UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

# Programowanie Robotów Mobilnych

## Planowanie ścieżki lokalnej

Rafał Szczepański
e-mail: szczepi (at) umk.pl
www.umk.pl/~szczepi

16.05.2023

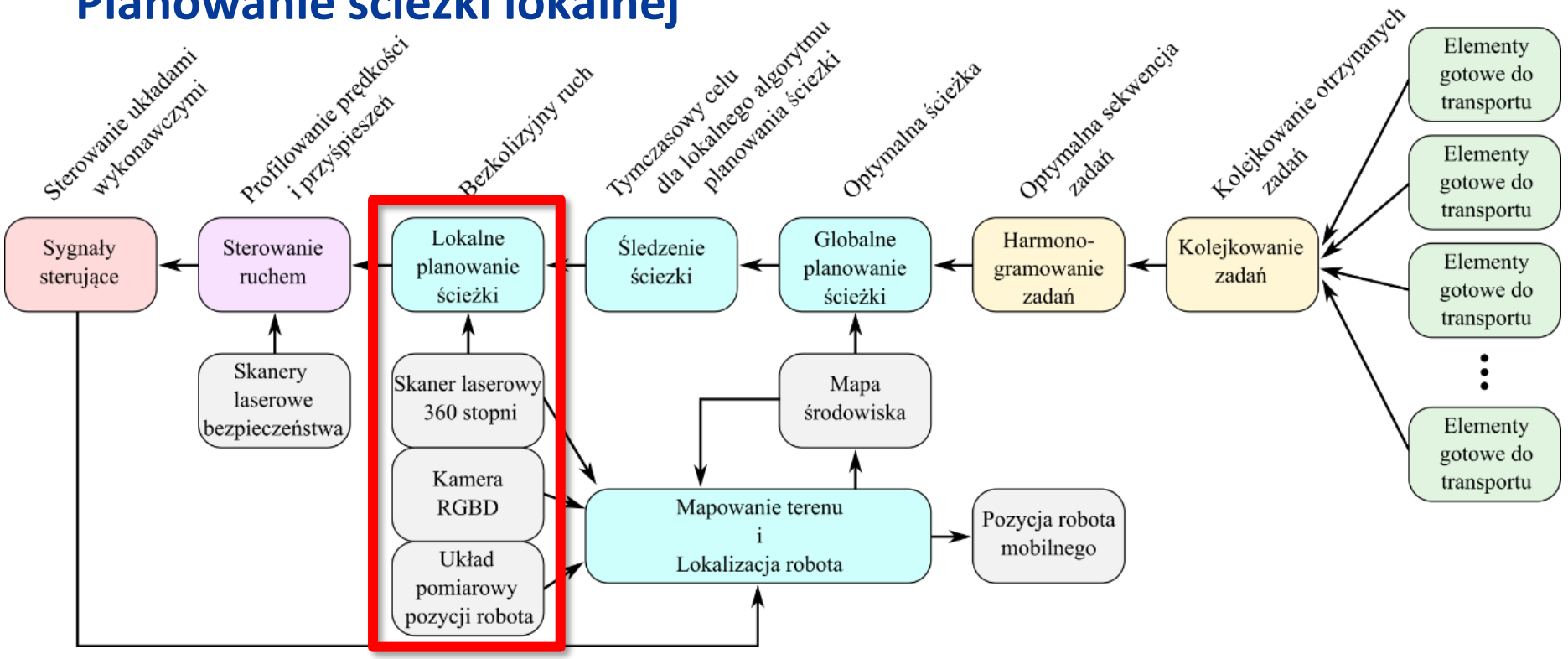
UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

## Plan prezentacji

- BUG0
- BUG1
- BUG2
- Algorytm sztucznych pól potencjałowych
- Algorytm dynamicznego okna

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

# Planowanie ścieżki lokalnej




**Założenia**:

- Znajomość pozycji robota
- Znajomość pozycji zadanej (cel)
- Brak znajomości mapy środowiska
- Lokalna znajomość środowiska poprzez wbudowane czujniki

**Oczekiwania**:

- Bezkolizyjne poruszanie się
- Osiągnięcie pozycji zadanej

UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

# Algorytmy BUG

Inspirowane przyrodą (insektami). W działaniu algorytmów BUG można wyróżnić dwie akcje:

- Podążaj na wprost celu,
- Podążaj wzdłuż ściany/przeszkody (lewostronnie lub prawostronnie).

Algorytmy będą działały nawet w przypadku zastosowania czujników uderzeniowych. Nie wymagają one czujników odległości ani skanera laserowego. Natomiast wymagają znajomości swojej pozycji (np. metodą odometryczną).

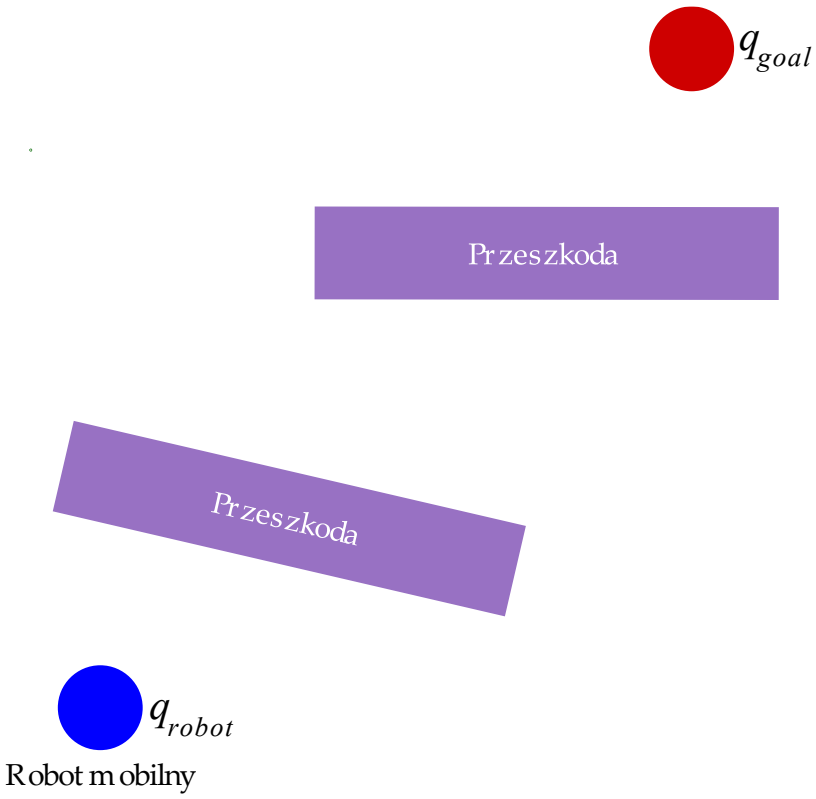
Oznaczmy wektorem $q = [x, y]^T$ pozycję w przestrzeni:

- $q_{robot}$ – pozycja robota,
- $q_{goal}$ – pozycja docelowa,
- $q^H$ – punkt zdarzenia wykrycia przeszkody (od ang. *hit*),
- $q^L$ – punkt opuszczenia przeszkody (od ang. *leave*).

UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

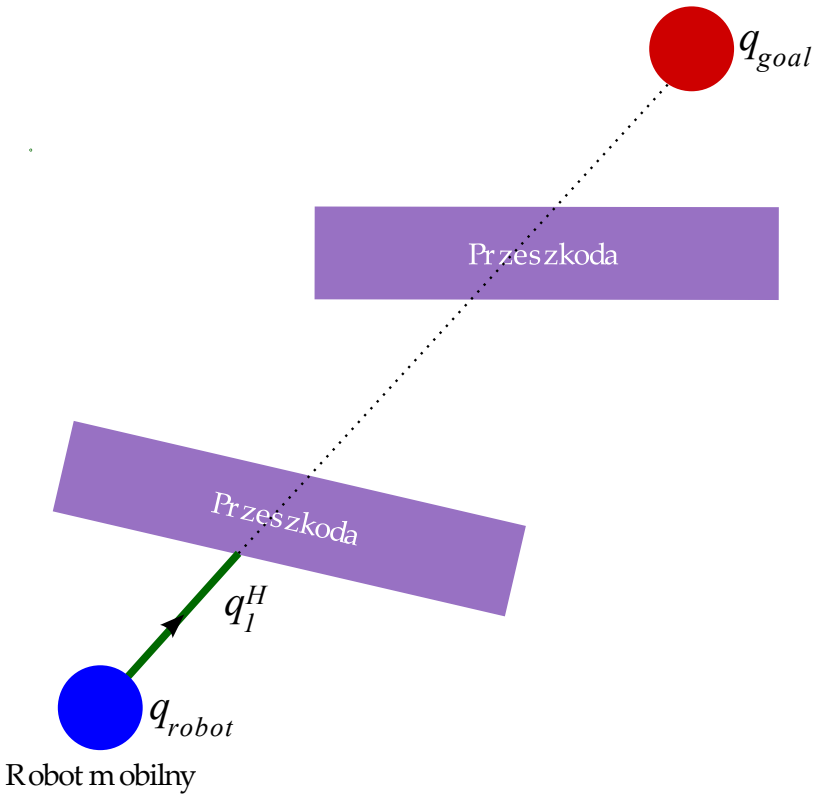
# BUG0 - Strategia

1. Podążaj na wprost celu
2. Jeżeli napotkałeś przeszkodę to podążaj wzdłuż niej aż do odzyskania możliwości poruszania na wprost celu
3. Kontynuuj



UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

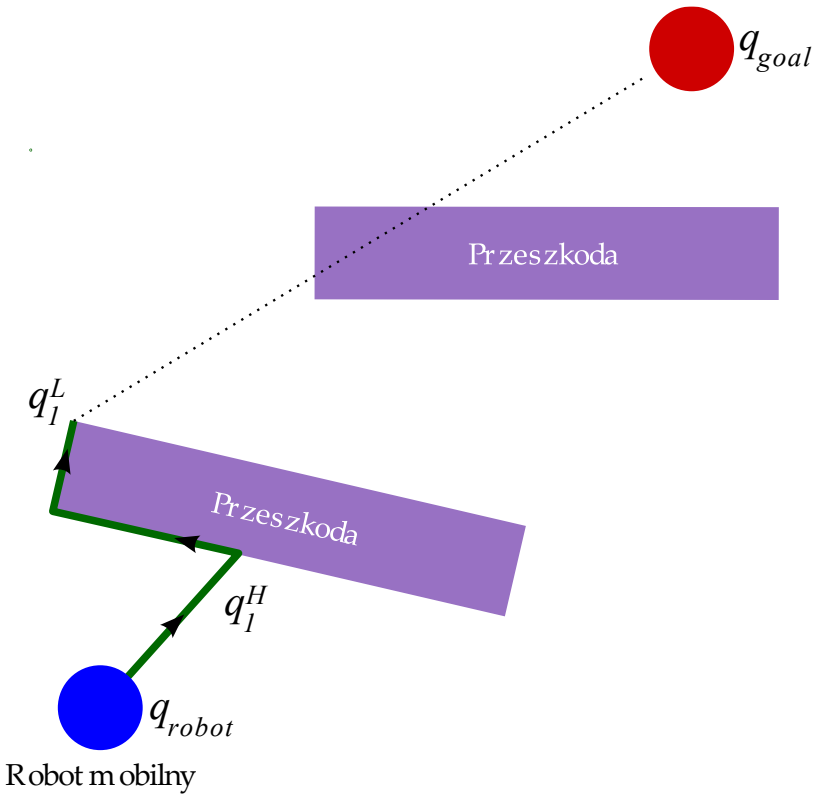
# BUG0 - Strategia




1. **Podążaj na wprost celu**
2. Jeżeli napotkałeś przeszkodę to podążaj wzdłuż niej aż do odzyskania możliwości poruszania na wprost celu
3. Kontynuuj

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

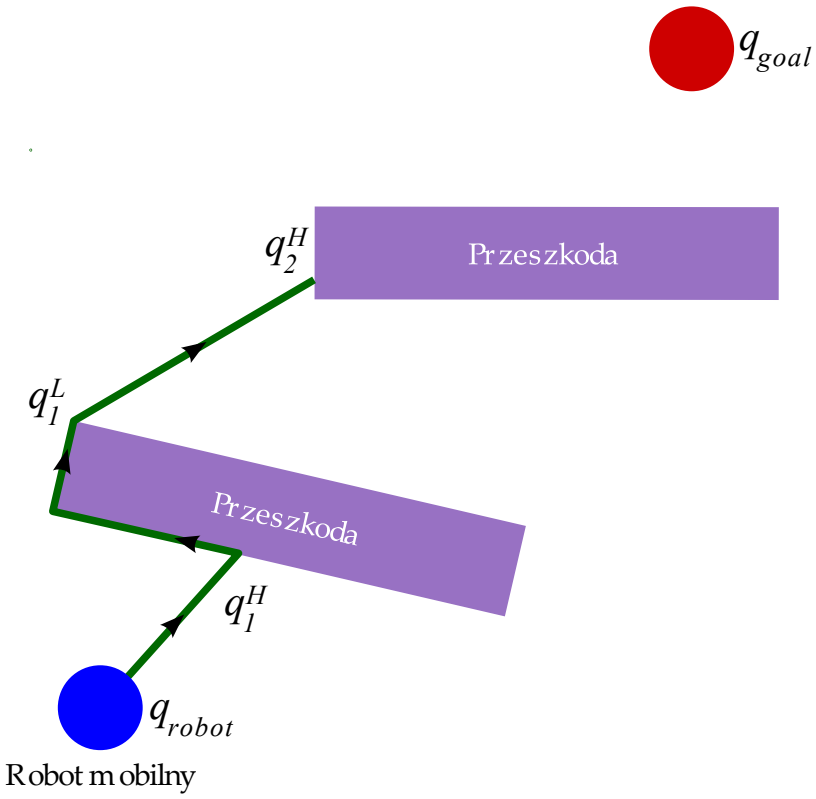
# BUG0 - Strategia




1. Podążaj na wprost celu
2. **Jeżeli napotkałeś przeszkodę to podążaj wzdłuż niej aż do odzyskania możliwości poruszania na wprost celu**
3. Kontynuuj

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

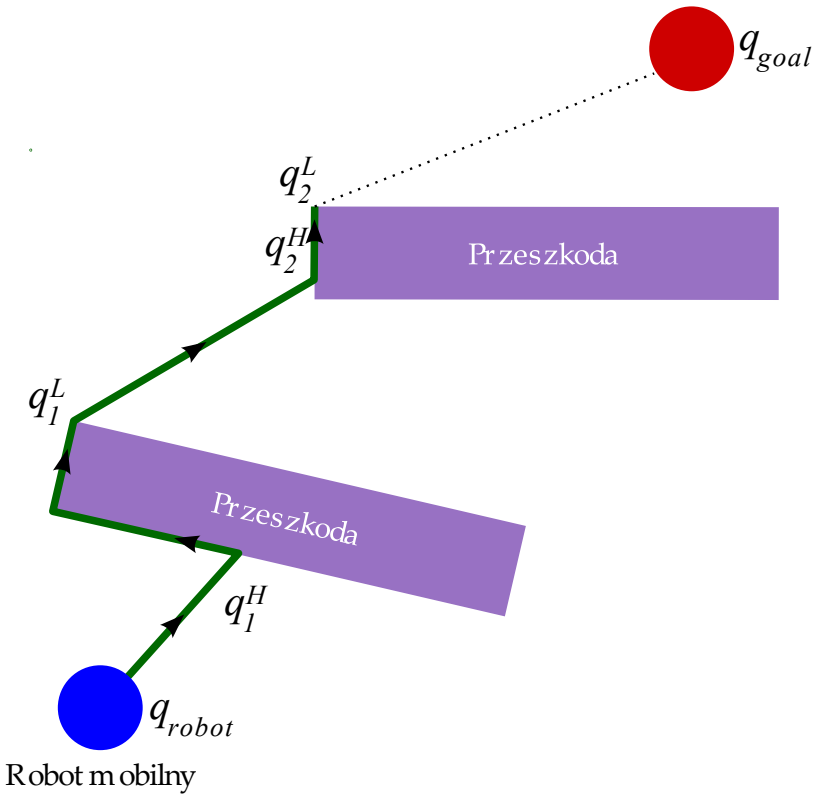
# BUG0 - Strategia




1. **Podążaj na wprost celu**
2. Jeżeli napotkałeś przeszkodę to podążaj wzdłuż niej aż do odzyskania możliwości poruszania na wprost celu
3. **Kontynuuj**

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

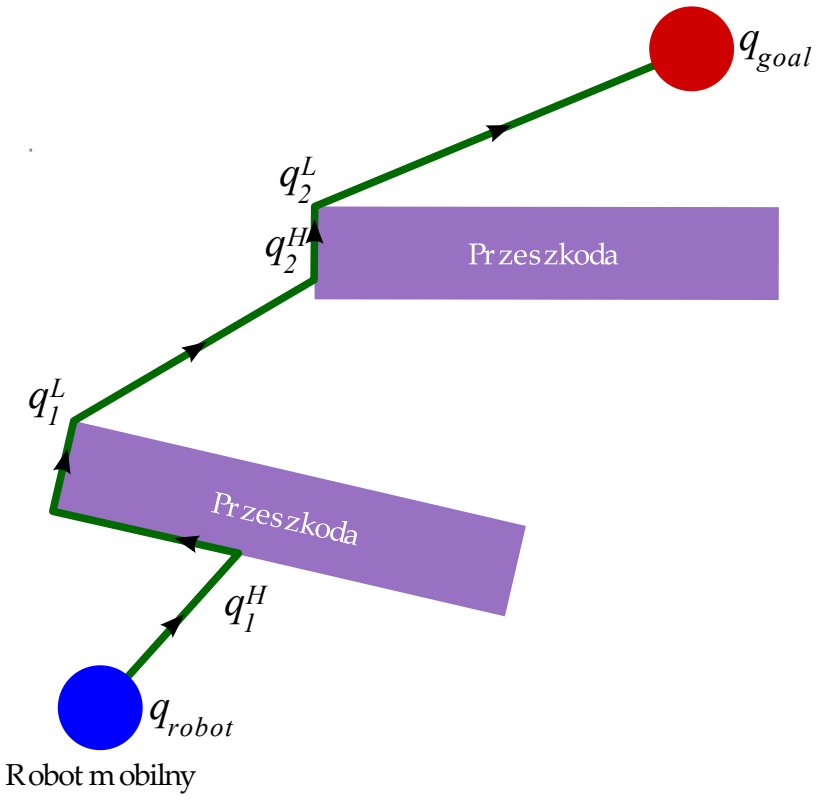
# BUG0 - Strategia




1. Podążaj na wprost celu
2. **Jeżeli napotkałeś przeszkodę to podążaj wzdłuż niej aż do odzyskania możliwości poruszania na wprost celu**
3. Kontynuuj

