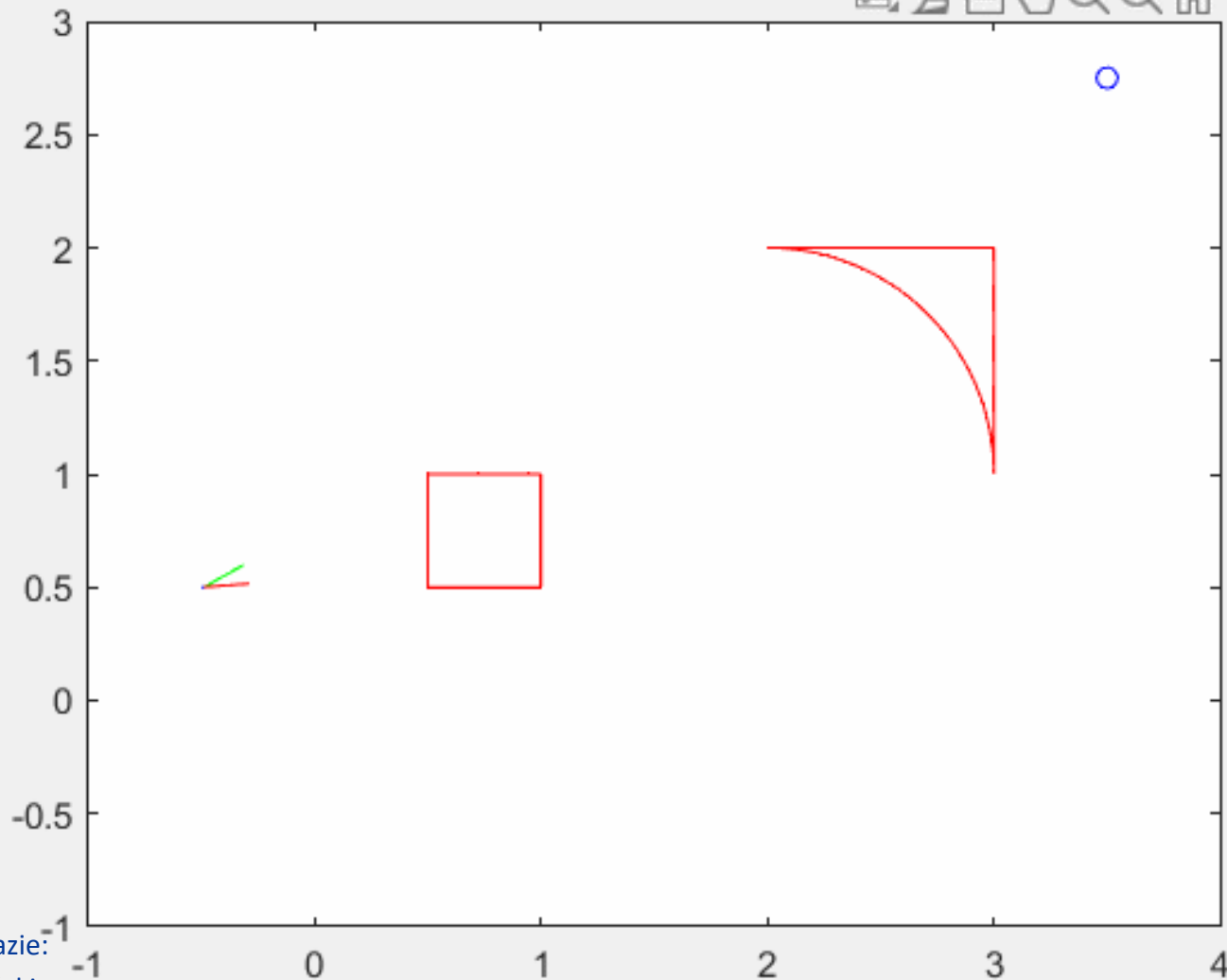
Potencjał odpychający jest sumą wszystkich potencjałów od każdej z przeszkód w analizowanym zakresie:

$$\nabla U_{rep}(q) = \sum_{i=1}^N \nabla U_{rep}^{O_i}(q)$$



Algorytm sztucznych pól potencjałowych

(ang. *Artificial Potential Field* algorithm, APF)