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

# BUG0 - Strategia




1. **Podążaj na wprost celu**
2. Jeżeli napotkałeś przeszkodę to podążaj wzdłuż niej aż do odzyskania możliwości poruszania na wprost celu
3. **Kontynuuj**

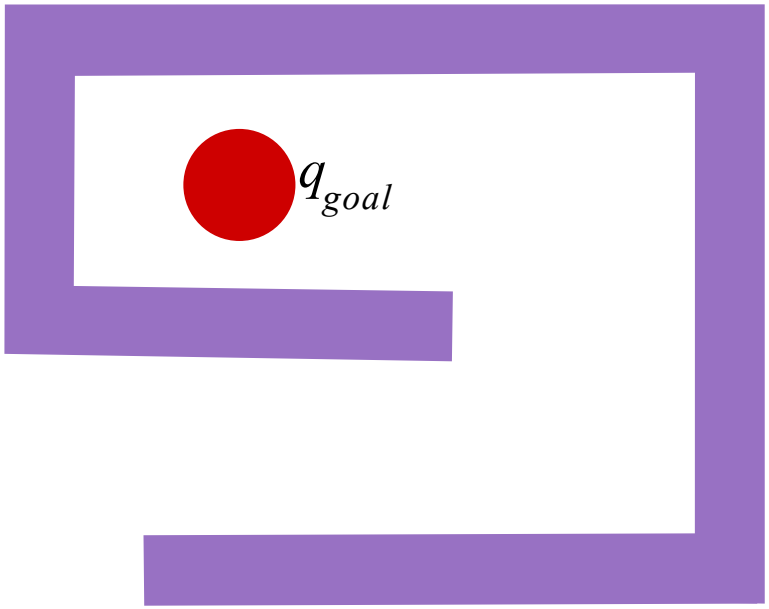
UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

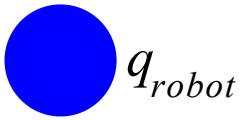
# BUG0 - pseudokod

| | **Pseudokod algorytmu BUG0** |
|---|---|
| | Wejście: $q_{start}$, $q_{goal}$<br>Wyjście: informacja o osiągnięciu celu lub informacja, że ścieżka do celu nie istnieje |
| 1 | **Dopóki** 1 == 1 wykonuj |
| 2 | **Powtórz** |
| 3 | Przesuń się wprost do celu |
| 4 | **Dopóki** $q_{goal}$ nie został osiągnięty lub przeszkoda nie została wykryta |
| 5 | **Jeżeli** $q_{goal}$ został osiągnięty wtedy |
| 6 | Zwróć flagę powodzenia |
| 7 | **Koniec jeżeli** |
| 8 | $q_i^H = q_{robot}$ |
| 9 | **Powtórz** |
| 10 | Podążaj wzdłuż przeszkody |
| 11 | **Dopóki** $q_{goal}$ nie został osiągnięty **lub** możliwy jest ruch na wprost przeszkody |
| 12 | **lub** $q_i^H$ został ponownie osiągnięty |
| 13 | $q_i^L = q_{robot}$ |
| 14 | **Jeżeli** robot nie może poruszyć się na wprost celu wtedy |
| 15 | Zwróć flagę porażki |
| 16 | **Koniec jeżeli** |
| 17 | i += 1 |
| 18 | **Koniec dopóki** |

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

# BUG0 – Zawsze zadziała?

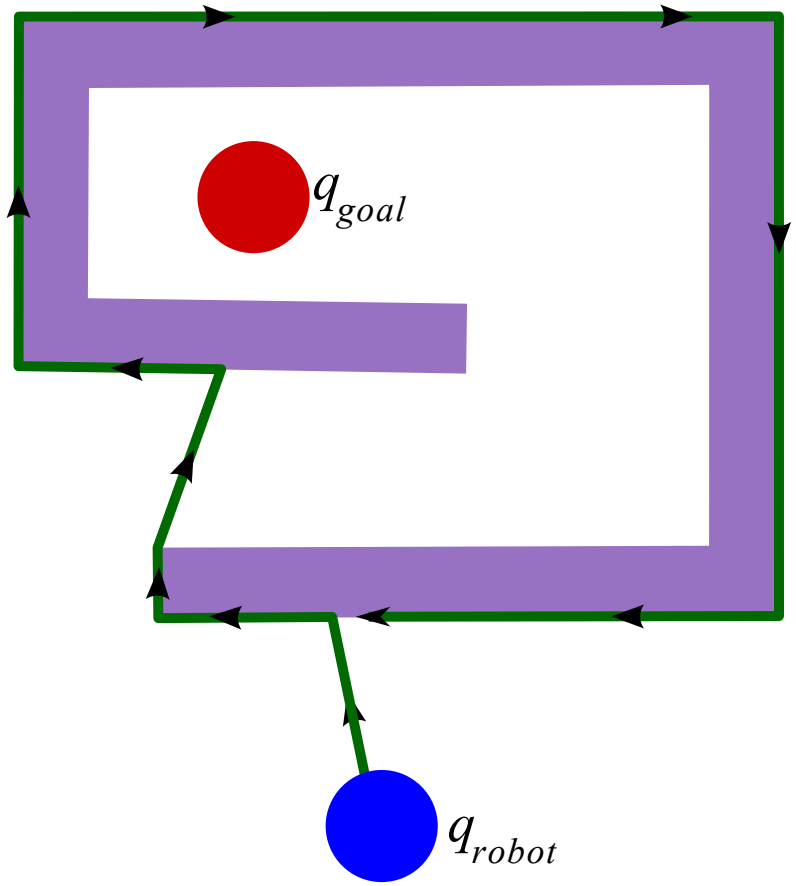



Robot mobilny

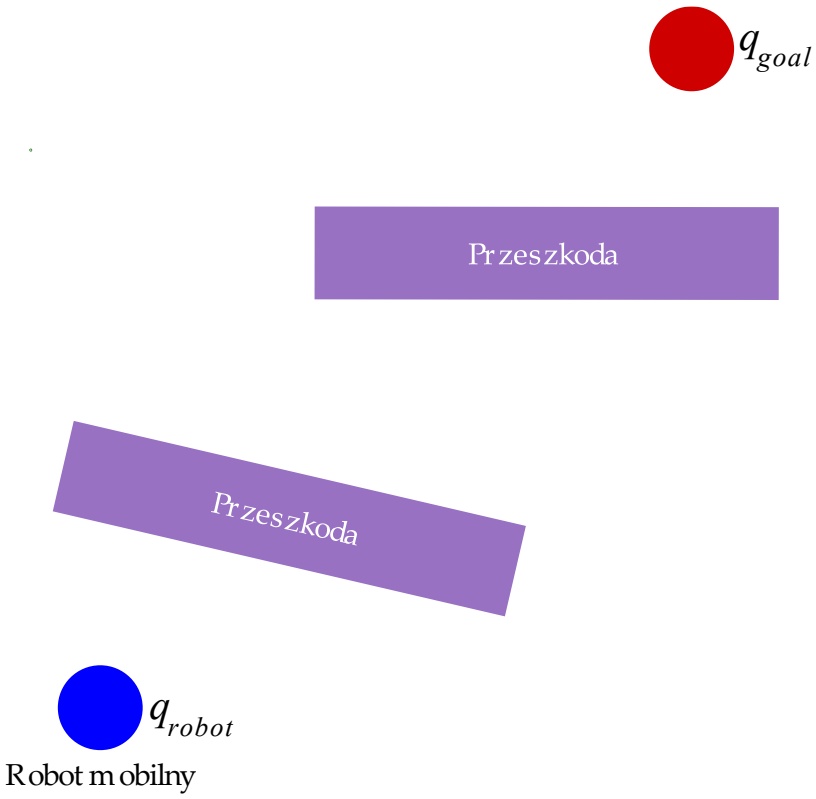
UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

# BUG0 – Zawsze zadziała?



UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

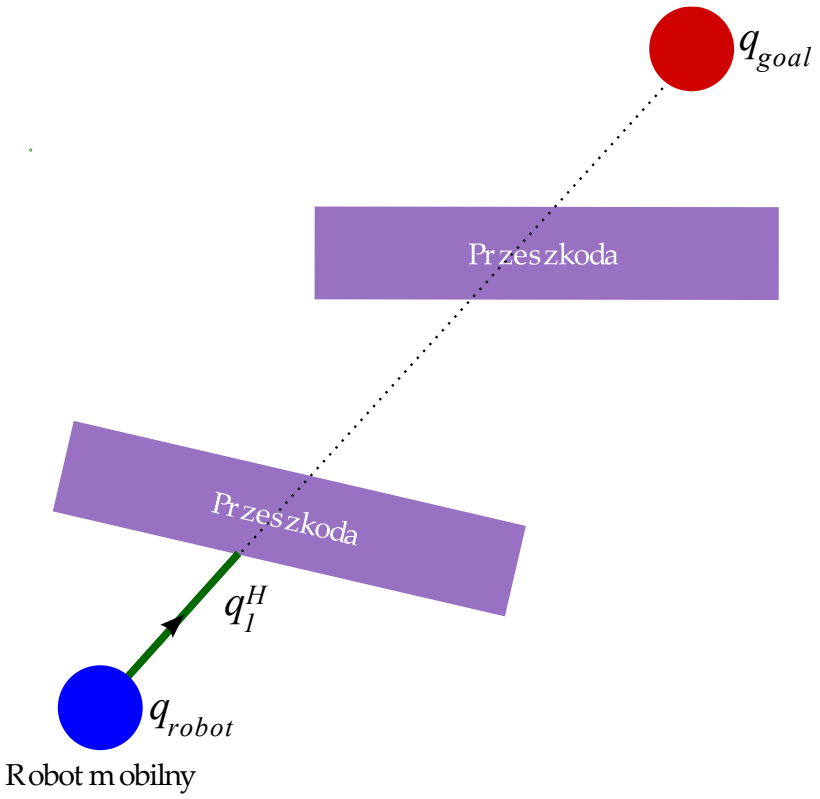
# BUG1 - Strategia




1. Podążaj na wprost celu
2. Jeżeli napotkałeś przeszkodę to podążaj wzdłuż niej **aż ją okrążysz zapamiętując pozycję, która jest najbliżej celu**
3. **Wróć do punktu, który był najbliżej celu i** kontynuuj

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

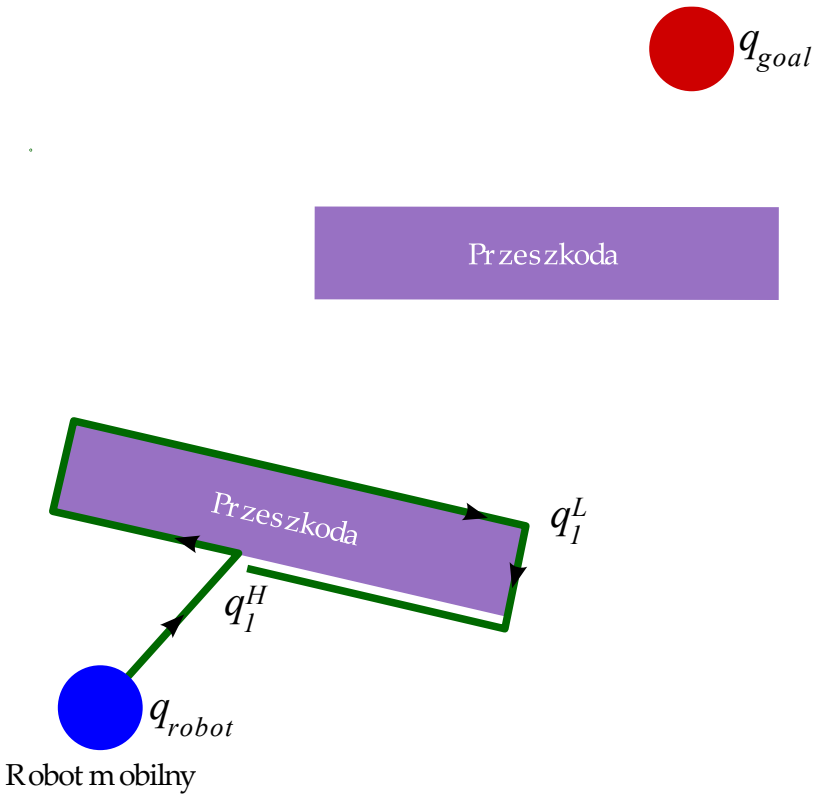
# BUG1 - Strategia




1. **Podążaj na wprost celu**
2. Jeżeli napotkałeś przeszkodę to podążaj wzdłuż niej aż ją okrążysz zapamiętując pozycję, która jest najbliżej celu
3. Wróć do punktu, który był najbliżej celu i kontynuuj

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

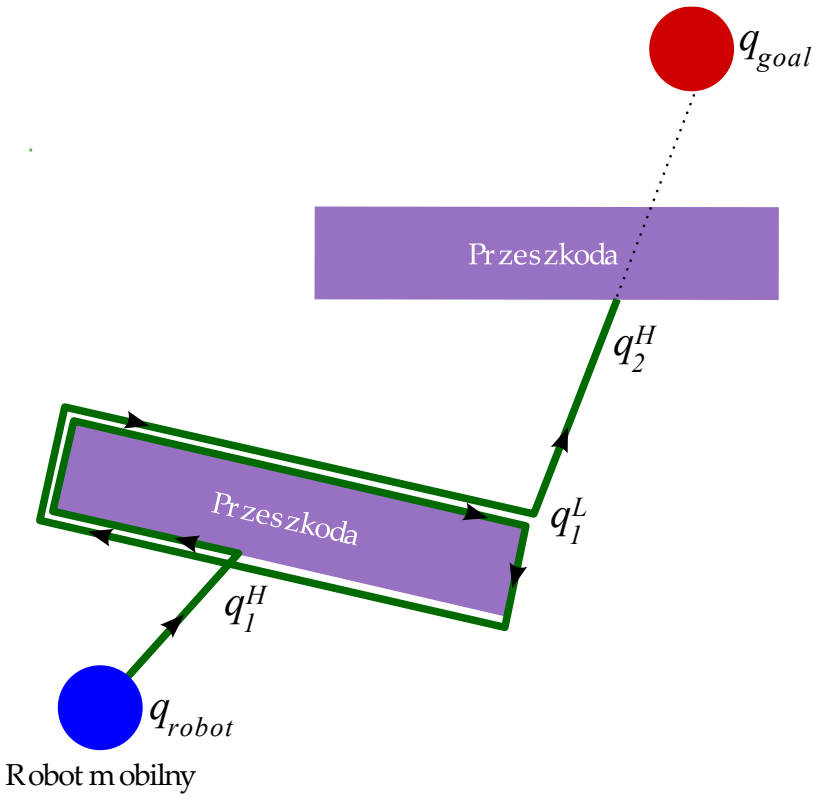
# BUG1 - Strategia




1. Podążaj na wprost celu
2. **Jeżeli napotkałeś przeszkodę to podążaj wzdłuż niej aż ją okrążysz zapamiętując pozycję, która jest najbliżej celu**
3. Wróć do punktu, który był najbliżej celu i kontynuuj

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

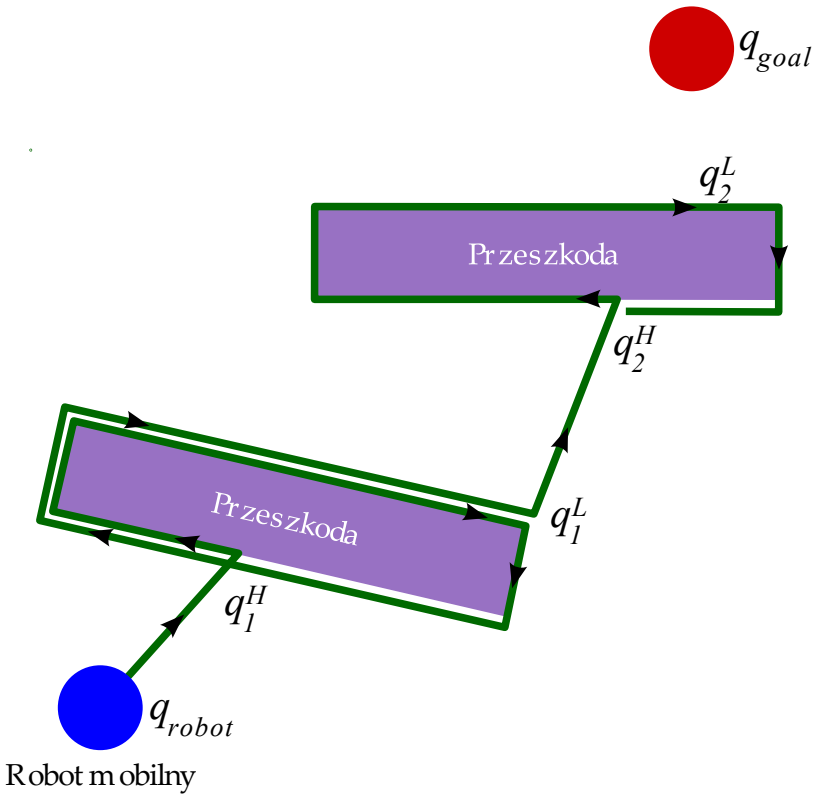
# BUG1 - Strategia




1. **Podążaj na wprost celu**
2. Jeżeli napotkałeś przeszkodę to podążaj wzdłuż niej aż ją okrążysz zapamiętując pozycję, która jest najbliżej celu
3. **Wróć do punktu, który był najbliżej celu i kontynuuj**