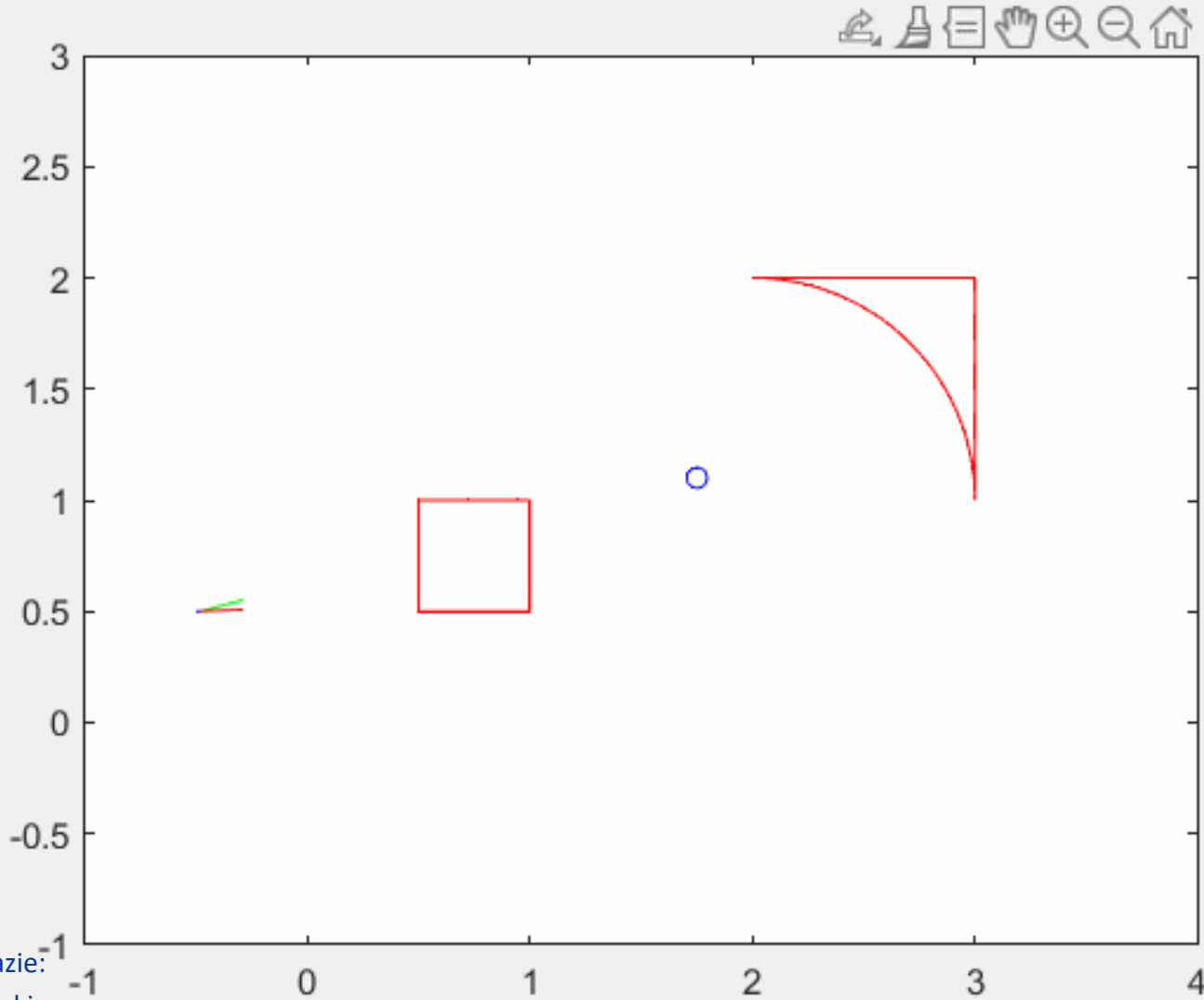


Wygenerowano na bazie:

Autor: Rafał Szczepański

Link: <https://www.mathworks.com/matlabcentral/fileexchange/126455-artificial-potential-field-path-planning-algorithm>

APF – Przykład 2

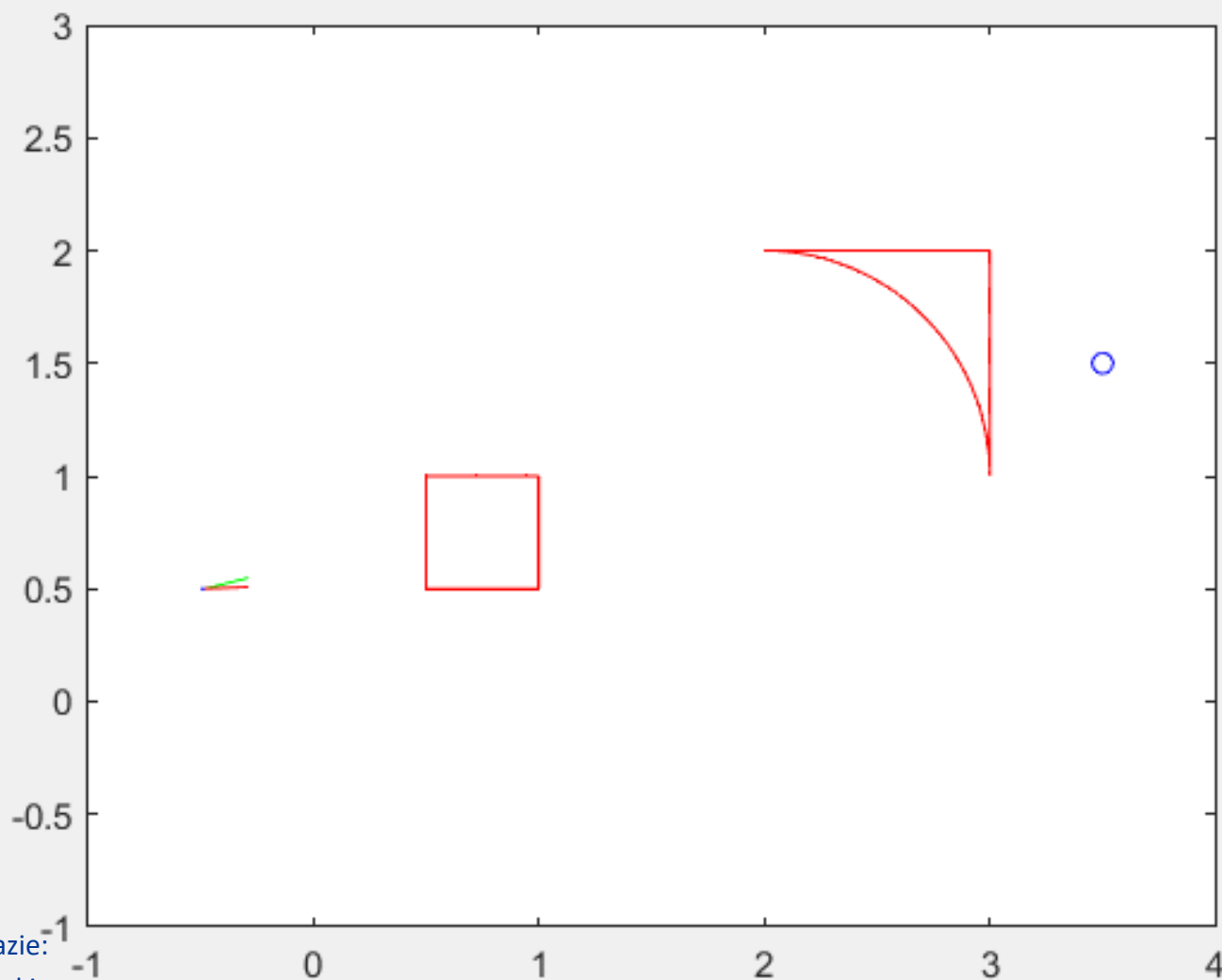


Wygenerowano na bazie:

Autor: Rafał Szczepański

Link: <https://www.mathworks.com/matlabcentral/fileexchange/126455-artificial-potential-field-path-planning-algorithm>

APF – Przykład 3



Wygenerowano na bazie:

Autor: Rafał Szczepański

Link: <https://www.mathworks.com/matlabcentral/fileexchange/126455-artificial-potential-field-path-planning-algorithm>



Algorytm dynamicznego okna

(ang. *Dynamic Window Approach, DWA*)

Jest to jeden z podstawowych algorytmów planowania ścieżki lokalnej, który jest jedną z bibliotek ROS. Sam algorytm można podzielić na dwie główne części:

- Przeszukiwanie przestrzeni rozwiązań,
- Optymalizacja.