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

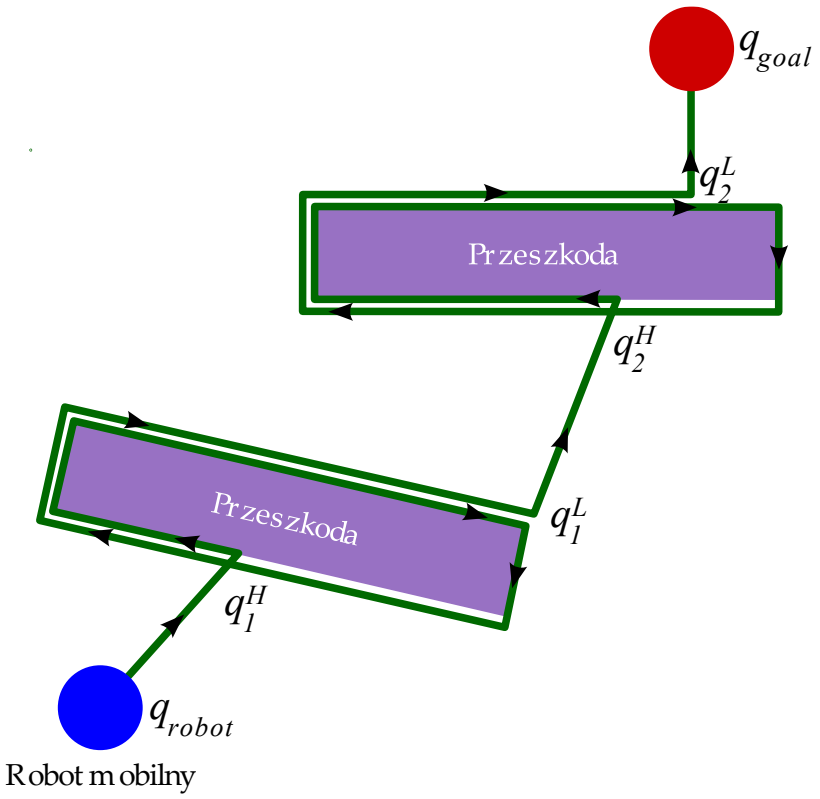
# BUG1 - Strategia




1. Podążaj na wprost celu
2. **Jeżeli napotkałeś przeszkodę to podążaj wzdłuż niej aż ją okrążysz zapamiętując pozycję, która jest najbliżej celu**
3. Wróć do punktu, który był najbliżej celu i kontynuuj

UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

# BUG1 - Strategia




1. **Podążaj na wprost celu**
2. Jeżeli napotkałeś przeszkodę to podążaj wzdłuż niej aż ją okrążysz zapamiętując pozycję, która jest najbliżej celu
3. **Wróć do punktu, który był najbliżej celu i kontynuuj**

UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

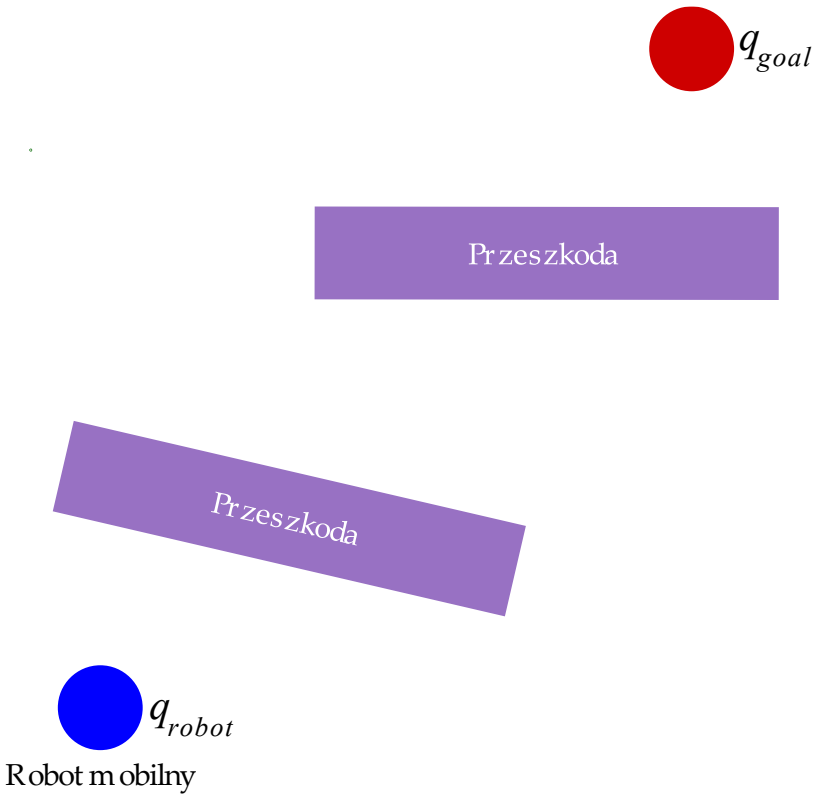
# BUG1 - pseudokod

| | **Pseudokod algorytmu BUG1** |
|---|---|
| | Wejście: $q_{start}$, $q_{goal}$<br><br>Wyjście: informacja o osiągnięciu celu lub informacja, że ścieżka do celu nie istnieje |
| 1 | **Dopóki** 1 == 1 wykonuj |
| 2 | **Powtórz** |
| 3 | Przesuń się wprost do celu |
| 4 | **Dopóki** $q_{goal}$ nie został osiągnięty lub przeszkoda nie została wykryta |
| 5 | **Jeżeli** $q_{goal}$ został osiągnięty wtedy |
| 6 | Zwróć flagę powodzenia |
| 7 | **Koniec jeżeli** |
| 8 | $q_i^H = q_{robot}$ |
| 9 | **Powtórz** |
| 10 | Podążaj wzdłuż przeszkody |
| 11 | **Dopóki** $q_{goal}$ nie został osiągnięty **lub** $q_i^H$ nie został ponownie osiągnięty |
| 12 | Wyznacz punkt $q_i^L$ jako punkt najbliższy do celu od przeszkody |
| 13 | Poruszaj się wzdłuż przeszkody do punktu $q_i^L$ |
| 14 | **Jeżeli** robot nie może poruszyć się na wprost celu wtedy |
| 15 | Zwróć flagę porażki |
| 16 | **Koniec jeżeli** |
| 17 | i += 1 |
| 18 | **Koniec dopóki** |

UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

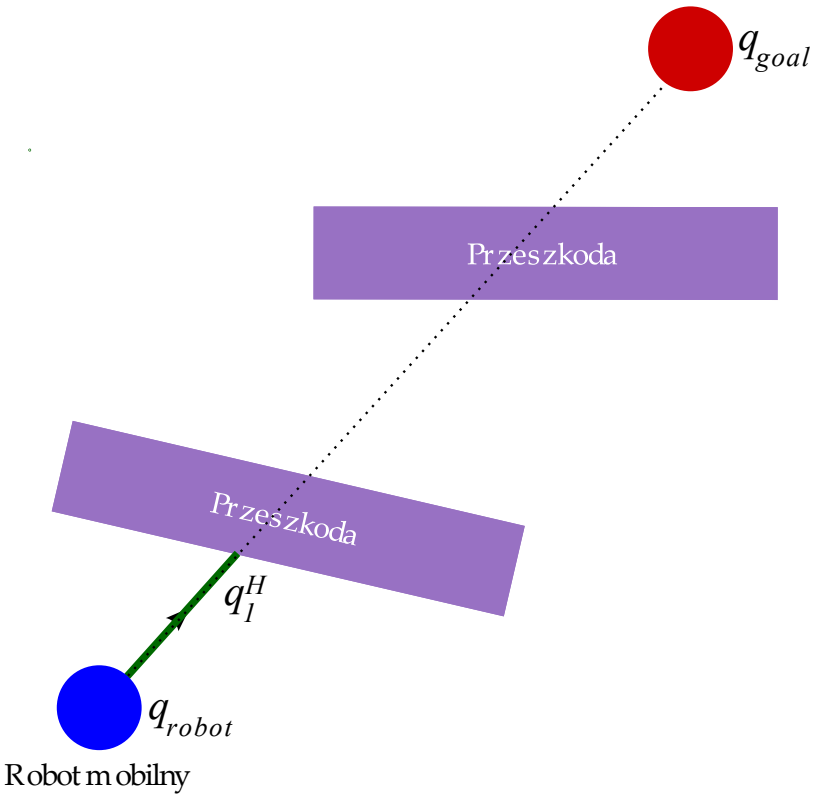
# BUG2 - Strategia

1. Podążaj na wprost celu
2. Jeżeli napotkałeś przeszkodę to podążaj wzdłuż niej **aż przetniesz linię kierunkową**
3. Kontynuuj



UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

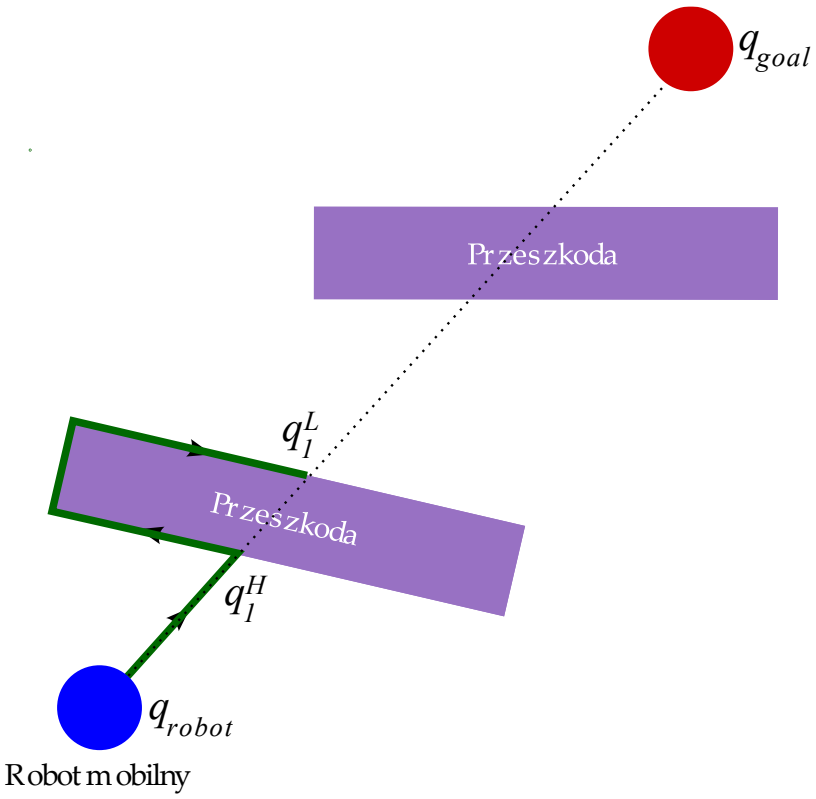
# BUG2 - Strategia




1. **Podążaj na wprost celu**
2. Jeżeli napotkałeś przeszkodę to podążaj wzdłuż niej aż przetniesz linię kierunkową
3. Kontynuuj

UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

# BUG2 - Strategia




1. Podążaj na wprost celu
2. **Jeżeli napotkałeś przeszkodę to podążaj wzdłuż niej aż przetniesz linię kierunkową**
3. Kontynuuj

Dopasowujemy parametry $a$ i $b$ równiania funkcji liniowej $y = ax + b$. Przecięcie funkcji możemy zdefiniować chwilą w której zmienia się znak w kolejnej chwili czasu równania:

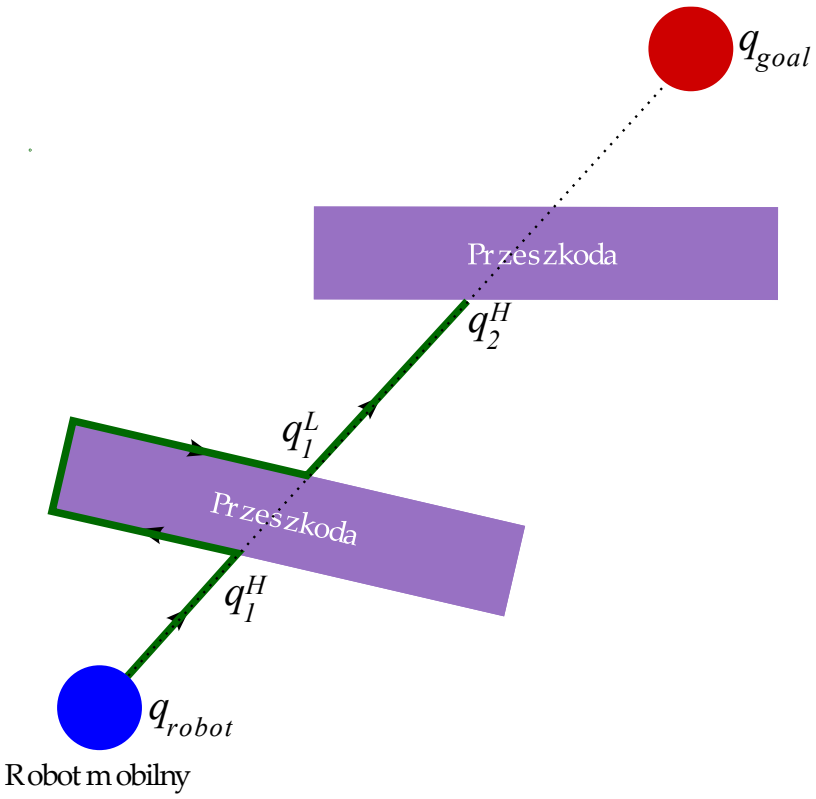
$$y(q^x_{robot}) - q^y_{robot}$$

Więc warunek przecięcia linii kierunkowej można zapisać następująco:

$$\left(aq^x_{robot}(k-1) + b - q^y_{robot}(k-1)\right) \cdot \left(aq^x_{robot}(k) + b - q^y_{robot}(k)\right) \leq 0$$

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

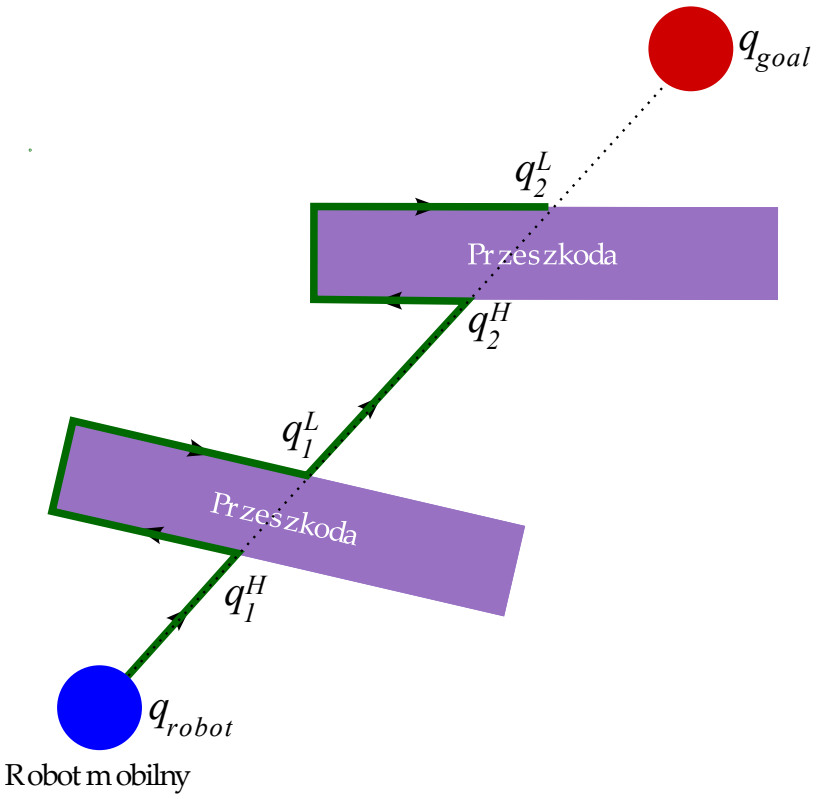
# BUG2 - Strategia




1. **Podążaj na wprost celu**
2. Jeżeli napotkałeś przeszkodę to podążaj wzdłuż niej aż przetniesz linię kierunkową
3. **Kontynuuj**

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

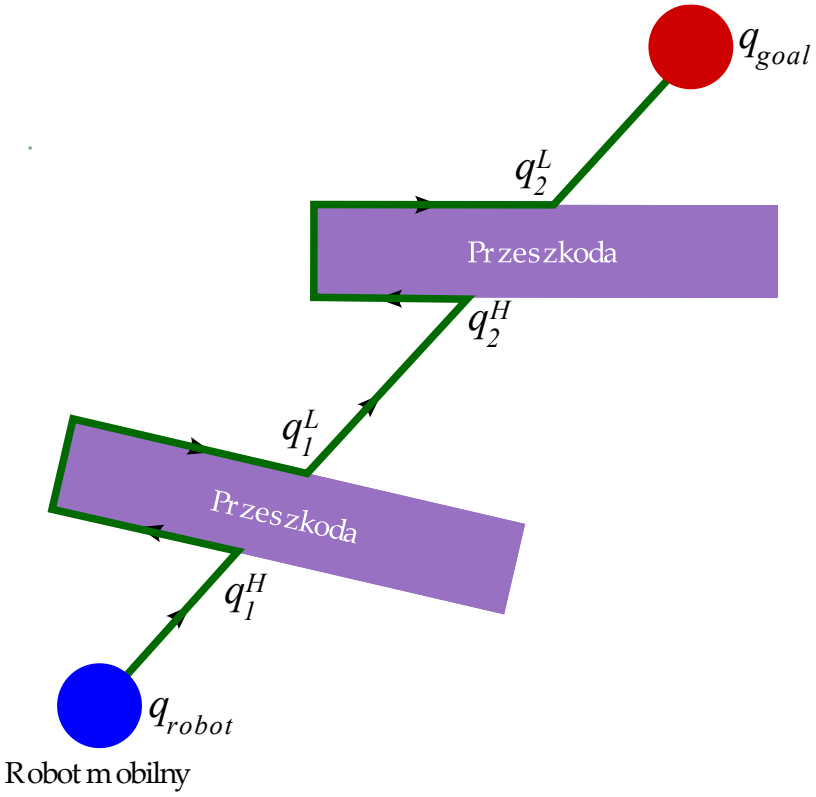
# BUG2 - Strategia




1. Podążaj na wprost celu
2. **Jeżeli napotkałeś przeszkodę to podążaj wzdłuż niej aż przetniesz linię kierunkową**
3. Kontynuuj

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

# BUG2 - Strategia




1. **Podążaj na wprost celu**
2. Jeżeli napotkałeś przeszkodę to podążaj wzdłuż niej aż przetniesz linię kierunkową
3. **Kontynuuj**

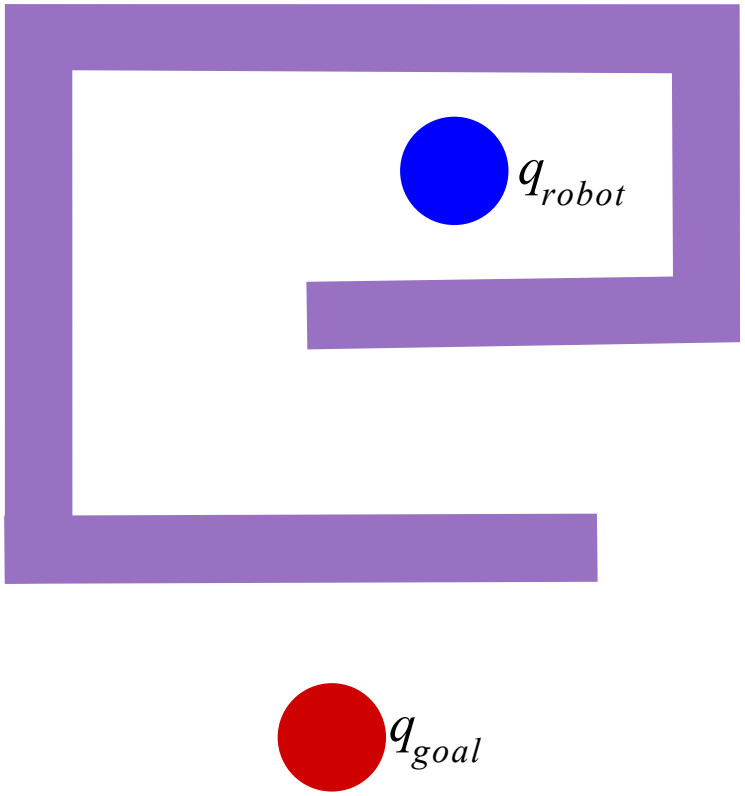
UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

# BUG2 - pseudokod

| | **Pseudokod algorytmu BUG2** |
|---|---|
| | Wejście: $q_{start}$, $q_{goal}$<br><br>Wyjście: informacja o osiągnięciu celu lub informacja, że ścieżka do celu nie istnieje |
| 1 | **Dopóki** 1 == 1 wykonuj |
| 2 | **Powtórz** |
| 3 | Przesuń się wprost do celu |
| 4 | **Dopóki** $q_{goal}$ nie został osiągnięty lub przeszkoda nie została wykryta |
| 5 | **Jeżeli** $q_{goal}$ został osiągnięty wtedy |
| 6 | Zwróć flagę powodzenia |
| 7 | **Koniec jeżeli** |
| 8 | $q_i^H = q_{robot}$ |
| 9 | Skręć w lewo (lub w prawo) |
| 10 | **Powtórz** |
| 11 | Podążaj wzdłuż przeszkody |
| 12 | **Dopóki** $q_{goal}$ nie został osiągnięty **lub** $q_i^H$ nie został ponownie osiągnięty |
| 13 | **lub** prosta łącząca punkty $q_{start}$ $i$ $q_{goal}$ nie została osiągnięta |
| 14 | **Jeżeli** robot nie może poruszyć się na wprost celu wtedy |
| 15 | Zwróć flagę porażki |
| 16 | **Koniec jeżeli** |
| 17 | i += 1 |
| 18 | **Koniec dopóki** |

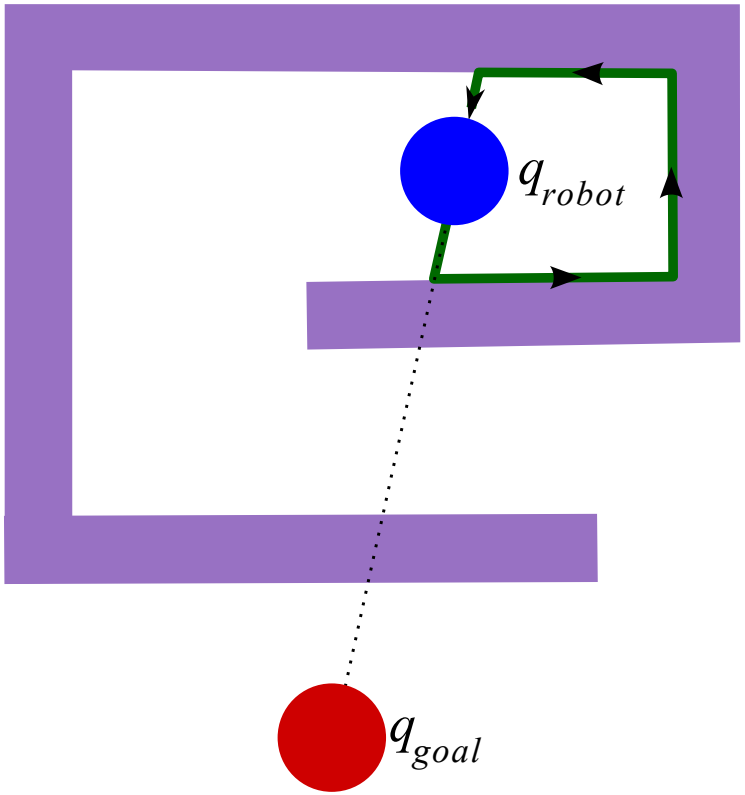
UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

# BUG2 – Zawsze zadziała?



UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

# BUG2 – Zawsze zadziała?



UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

# Algorytmy BUG0, BUG1, BUG2 - Porównanie

- Algorytm BUG1 przeszukuje wszystkie możliwości przed wykonaniem kolejnego etapu. Dzięki temu, jeżeli to możliwe, znajdzie ścieżkę do punktu docelowego.

- Algorytmy BUG0 i BUG2 przeszukują „zachłannie” i w razie wystąpienia odpowiedniego warunku zaczynają ponownie poruszać się na wprost punktu docelowego. Powoduje to możliwość utknięcia algorytmu.

- W ogólności algorytmy BUG0 i BUG2 dostarczają lepsze rozwiązanie niż BUG1 (krótsza ścieżka). Niestety, mogą powodować problemy z dodarciem do punktu docelowego.

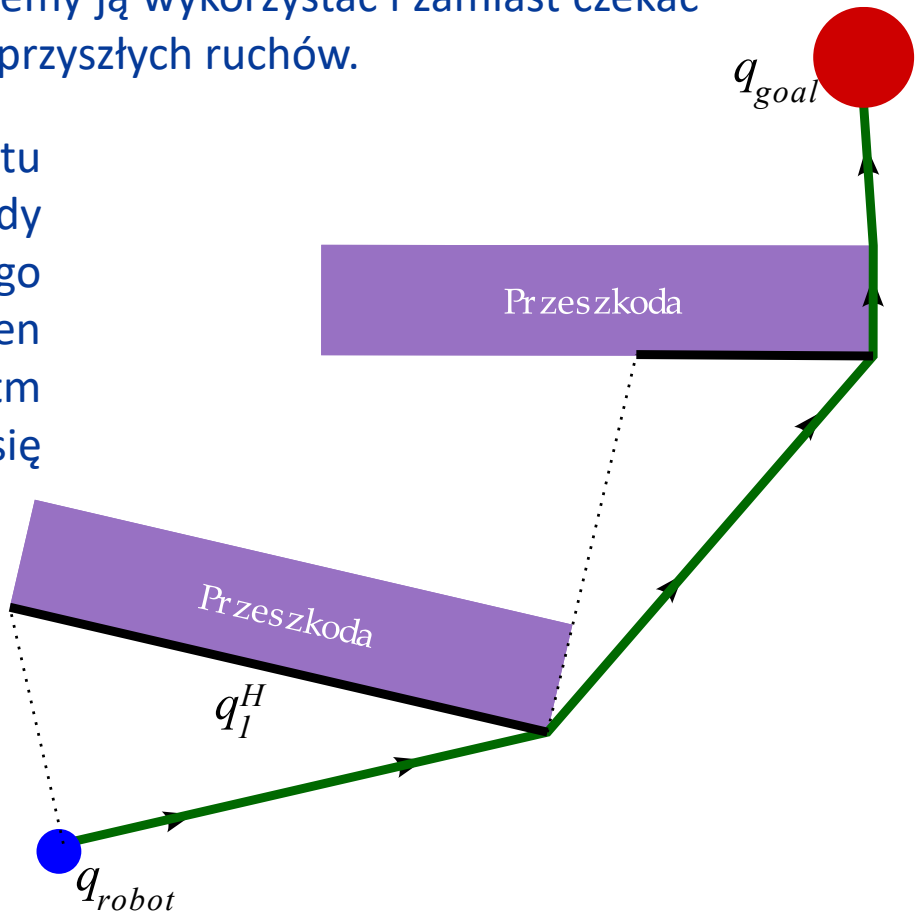
- Ruch lewostronny lub prawostronny może zostać zdefiniowany odgórnie w algorytmie lub może być wybierany losowo podczas wykrycia kolejnej przeszkody. Zastosowanie losowości powoduje, że algorytmy BUG0 i BUG2 znajdą rozwiązanie, lecz może być ono gorsze niż algorytmu BUG1.

UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

# Algorytm BUG0 - skaner laserowy 360 stopni

Założeniem algorytmów BUG jest podążanie wzdłuż przeszkód. Można to zrealizować czujnikami uderzeniowymi. Mając natomiast informację o przeszkodach do których się zbliżamy możemy ją wykorzystać i zamiast czekać na „zderzenie” możemy wykonać predykcję przyszłych ruchów.

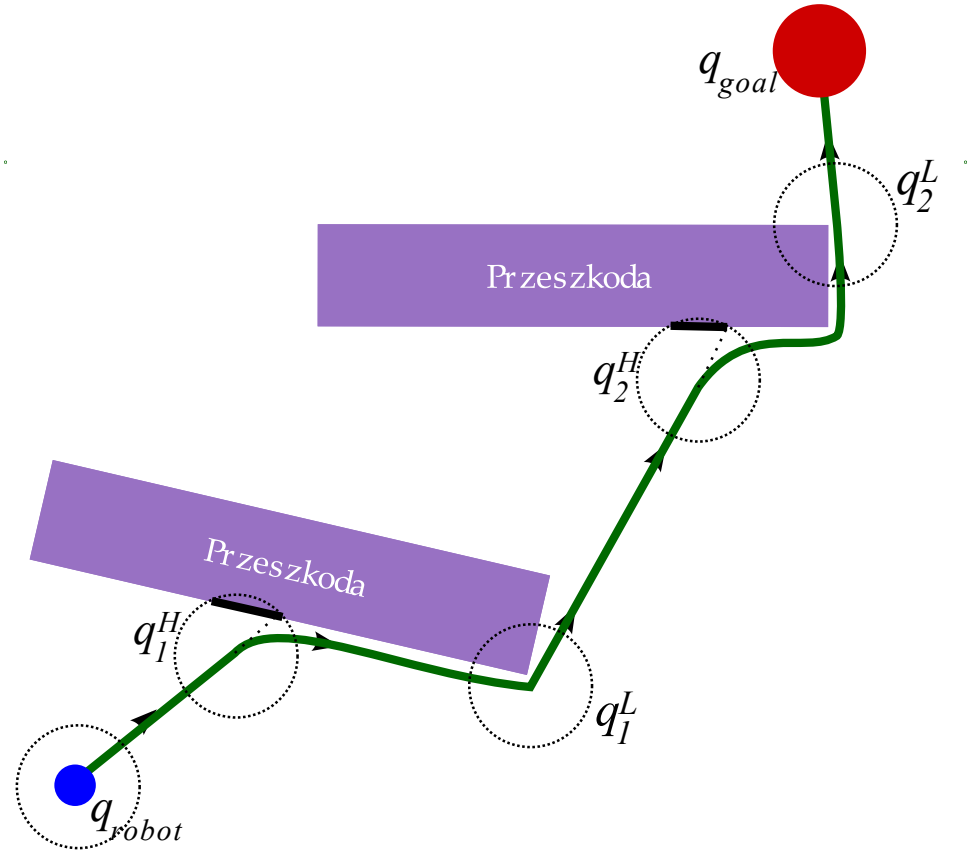
Predykcja polega na znalezieniu punktu „opuszczenia” śledzenia przeszkody w przypadku poruszania się lewostronnego i prawostronnego. Następnie wybierając ten z punktów, który jest bliżej (algorytm zachłanny) jako kolejny cel poruszania się możemy skrócić przebytą drogę robota.



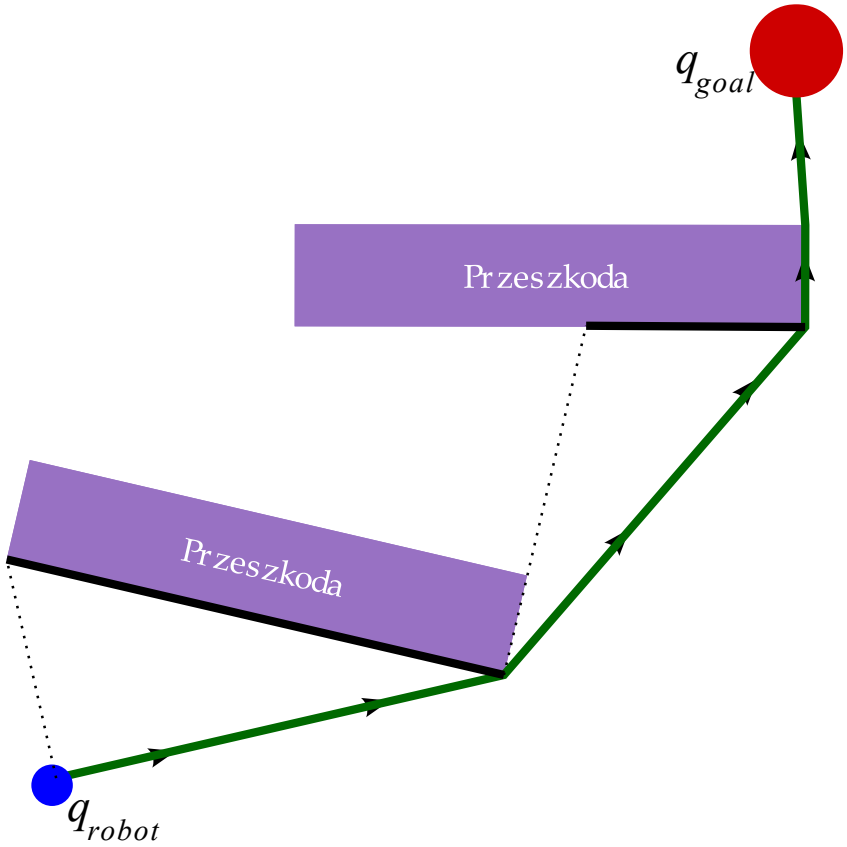
UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

# Algorytm BUG2 - zasięg skanera laserowego 360 stopni

## Ograniczony zasięg czujnika




## Nieograniczony zasięg czujnika

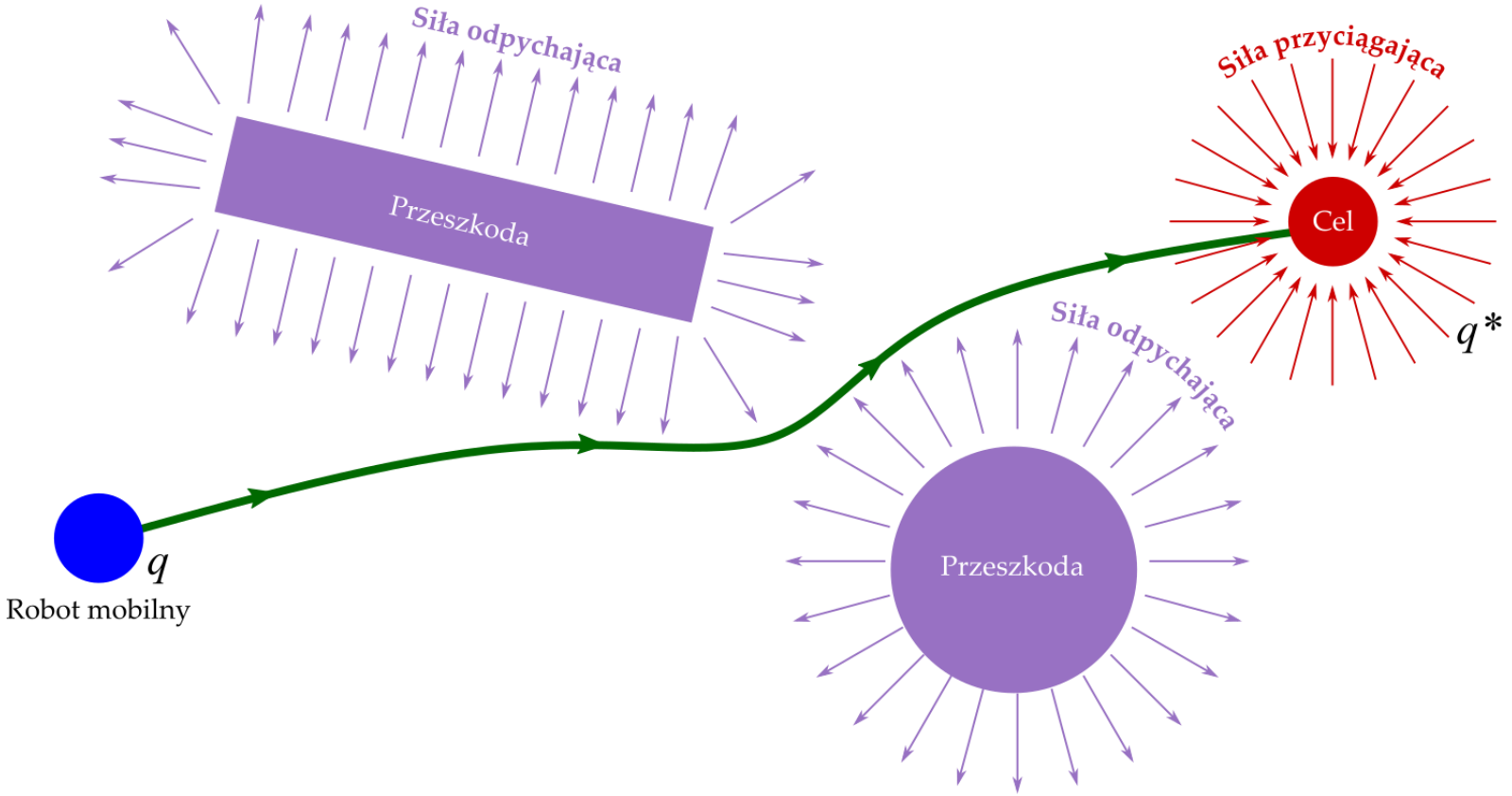


UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

# 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.



UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

# 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 intepretowana jako energia, a wtedy gradient potencjału jest siłą. Gradient funkcji potencjału jest następujący:

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

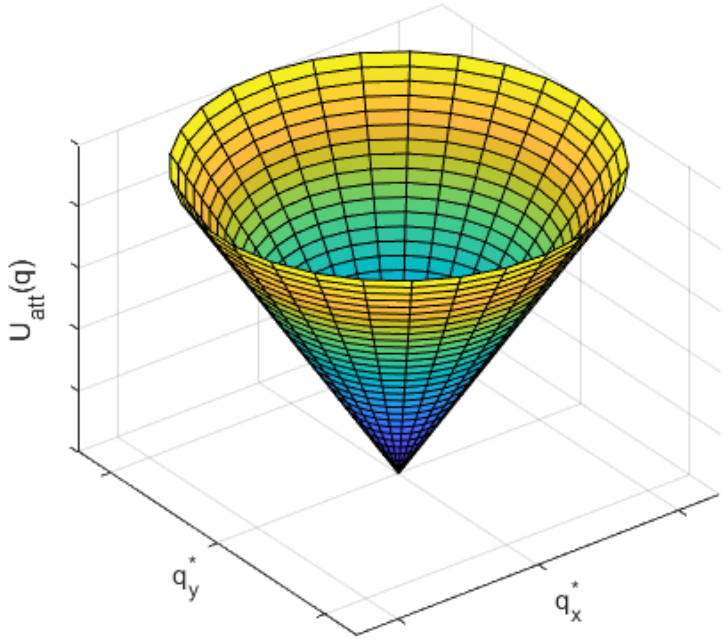
UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

# 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.

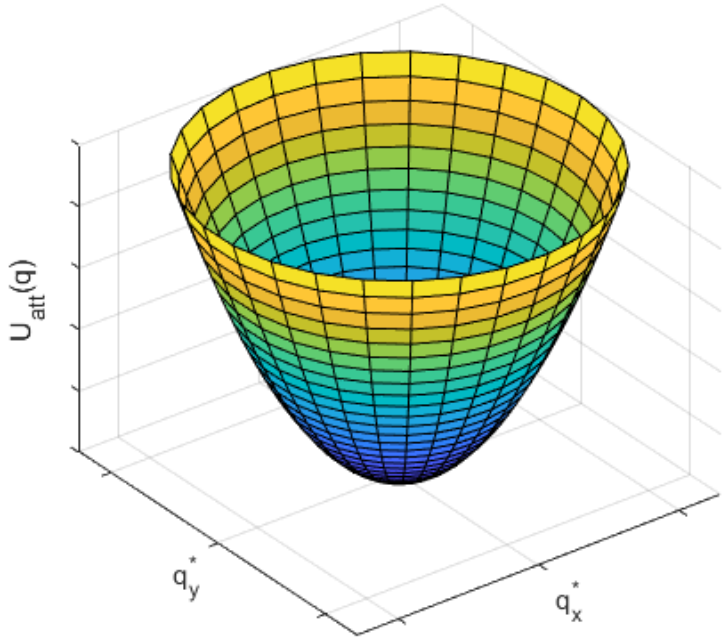
$$U_{att}(q) = \zeta ||e(q)||$$

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




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

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



UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

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

Funkcja stożkowa generuje gradient o stałej długości (niezależny od odległości robota od celu. Powoduje to, że w przypadku bardzo małych odległości pomiędzy punktem docelowym a robotem powstanie nadal stosunkowo duża siła.

Z drugiej strony, funkcja paraboliczna dostarcza gradient liniowo zależny od odległości, co może spowodować bardzo duże wartości przy znacznie oddalonych pozycjach docelowych.

W praktycznych implementacjach stosuje się połączenie tych dwóch funkcji, dzięki czemu uzyskujemy spadek wartości gradientu przy dojeżdżaniu do punktu docelowego i stałą wartość przy odległych pozycjach docelowych.

W celu zapewnienia płynnego przejścia wartości gradientu przez graniczną wartość przełączania pomiędzy potencjałami należy dodać odpowiedni człon do potencjału przyciągającego.

UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

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

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

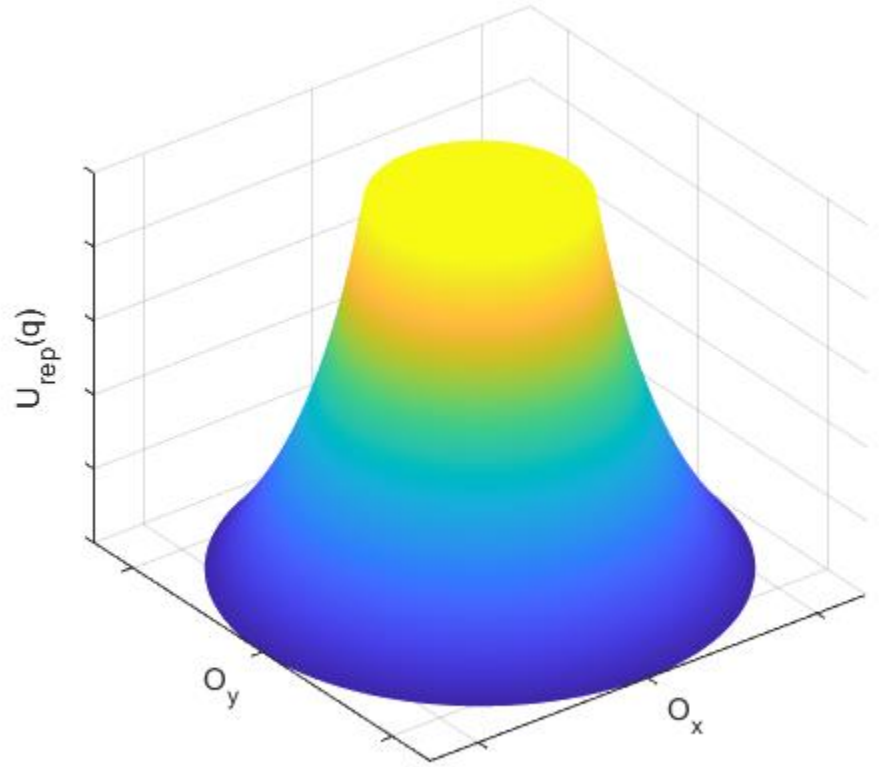
gdzie: $d_g^*$ jest wartością graniczną, tj. od tej odległości długość wektora gradientu maleje z razem z odległością od punktu docelowego pozwalając na płynne zatrzymanie się robota.

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

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

# 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*) $c_{Oi}(q) = q - O_i$



UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

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

Warto pokreślić, 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:

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

Powyższe wyrażenie prowadzi do następującego gradientu:

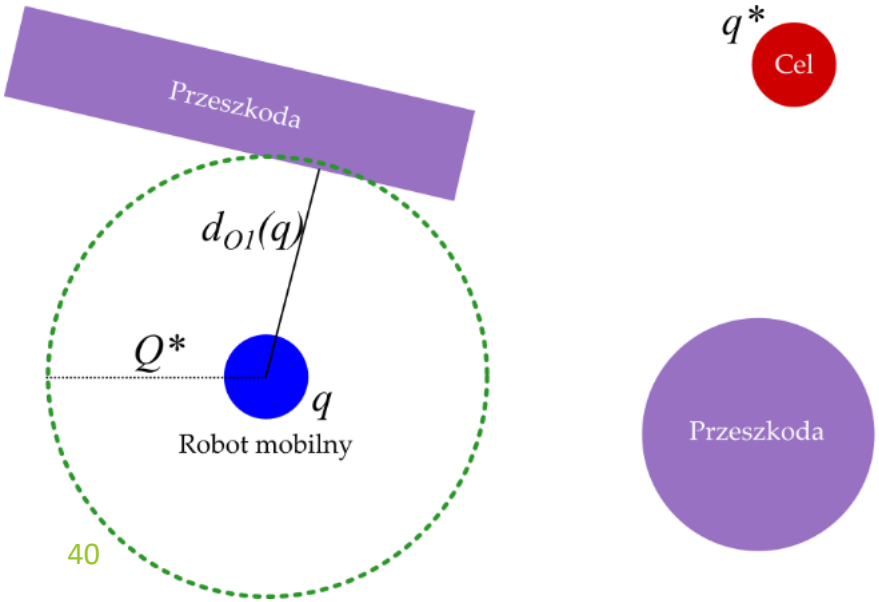
$$\nabla U_{rep}^{O_i}(q) = \begin{cases} \eta\left(\frac{1}{Q^*} - \frac{1}{||c_{O_i}||}\right)\frac{c_{O_i}}{||c_{O_i}||^2}, & jeżeli\ ||c_{O_i}|| \leq Q^* \\ 0, & w\ pozostałych\ przypadkach \end{cases}$$

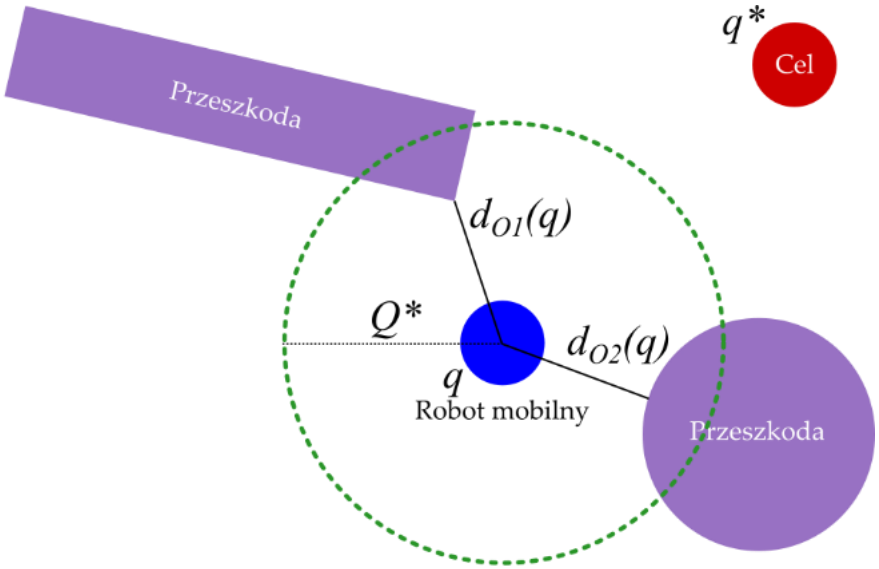
UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

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

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

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



UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

# APF – Funkcja potencjału

Całościowy potencjał jest sumą potencjału przyciągającego oraz sumy potencjałów odpychających:

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

A ich gradient:

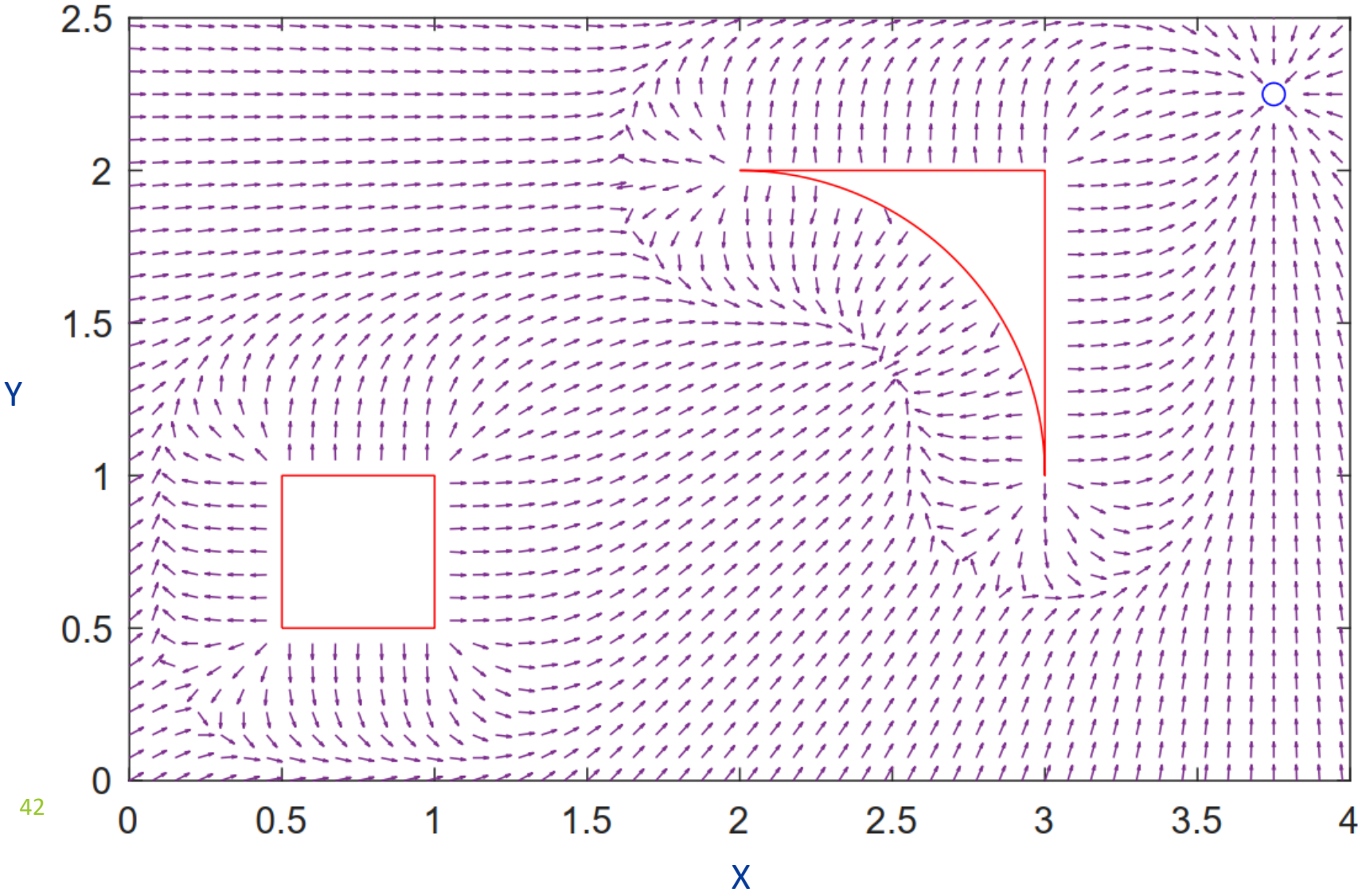
$$\nabla U(q) = \nabla U_{att}(q) + \nabla U_{rep}(q)$$

Siła jest równa gradientowi z odwrotnym znakiem:

$$F(q) = -\nabla U(q)$$

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

# APF – Suma potencjałów przyciągającego i odpychających



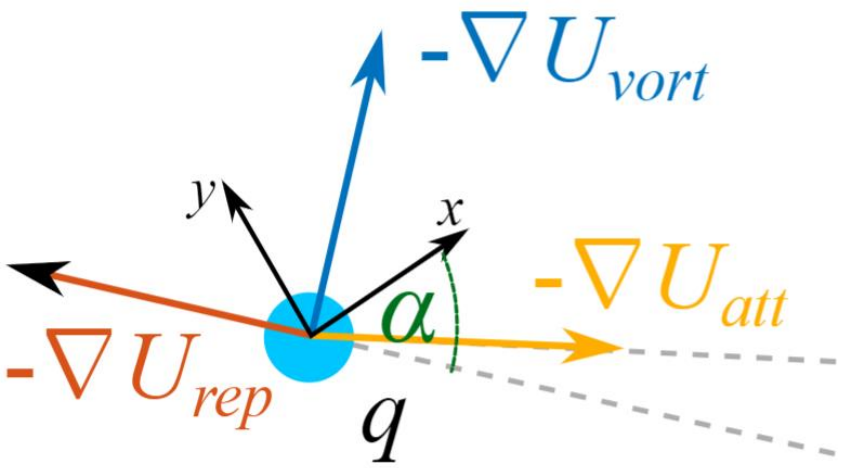
UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

# APF – potencjał wirowy (ang. *vortex potential*)

Jedną z modyfikacji APF jest zastosowanie transformacji funkcji odpychającej, aby przeszkody nie odpychały robota od siebie, a skierowały się wzdłuż jego konturów. Transformacja polega na przemnożeniu gradientu potencjału odpychającego przez macierz obrotu (dwuwymiarową w analizowanym przypadku):

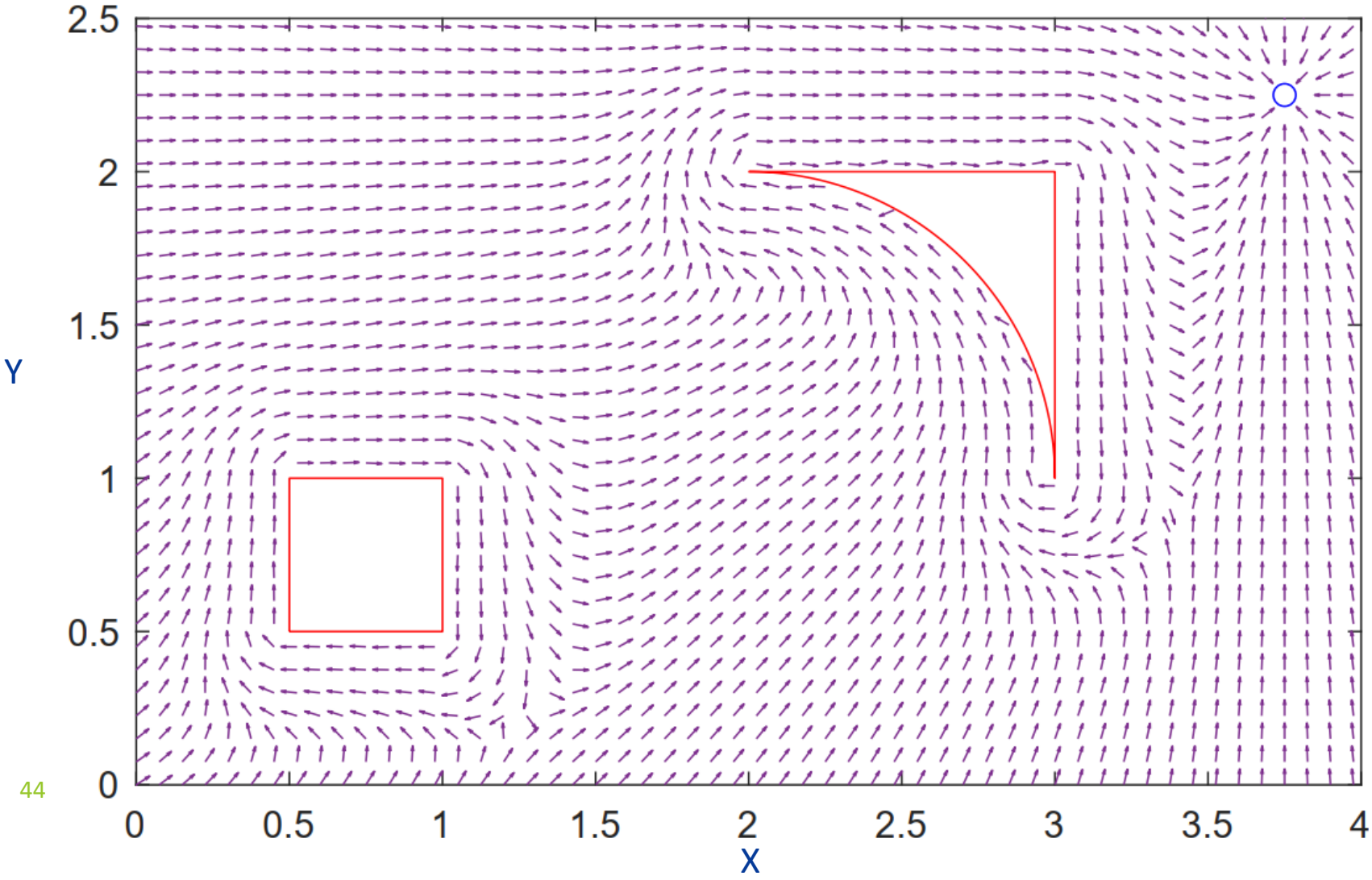
$$\nabla U_{vort}(q) = \begin{bmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{bmatrix} \cdot \nabla U_{rep}(q)$$

gdzie: $\gamma$ wynosi $\pm 90°$ i określa kierunek zgodny lub przeciwny do ruchu wskazówek zegara.