Polega na predykcji przyszłych ruchów robota oraz sprawdzaniu ew. kolizji. Predykcja jest ograniczona do dynamicznego okna uwzględniając o ile może zostać zmieniona prędkość liniowa oraz kątowa względem aktualnej wartości. Dzięki temu algorytm zapewnia płynność ruchu.

Następnie dokonywana jest optymalizacja, czyli wybór najlepszego rozwiązania względem predefiniowanych kryteriów.



DWA – tworzenie zbioru możliwych sygnałów sterujących

Pierwsza część jest odpowiedzialna za przygotowanie dyskretnego zbioru możliwych sygnałów sterujących, tj. prędkość liniowa i kątowa. Można wyróżnić w niej trzy elementy:

- **Trajektorie kołowe:** założeniem algorytmu dynamicznego okna jest rozważanie trajektorii kołowych, które można zdefiniować jako pary prędkości liniowej i kątowej (v, ω) . Pozwala to zredukować planowanie ścieżki do dwuwymiarowego problemu.
- **Dopuszczalne prędkości:** wymaganiem rozważania danej pary prędkości jest bezpieczeństwo uzyskanej trajektorii, tj. brak kolizji z przeszkodą. Warunkiem dopuszczenia pracy (v, ω) jest możliwość zatrzymania się robota zanim osiągnie najbliższą przeszkodę na wygenerowanej trajektorii.
- **Dynamiczne okno:** zakładamy ograniczone przemieszczenia robota. W związku z tym uwzględniając aktualne prędkości robota (v_R, ω_R) prędkości mogą się zmieniać w ograniczonym zakresie zdefiniowanym przez maksymalne przyśpieszenie liniowe i kątowe.



DWA - optymalizacja

Druga część, tj. optymalizacja, jest związana z maksymalizacją funkcji celu wyrażonej równaniem:

$$G(v, \omega) = \sigma(\alpha \cdot heading(v, \omega) + \beta \cdot dist(v, \omega) + \gamma \cdot vel(v, \omega))$$

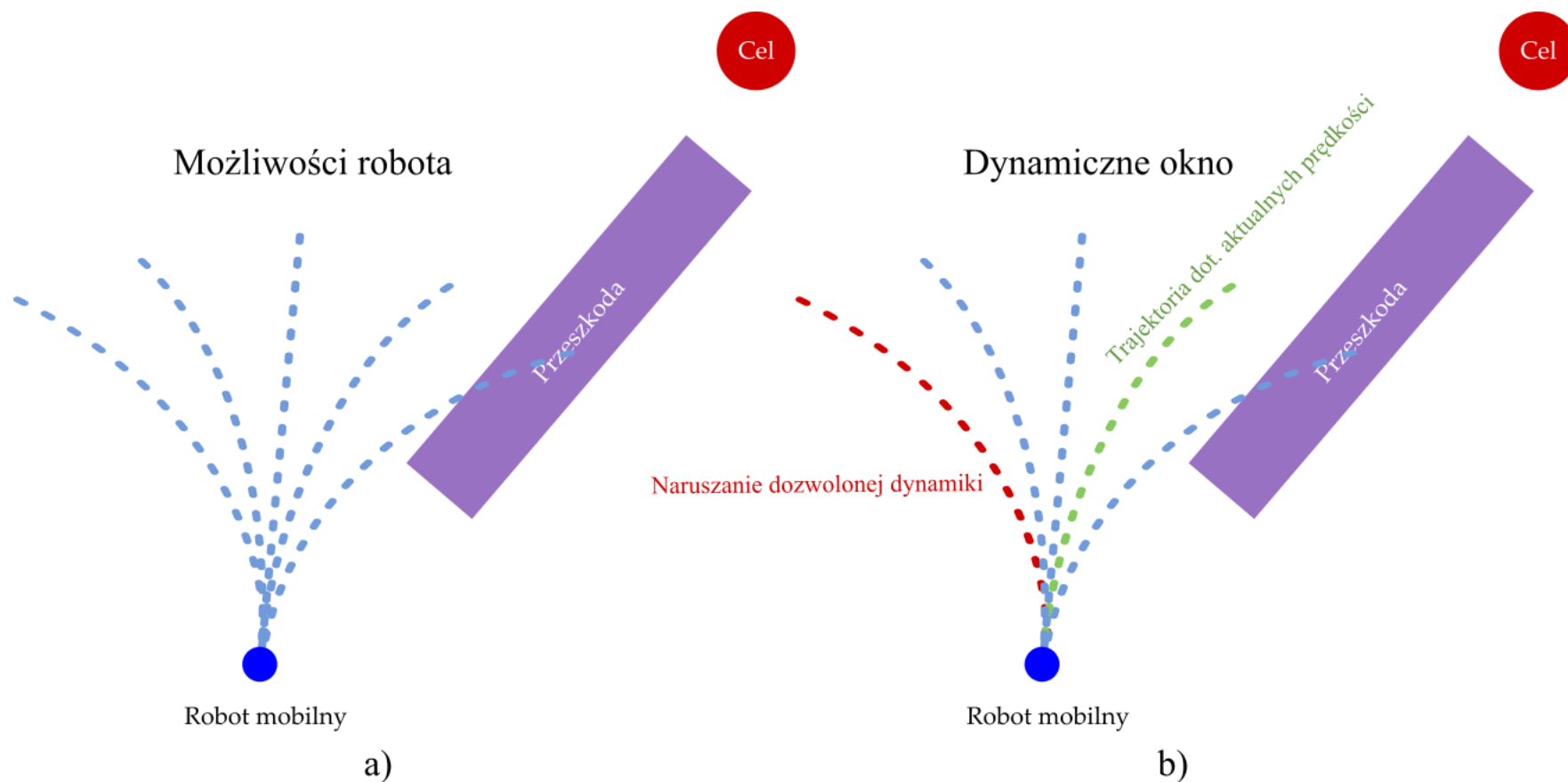
gdzie: α , β i γ to parametry funkcji celu, a σ jest funkcją wygładzającą.