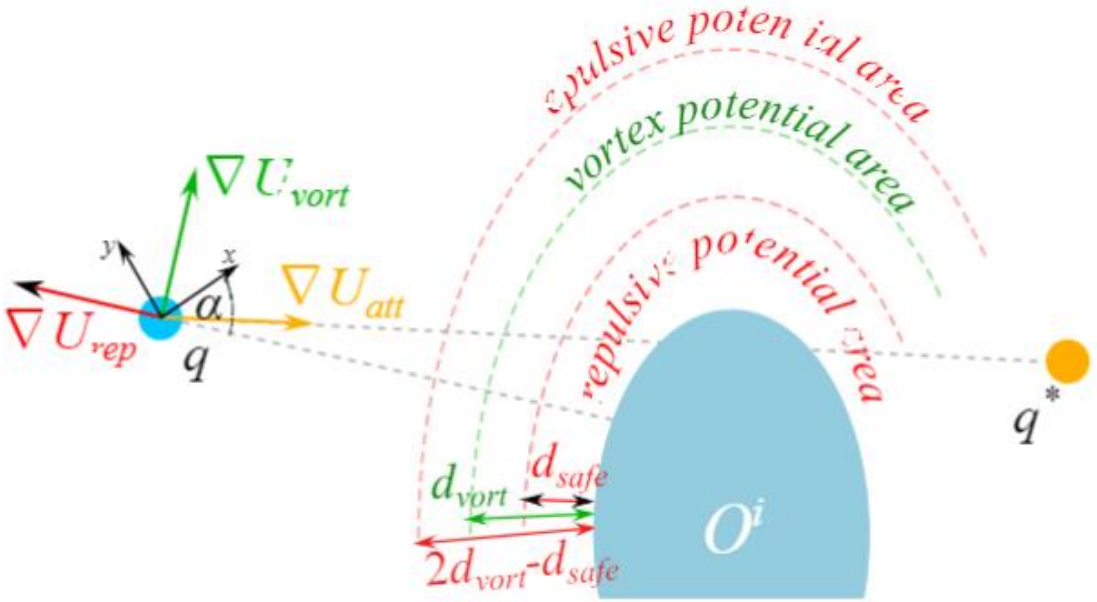
UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

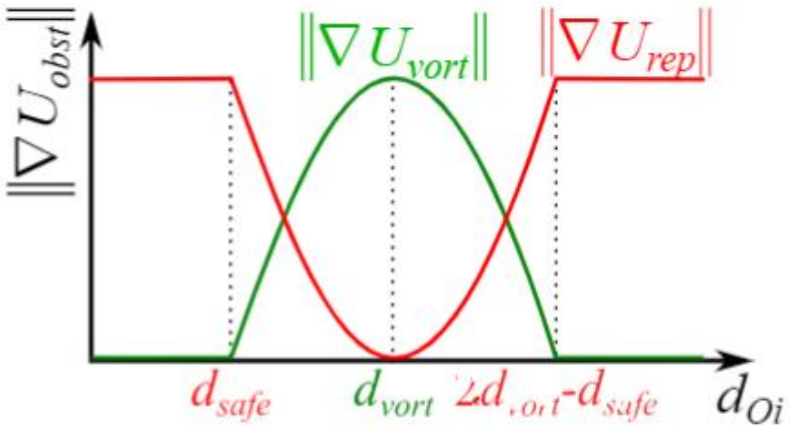
# APF – Suma potencjałów przyciągającego i wirowych



UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

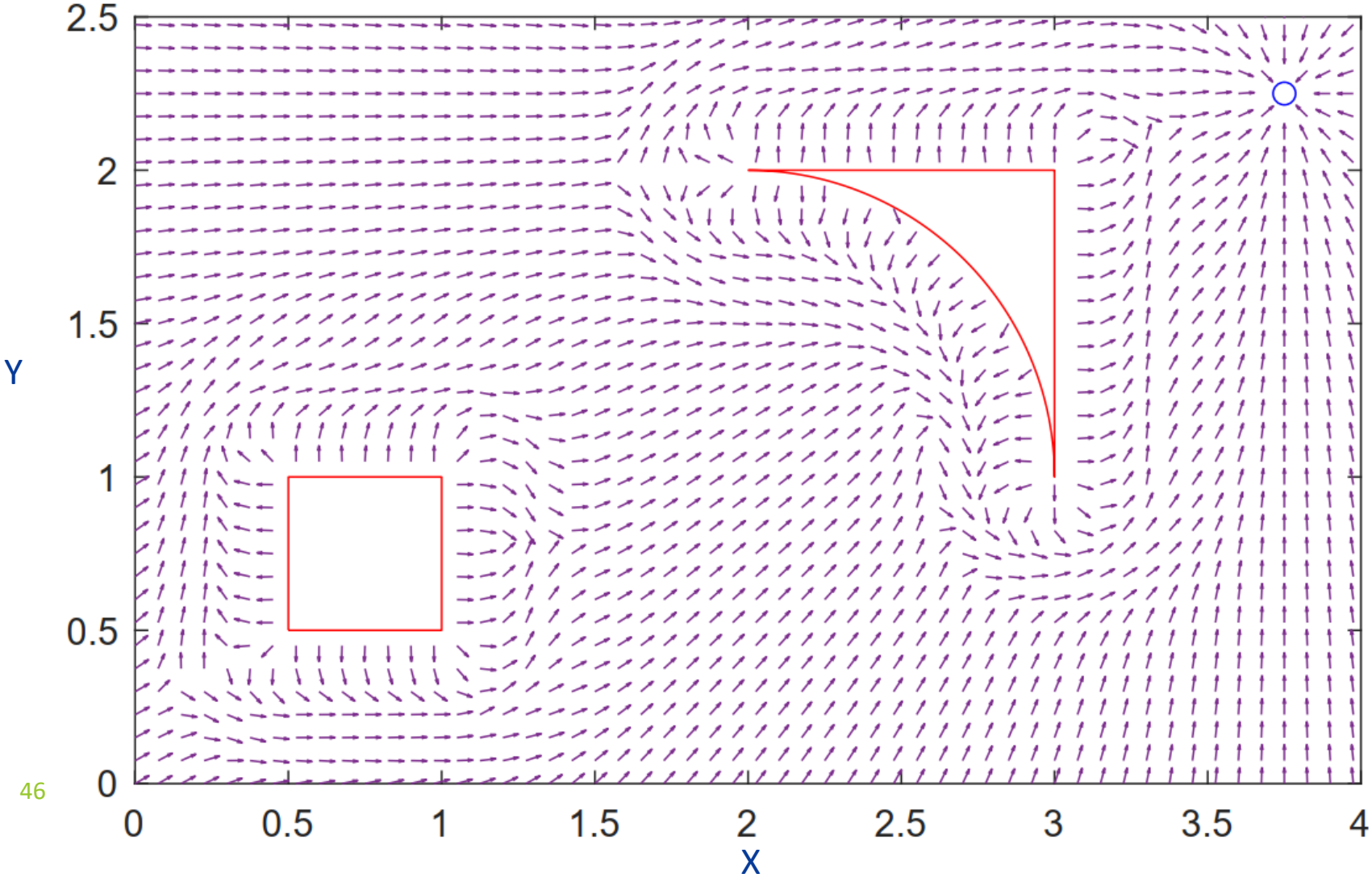
# Safe APF – Suma potencjałów przyciągającego i kombinacji potencjałów odpychających i wirowych



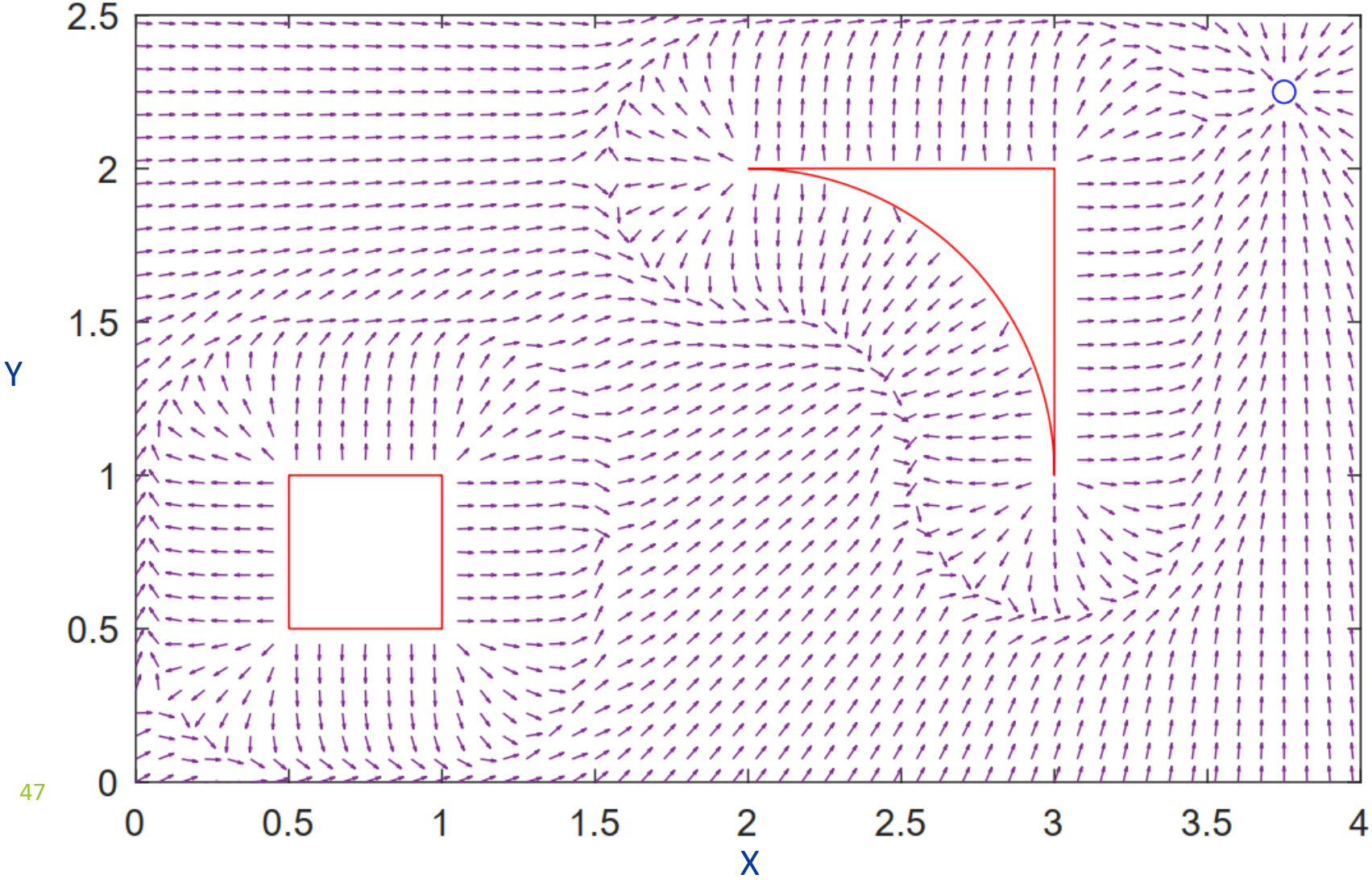
UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

# Safe APF – Suma potencjałów przyciągającego i kombinacji potencjałów odpychających i wirowych ($d_{vort}$ = 0.3 m)



UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

# Safe APF – Suma potencjałów przyciągającego i kombinacji potencjałów odpychających i wirowych ($d_{vort}$ = 0.5 m)



UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

# APF – interpretacja siły

Wektor siły możemy zinterpretować na kilka sposobów:

- Jako uogólnioną siłę przyłożoną do robota uwzględniając jego dynamikę

$$\tau = F(q)$$

- Jako uogólnione przyśpieszenie:

$$\ddot{q} = F(q)$$

- Jako uogólnione prędkości:

$$\dot{q} = F(q)$$

Tylko ostatnia metoda zapewnia asymptotyczną stabilności wokół punktu zadanego.

UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

# APF – wyznaczenie prędkości liniowej i kątowej robota

Wyznaczenie prędkości liniowej stycznej do toru ruchu robota oraz orientacji w jakim kierunku robot powinien się poruszać jest następujące:

$$v^{ref} = Limit\left(\left|\left|\nabla U(q)\right|\right|, v^{max}\right)$$

$$\theta^{ref} = arctan\left(\frac{-\nabla U^{y}(q)}{-\nabla U^{x}(q)}\right)$$

gdzie: $v^{max}$ jest maksymalną prędkością liniową robota.

UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

# APF – wyznaczenie prędkości liniowej i kątowej robota

Dodatkowym zabezpieczeniem przed poruszaniem się z zbyt dużą prędkością w niewłaściwym kierunku względem wyznaczonej orientacji robota jest uwarunkowanie liniowej prędkości od uchybu orientacji robota względem wartości referencyjnej:

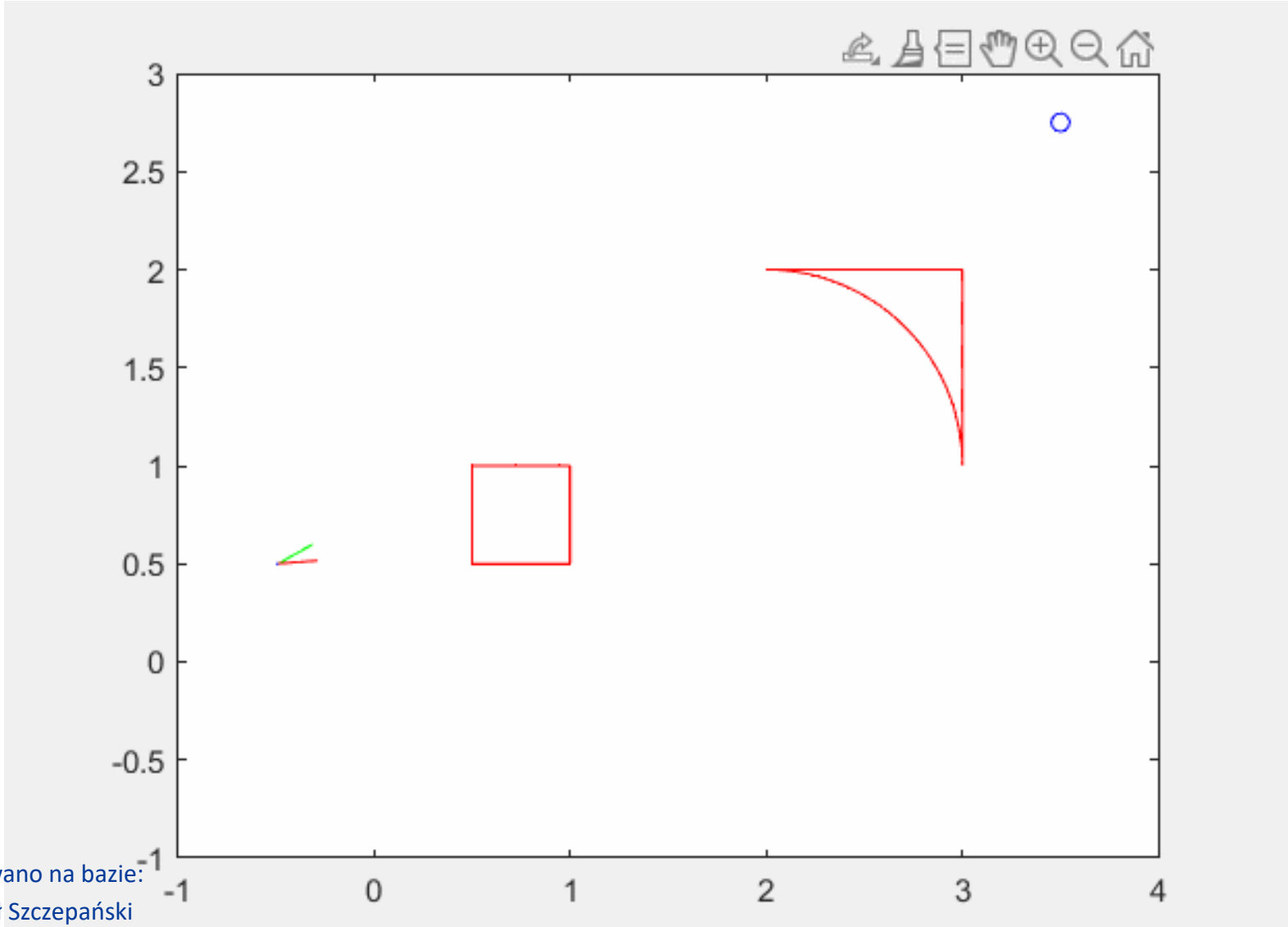
$$v^{ref} = \begin{cases} Limit\left(\alpha\left|\left|\nabla U(q)\right|\right|, v^{max}\right), & jeżeli\ abs\left(\theta_{error}\right) \leq \theta_{error}^{max} \\ 0, & w\ pozostałych\ przypadkach \end{cases}$$

$$\alpha = \frac{\theta_{error}^{max} - abs\left(\theta_{error}\right)}{\theta_{error}^{max}}$$

gdzie: $\theta_{error}^{max}$ jest granicznym uchybem powodującym zatrzymanie prędkości liniowej.

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

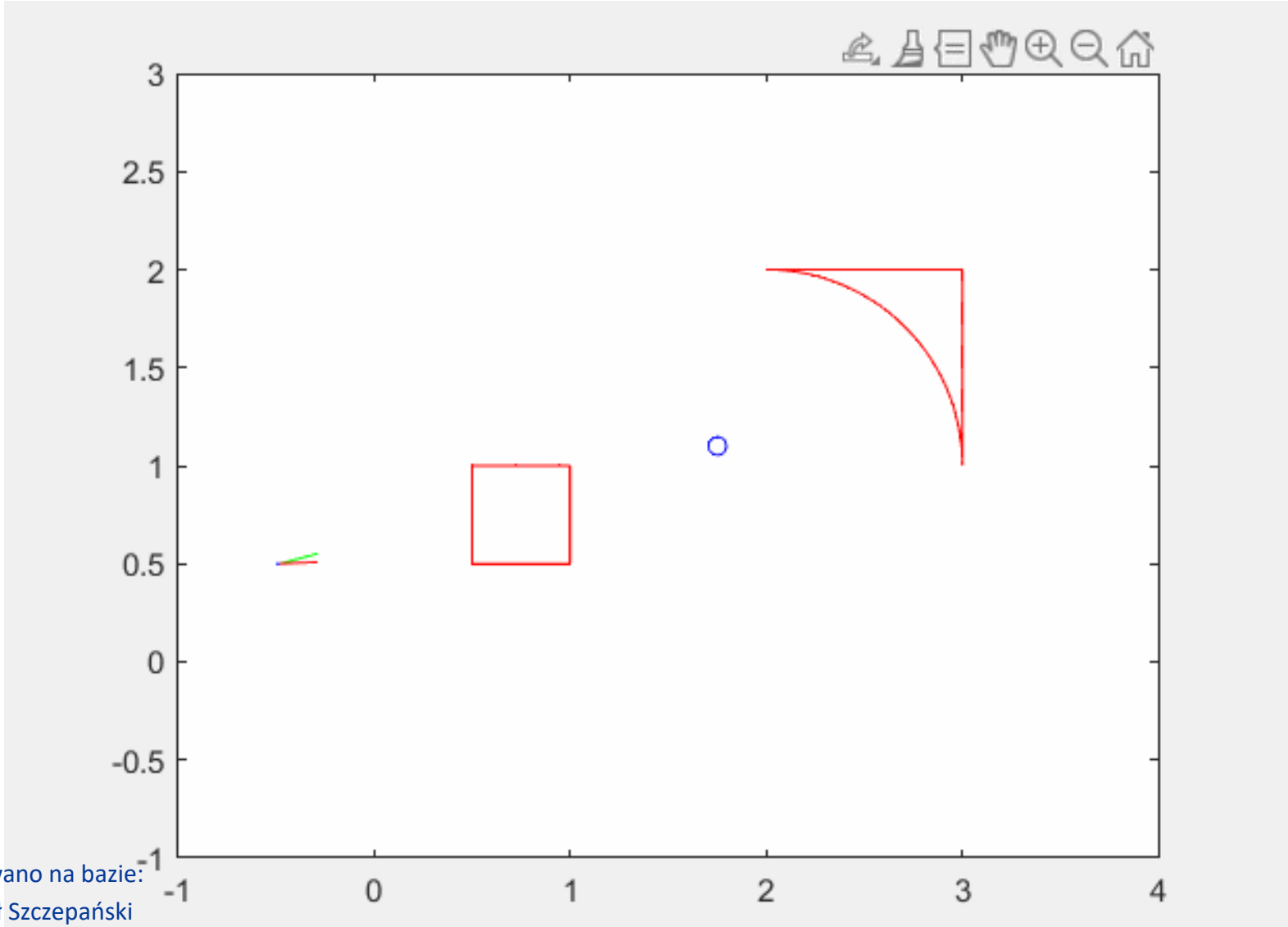
# APF – Przykład 1




Wygenerowano na bazie:
Autor: Rafał Szczepański
Link: https://www.mathworks.com/matlabcentral/fileexchange/126455-artificial-potential-field-path-planning-algorithm

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

# APF – Przykład 2

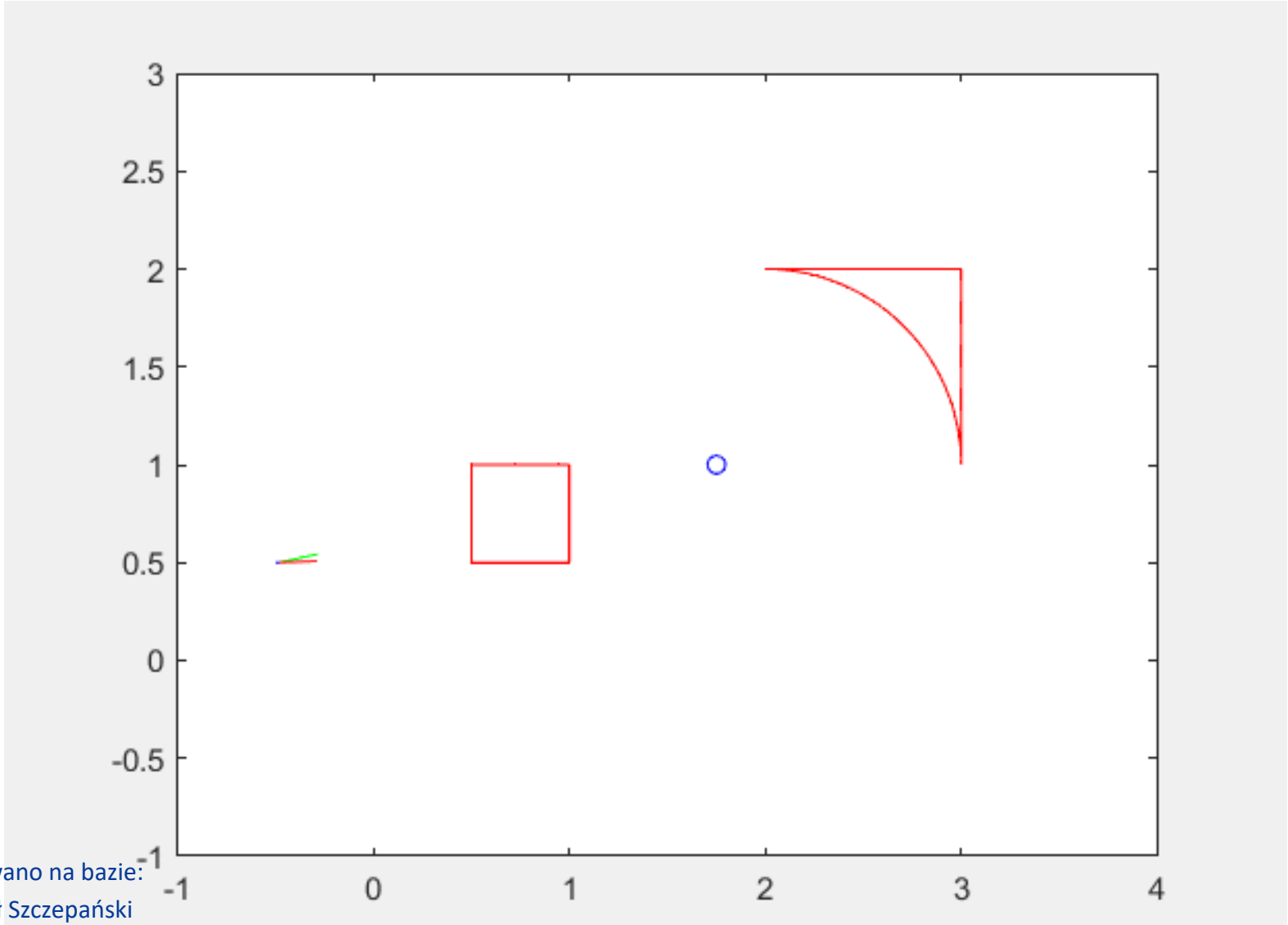





Wygenerowano na bazie:
Autor: Rafał Szczepański
Link: https://www.mathworks.com/matlabcentral/fileexchange/126455-artificial-potential-field-path-planning-algorithm

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

# APF – Przykład 3

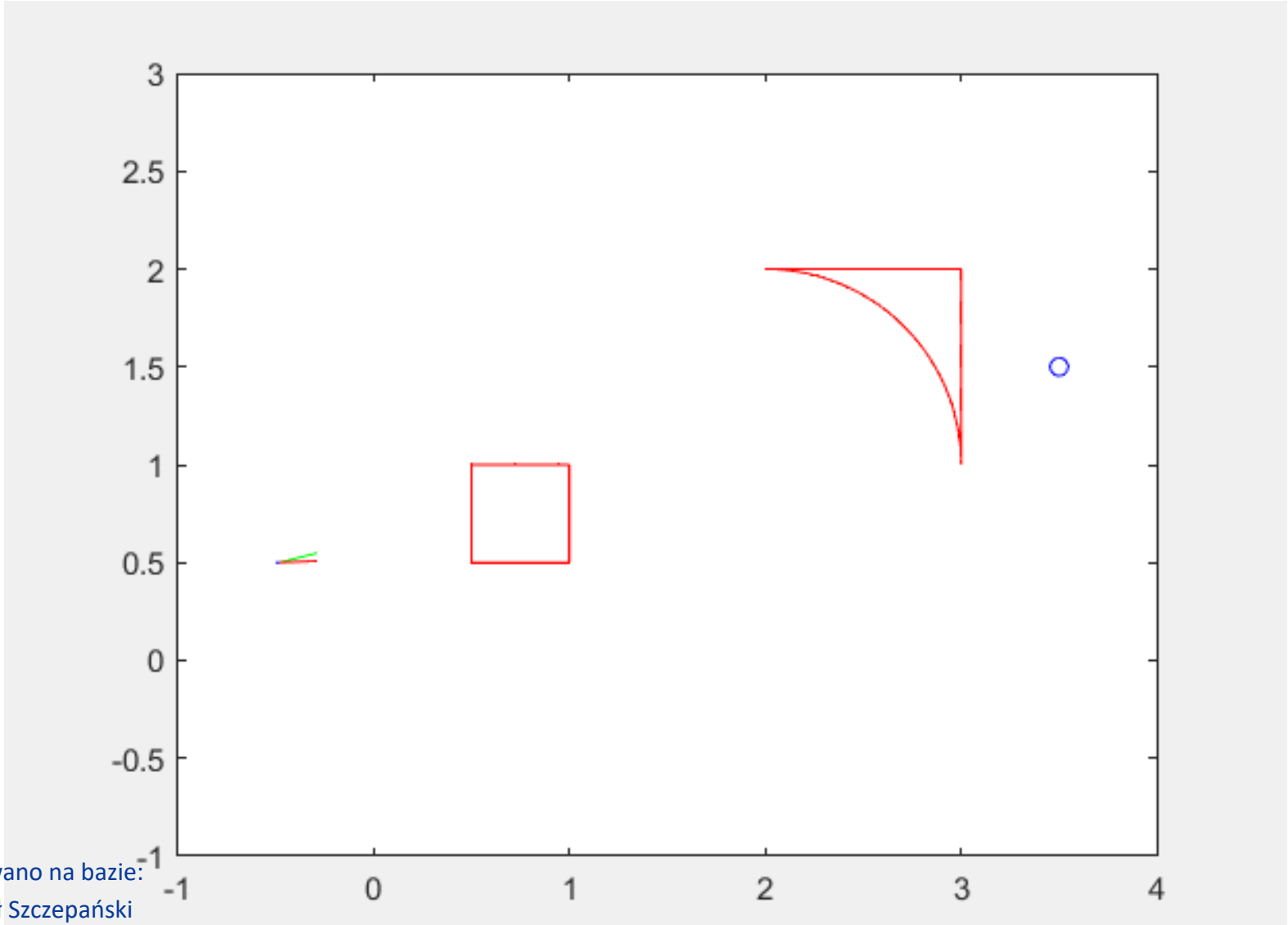





Wygenerowano na bazie:
Autor: Rafał Szczepański
Link: https://www.mathworks.com/matlabcentral/fileexchange/126455-artificial-potential-field-path-planning-algorithm

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

# APF – Przykład 4






Wygenerowano na bazie:
Autor: Rafał Szczepański
Link: https://www.mathworks.com/matlabcentral/fileexchange/126455-artificial-potential-field-path-planning-algorithm

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

# Algorytm dynamicznego okna

## (ang. *Dynamic Window Approach*, DWA)

Algorytm dynamicznego okna już wykorzystywaliśmy w poprzednim ćwiczeniu. Jest to jeden z podstawowych algorytmów planowania ścieżki lokalnej, który jest jedną z bibliotek ROS. Sam algorytm można podzielić na dwie główne części:

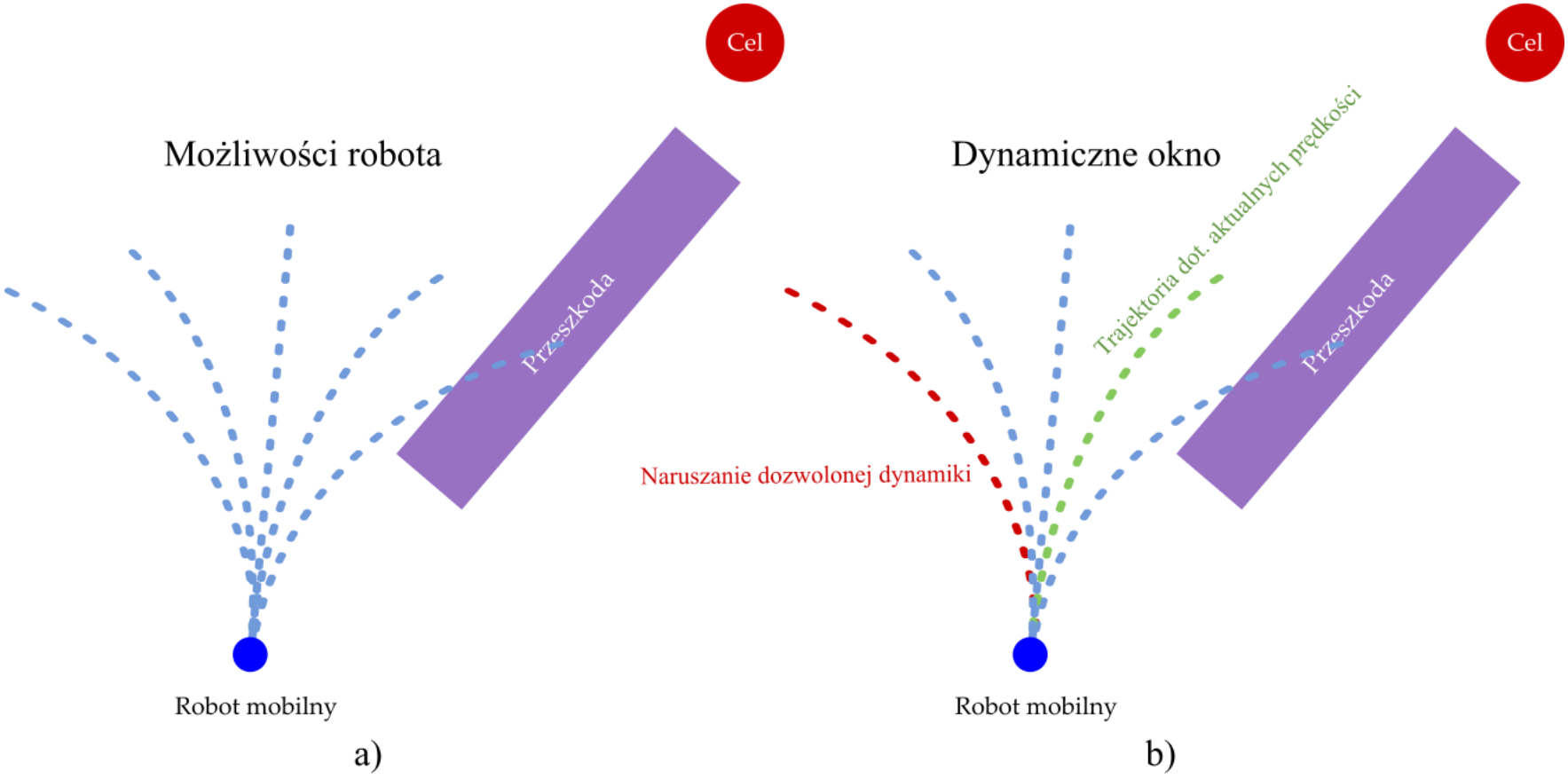
- Przeszukiwanie przestrzeni rozwiązań,
- Optymalizacja.

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.