Powyższa funkcja celu jest złożona z następujących elementów:

- **Docelowy kierunek** (ang. *target heading*): jest to nagroda za poruszanie się na wprost celu, tj. jazda na wprost celu dostarcza maksymalną wartość
- **Prześwit** (ang. *clearance*): funkcja *dist* jest najbliższym punktem do przeszkody dla trajektorii. Im mniejsza odległość do przeszkody tym większa szansa, że robot będzie próbował objechać przeszkodę dookoła.
- **Prędkość** (ang. *velocity*): jest to nagroda w funkcji celu za wysoką prędkość liniową robota.

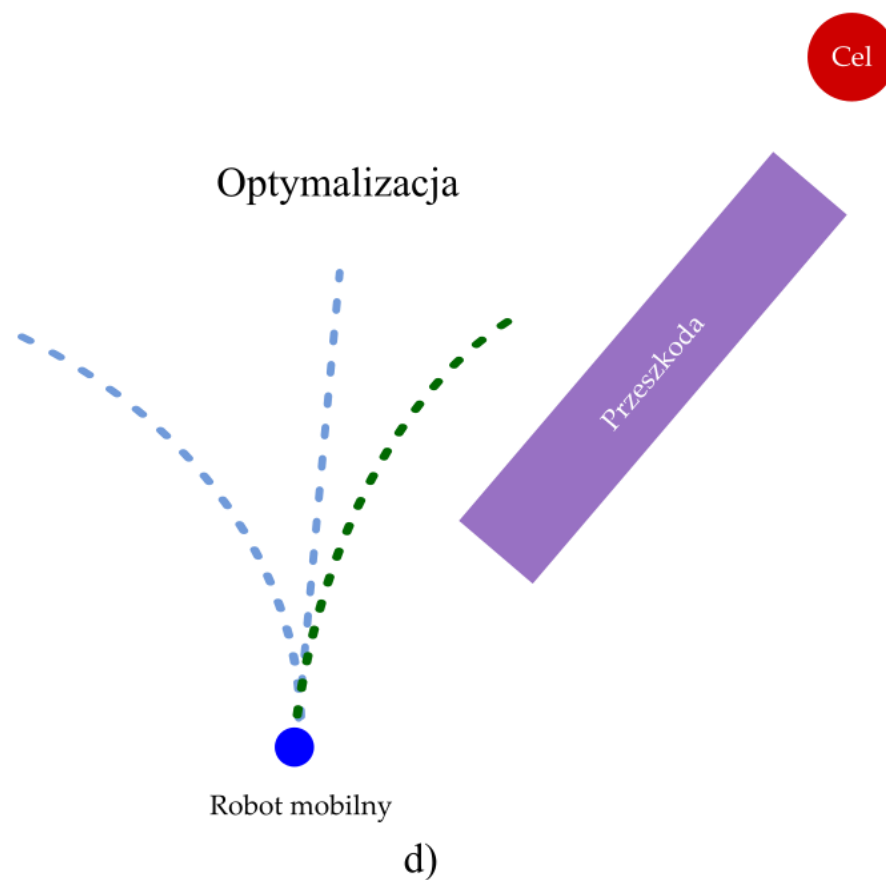
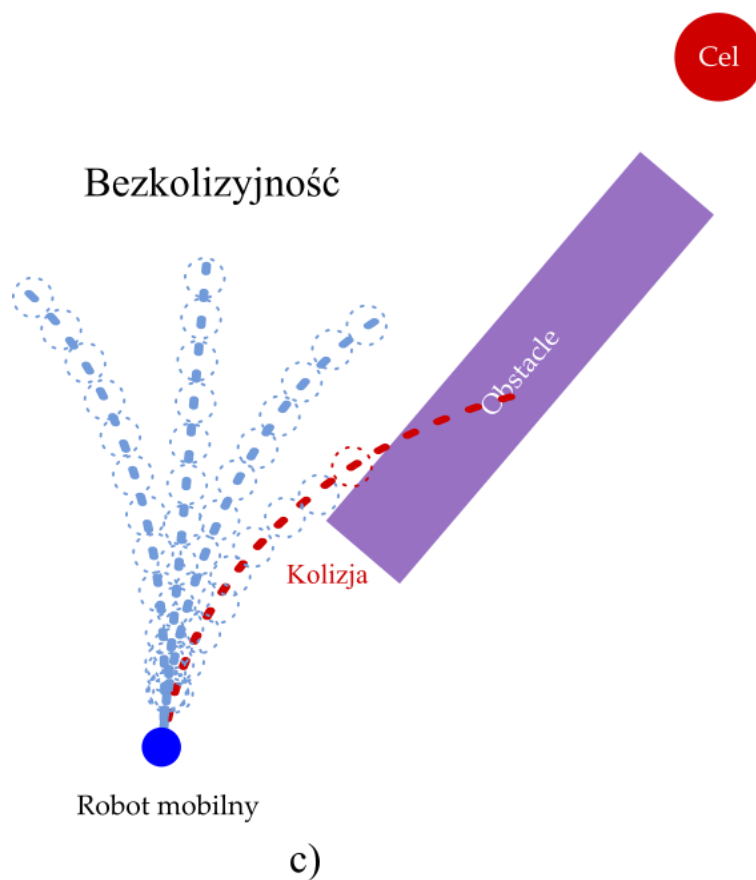
Algorytm dynamicznego okna

(ang. *Dynamic Window Approach, DWA*)



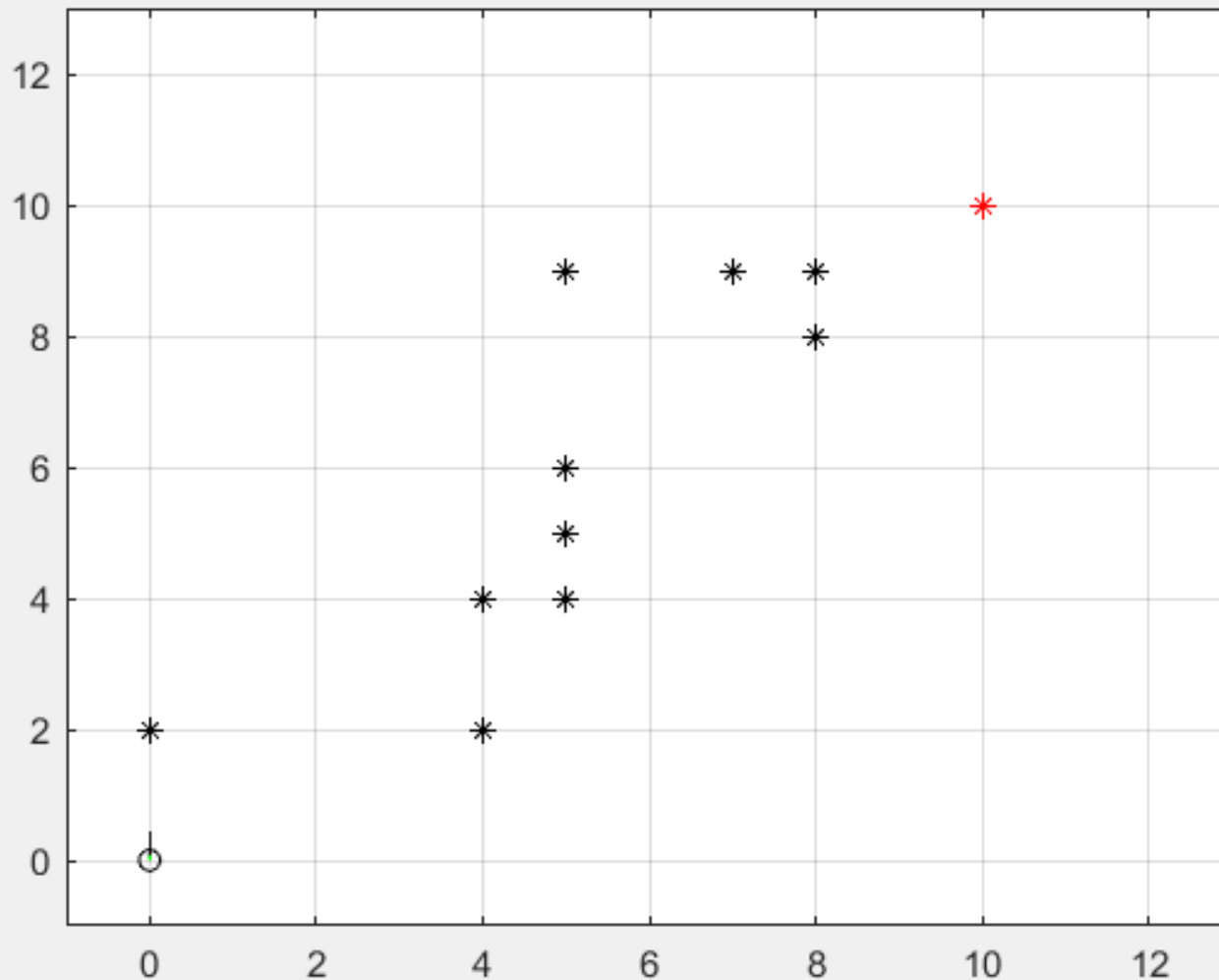
Algorytm dynamicznego okna

(ang. *Dynamic Window Approach, DWA*)





DWA – Przykład 1



Wygenerowano na bazie:

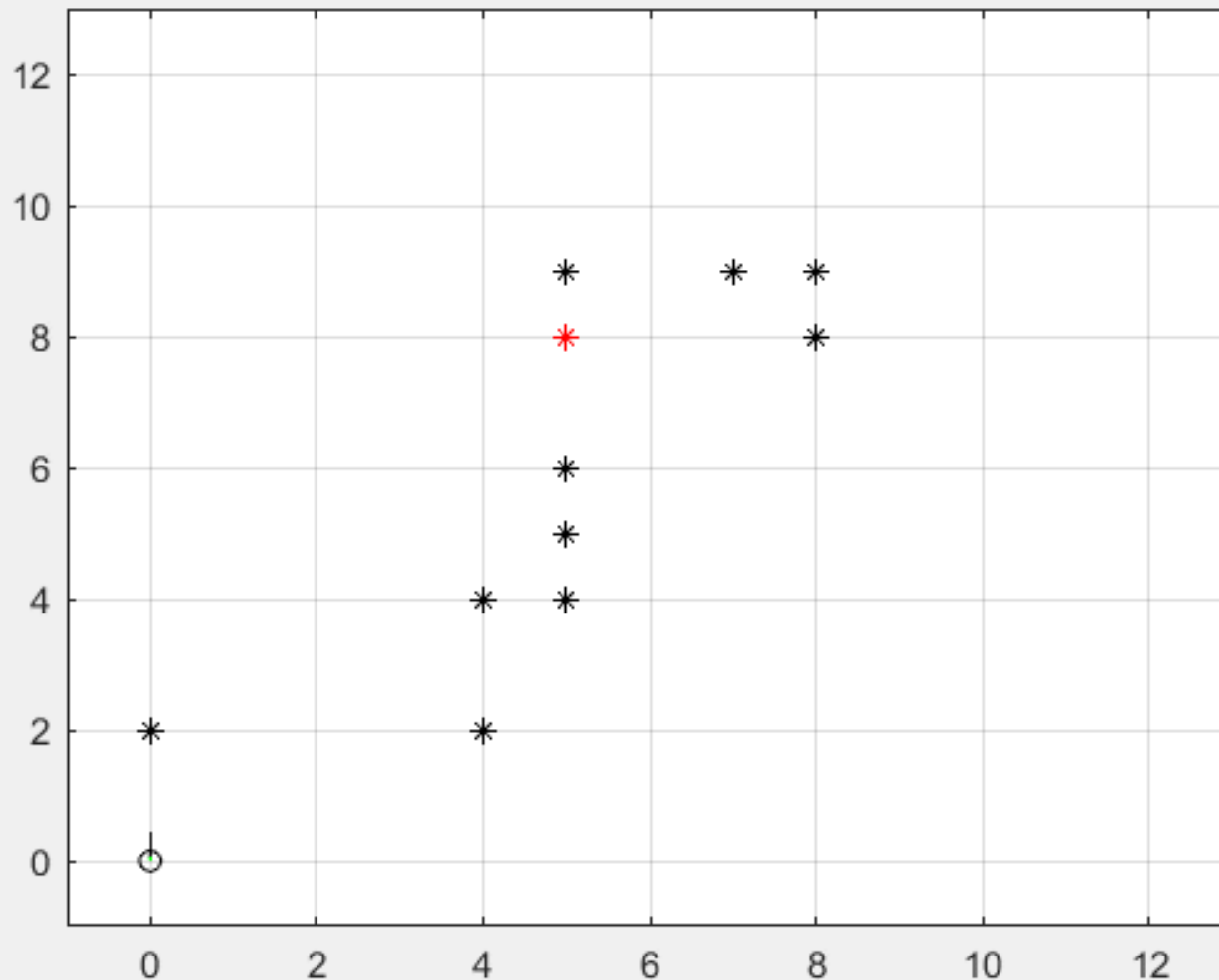
Autor: Atsushi Sakai

Link:

<https://github.com/AtsushiSakai/MATLABRobotics/blob/master/PathPlanning/DynamicWindowApproach/DynamicWindowApproachSample.m>



DWA – Przykład 2



Wygenerowano na bazie:

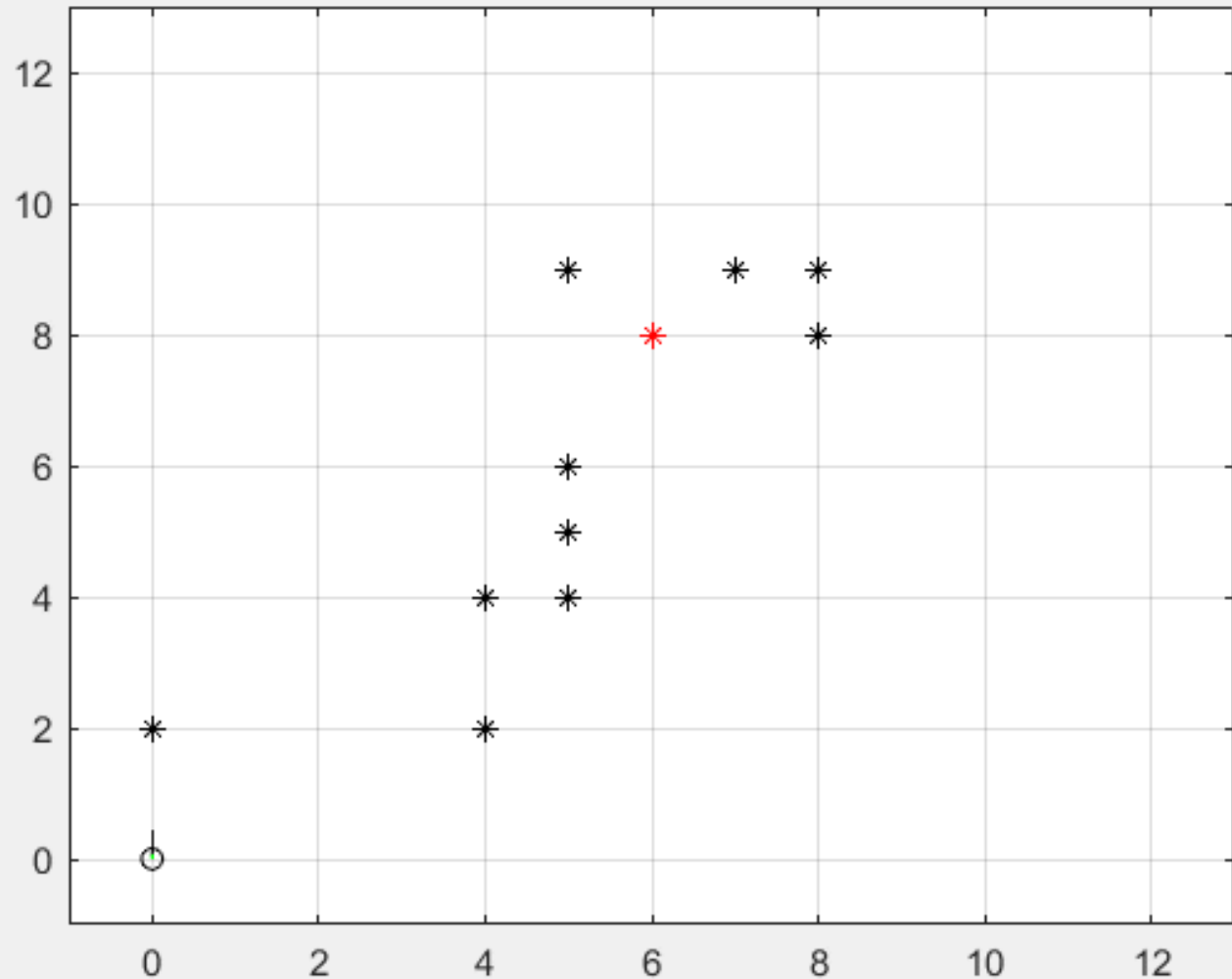
Autor: Atsushi Sakai

Link:

<https://github.com/AtsushiSakai/MATLABRobotics/blob/master/PathPlanning/DynamicWindowApproach/DynamicWindowApproachSample.m>



DWA – Przykład 3



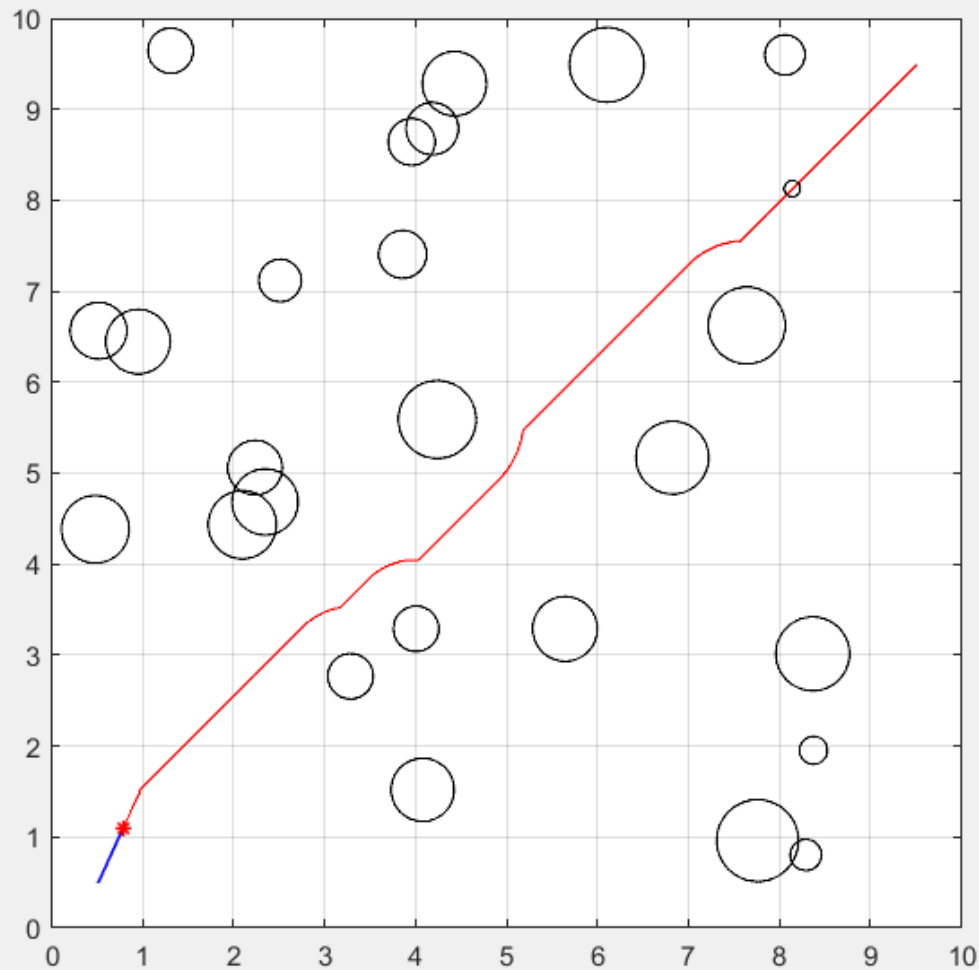
Wygenerowano na bazie:

Autor: Atsushi Sakai

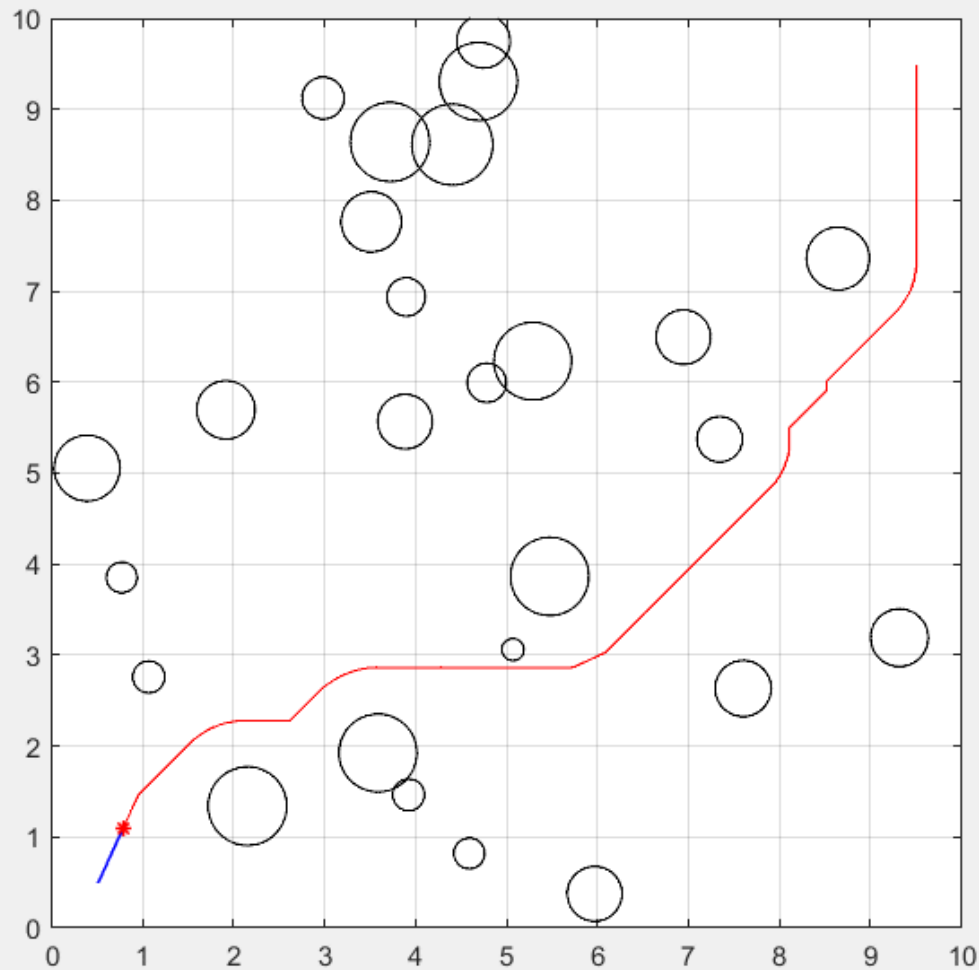
Link:

<https://github.com/AtsushiSakai/MATLABRobotics/blob/master/PathPlanning/DynamicWindowApproach/DynamicWindowApproachSample.m>

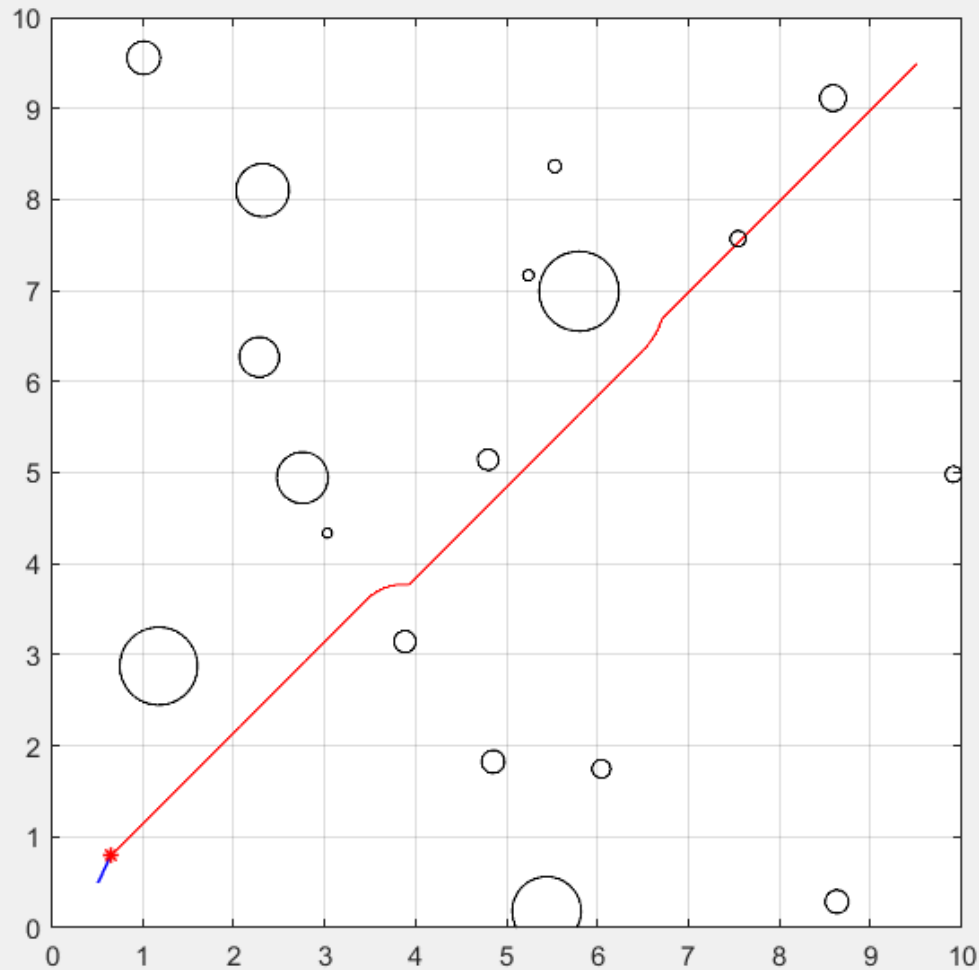
Śledzenie ścieżki globalnej (ang. *Waypoint tracking*)



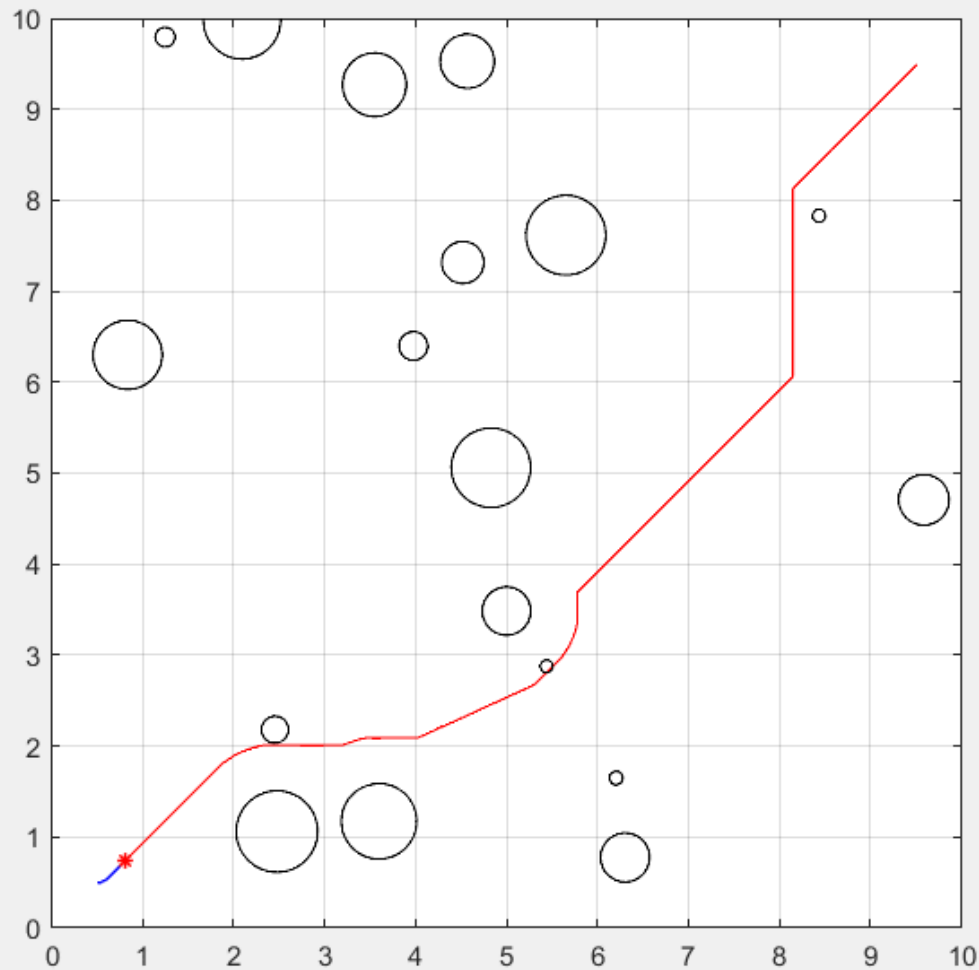
Śledzenie ścieżki globalnej (ang. *Waypoint tracking*)



Śledzenie ścieżki globalnej (ang. *Waypoint tracking*)



Śledzenie ścieżki globalnej (ang. *Waypoint tracking*)



Tesla AI Day 2021 (od 1:22:51)



Podsumowanie

Poszczególne elementy autonomicznej pracy pojazdu:

- Lokalizacja - odometria, czujniki bezwładnościowe, analiza obrazu, błędy i niepewności, Rozszerzony Filtr Kalmana, Filtr cząsteczkowy
- Mapowanie – metryczne, siatki zajętości, hybrydowe, topologiczne, budowa mapy siatki zajętości
- Globalne planowanie ścieżki – algorytm Dijkstra, Algorytm eksplorowania drzew losowych, algorytm mrówkowy
- Śledzenie ścieżki globalnej
- Lokalne planowanie ścieżki – Algorytm sztucznych pól potencjałowych, Algorytm dynamicznego okna



Podsumowanie