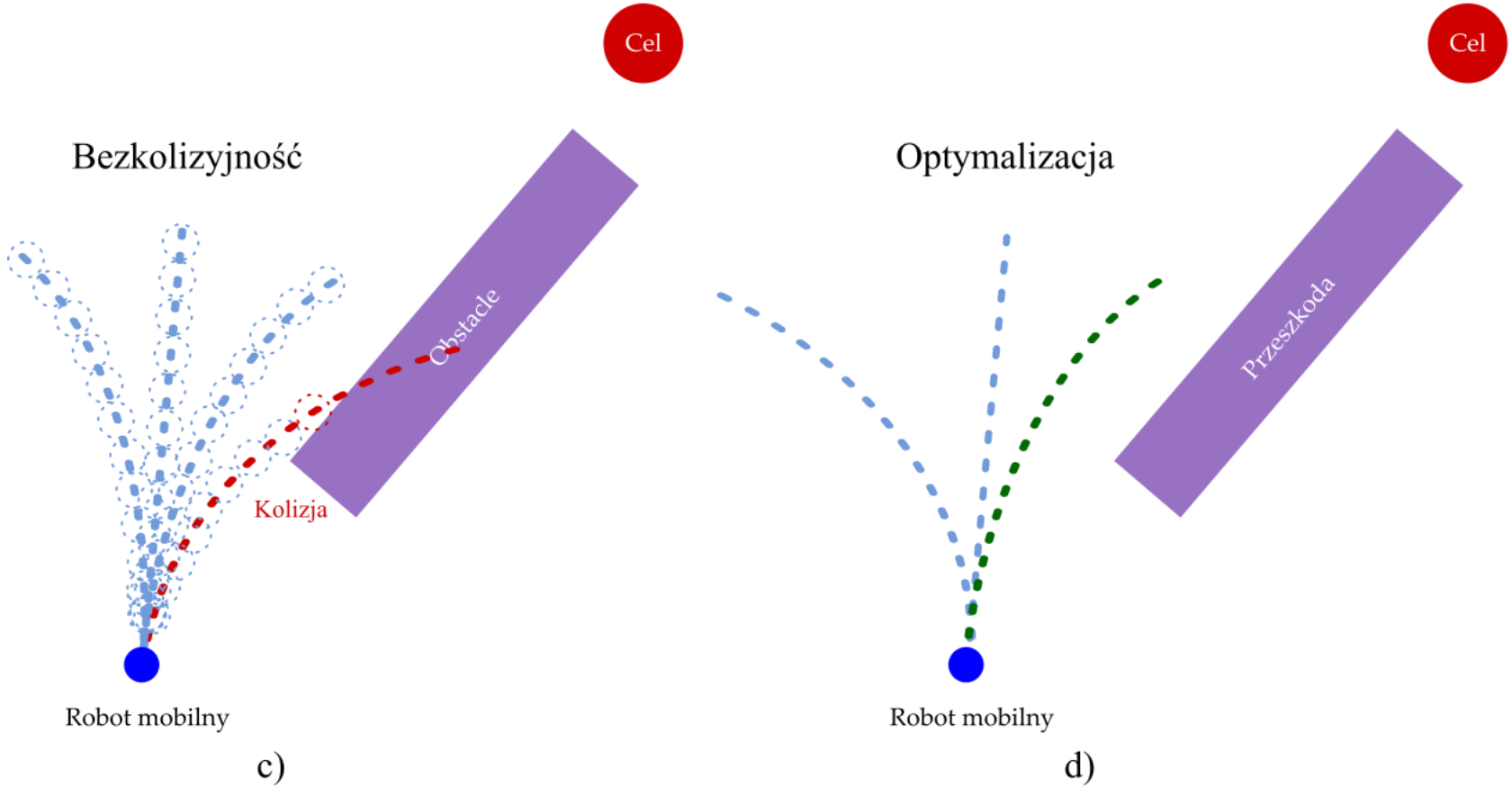
UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

# DWA - wizualizacja



UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

# DWA - wizualizacja



UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

# 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 $(v, \omega)$. 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 $(v, \omega)$ jest możliwość zatrzymania się robota zanim osiągnie najbliższą przeszkodę na wygenerowanej trajektorii.
- **Dynamiczne okno**: zakładamy ograniczone przepieszczenia robota. W związku z tym uwzględniając aktualne prędkości robota $(v_R, \omega_R)$ prędkości mogą się zmieniać w ograniczonym zakresie zdefiniowanym przez maksymalne przyśpieszenie liniowe i kątowe.

UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

# DWA - optymalizacja

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

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

gdzie: $\alpha$, $\beta$ i $\gamma$ to parametry funkcji celu, a $\sigma$ 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.

UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

# DWA – maksymalne prędkości i dynamiczne okno

Zacznijmy od wygenerowania możliwych prędkości liniowych i kątowych w zakresie maksymalnych wartości jakie może zapewnić robot. Dodatkowo, zakładając, że robot aktualnie porusza się z prędkościami $(v_R, \omega_R)$ i zakładając maksymalne przyśpieszenia równe $(a_R^{max}, \epsilon_R^{max})$ możemy zdefiniować dolną oraz górną granicę dopuszczalnych prędkości:

$$v^{max}(k) = v_R(k-1) + \int_{t(k-1)}^{t(k)} a_R^{max} d\tau \qquad v^{min}(k) = v_R(k-1) - \int_{t(k-1)}^{t(k)} a_R^{max} d\tau$$

$$\omega^{max}(k) = \omega_R(k-1) + \int_{t(k-1)}^{t(k)} \epsilon_R^{max} d\tau \qquad \omega^{min}(k) = \omega_R(k-1) - \int_{t(k-1)}^{t(k)} \epsilon_R^{max} d\tau$$

gdzie: $\mathrm{t(k)}$ oznacza czas w $k$-tej dyskretnej chwili czasu.

UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

# DWA – predykcja przyszłych pozycji robota

Następnie, przyjmując pewną rozdzielczość lub dzieląc zakres na $N$ części uzyskujemy dwuwymiarową przestrzeń przeszukiwań. Kolejnym krokiem jest predykcja przyszłych pozycji (tj. trajektorii robota). W tym celu można skorzystać następujących równań:

$$\theta(k+1) = \theta(k) + \omega_R(k) \cdot \big(t(k+1) - t(k)\big)$$

$$x_R(k+1) = x_R(k) + v_R(k) \cdot cos\big(\theta(k+1)\big) \cdot \big(t(k+1) - t(k)\big)$$

$$y_R(k+1) = y_R(k) + v_R(k) \cdot sin\big(\theta(k+1)\big) \cdot \big(t(k+1) - t(k)\big)$$

gdzie: $\theta$ jest orientacją robota, $x_r$ i $y_r$ są pozycjami odpowiednio w osi $x$ oraz $y$, a $t$ jest czasem. Predykcje możemy wykonać jako np. pięć sekund jazdy ze stałym okresem próbkowania wynoszącym 0.1 sekundy.

UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

# DWA – warunek bezkolizyjności

Jeżeli mamy już trajektorię dla każdego zestawu prędkości kątowej i liniowej $(v, \omega)$, to jesteśmy w stanie upewnić się, że trasy są bezkolizyjne, a dodatkowo, że robot będzie w stanie się zatrzymać zanim zahamuje. Można to zapisać łącząc dwa następujące warunki:

$$v \leq \sqrt{2 \cdot dist(v, \omega) \cdot a_b} \wedge \omega \leq \sqrt{2 \cdot dist(v, \omega) \cdot \epsilon_b}$$

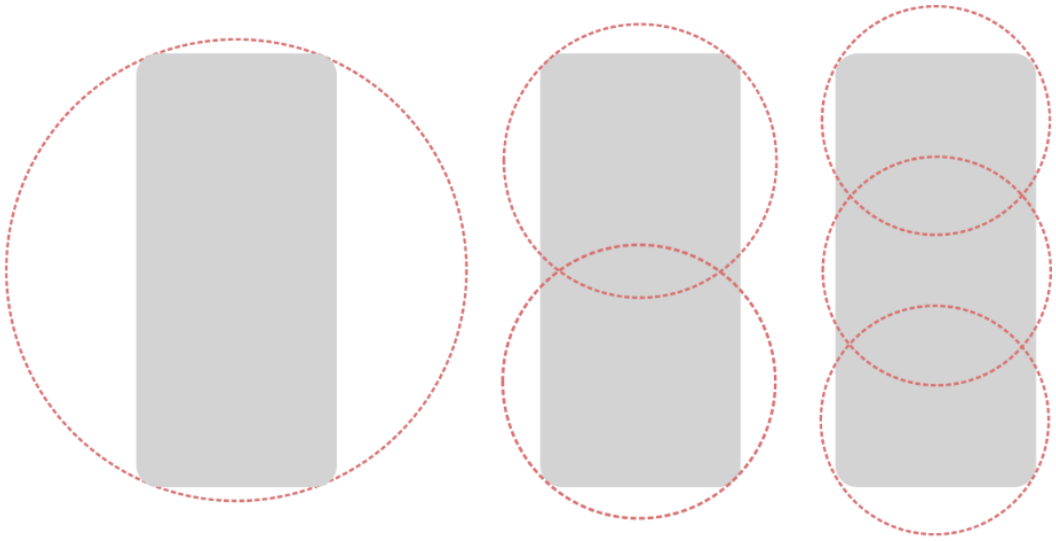
gdzie: $a_b$ i $\epsilon_b$ są przyśpieszeniami przy hamowaniu. Zakładając płynność ruchu, można przyjąć dla uproszczenia, że $a_b = a_R^{max}$ i $\epsilon_b = \epsilon_R^{max}$. W ten sposób wykluczając kolizyjne trajektorie, otrzymujemy zbiór par prędkości liniowej i kątowej do procesu optymalizacji. Dla każdej pary obliczana jest wartość wskaźnika jakości, a następnie wybierana para, która zapewnia wartość maksymalną ze wszystkich analizowanych możliwości.

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

# DWA – warunek bezkolizyjności dla robota o nieregularnym kształcie

Wydajną metodą wykrycia kolizji jest rozpatrywanie robota mobilnego poprzez zestaw okręgów lub jeden okrąg. Dzięki temu kolizja występuje gdy odległość od przeszkody jest mniejsza bądź równa promieniowi okręgu. Wykorzystując dane ze skanera laserowego, który jest zamontowany w środku okręgu o promieniu $r$, otrzymujemy prosty warunek na sprawdzenie czy pomiar $d$ jest kolizyjny:

$$kolizja(d) = \begin{cases} prawda, & jeżeli\ d < r \\ fałsz, & w\ pozostałych\ przypadkach \end{cases}$$

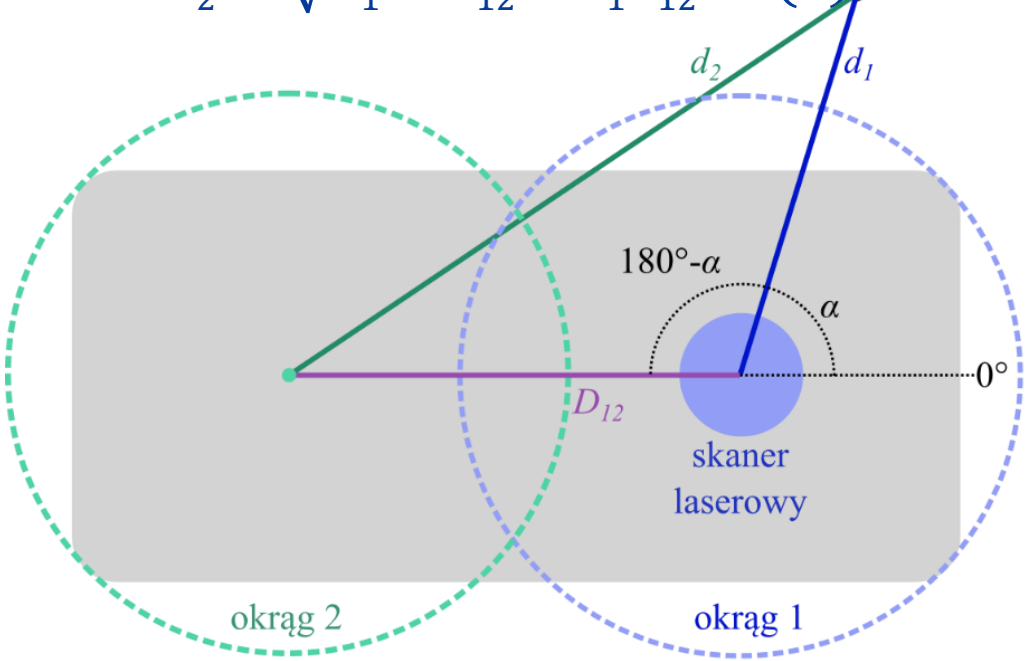


UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

# DWA – warunek bezkolizyjności dla robota o nieregularnym kształcie

Stosując twierdzenie cosinusów oraz wzór redukcyjny oraz znając pomiar względem środka okręgu pierwszego ($d_1$) pod kątem α i odległość pomiędzy środkami tych okręgów ($D_{12}$) możemy wyznaczyć odległość od środka drugiego okręgu ($d_2$):

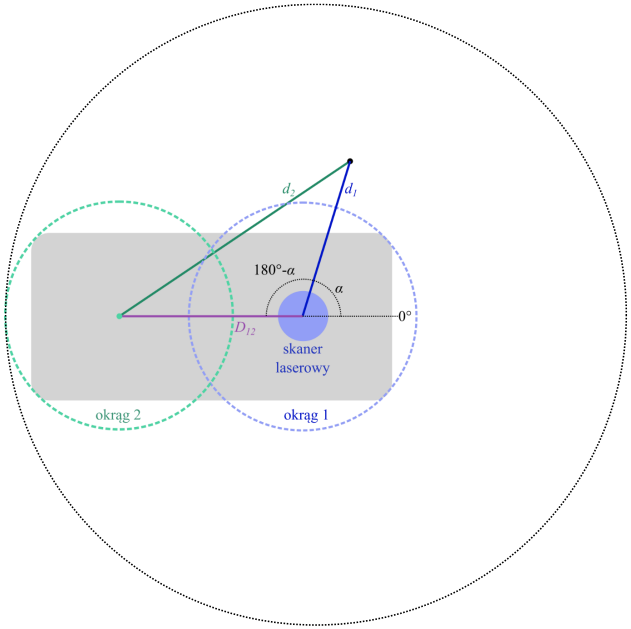
$$d_2 = \sqrt{d_1^2 + D_{12}^2 + d_1 D_{12} cos(\alpha)}$$



UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

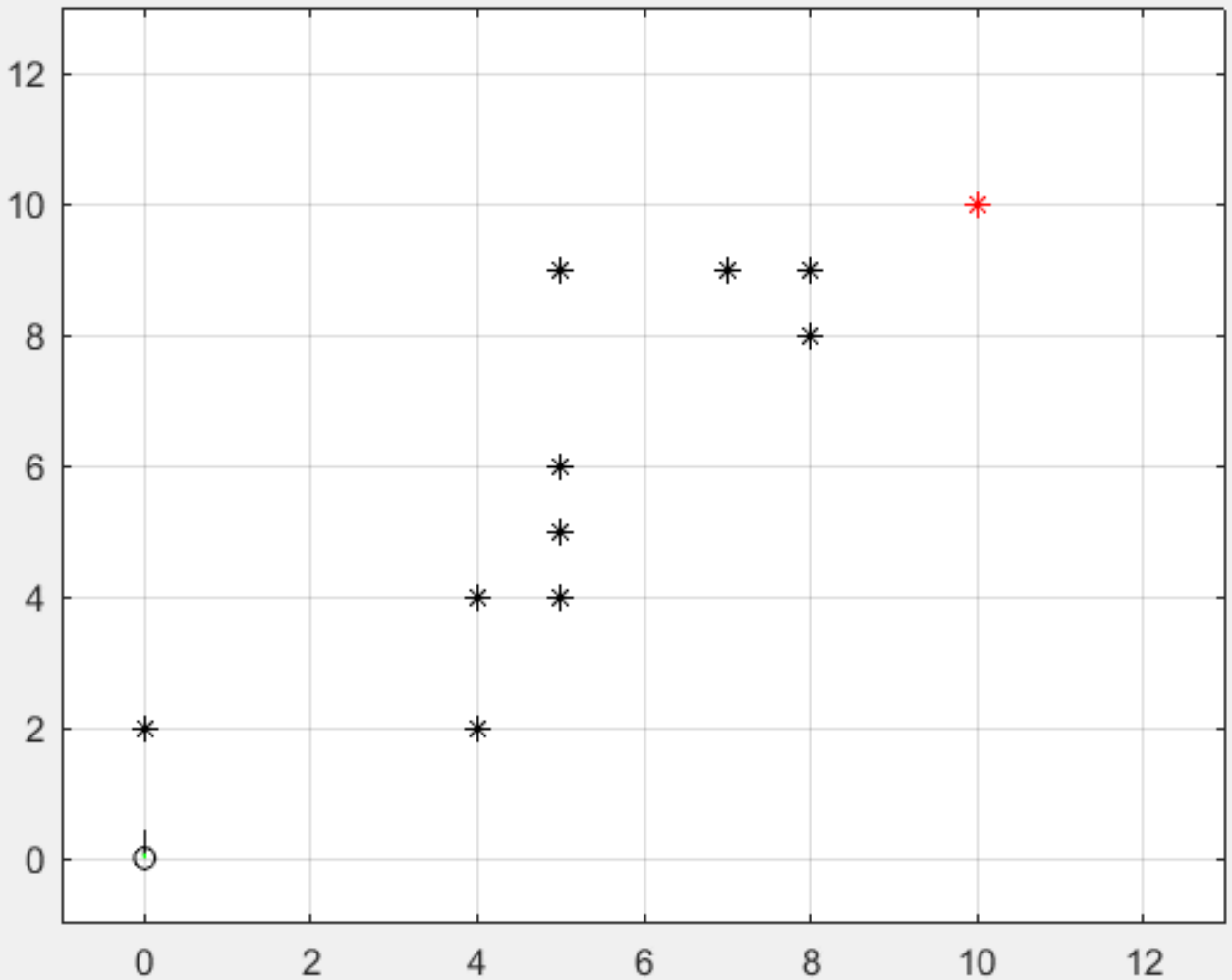
# DWA – warunek bezkolizyjności dla robota o nieregularnym kształcie

Dla skanera laserowego, który z częstotliwością 10 Hz dostarcza 1440 próbek (rozdzielczość 0.25 stopnia) zastosowanie powyższego równania może okazać się kosztowne obliczeniowo. W związku z tym, możemy zaprojektować dodatkowy okrąg o promieniu $R_{12}$, którego środek jest środkiem okręgu pierwszego, i który mieści w sobie okrąg drugi. Dzięki temu stosując najpierw porównanie odległości ze skanera laserowego oraz promienia $R_{12}$, odległość względem okręgu drugiego będziemy musieli obliczać tylko jeżeli istnieje szansa, że okrąg drugi może zostać naruszony.



UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

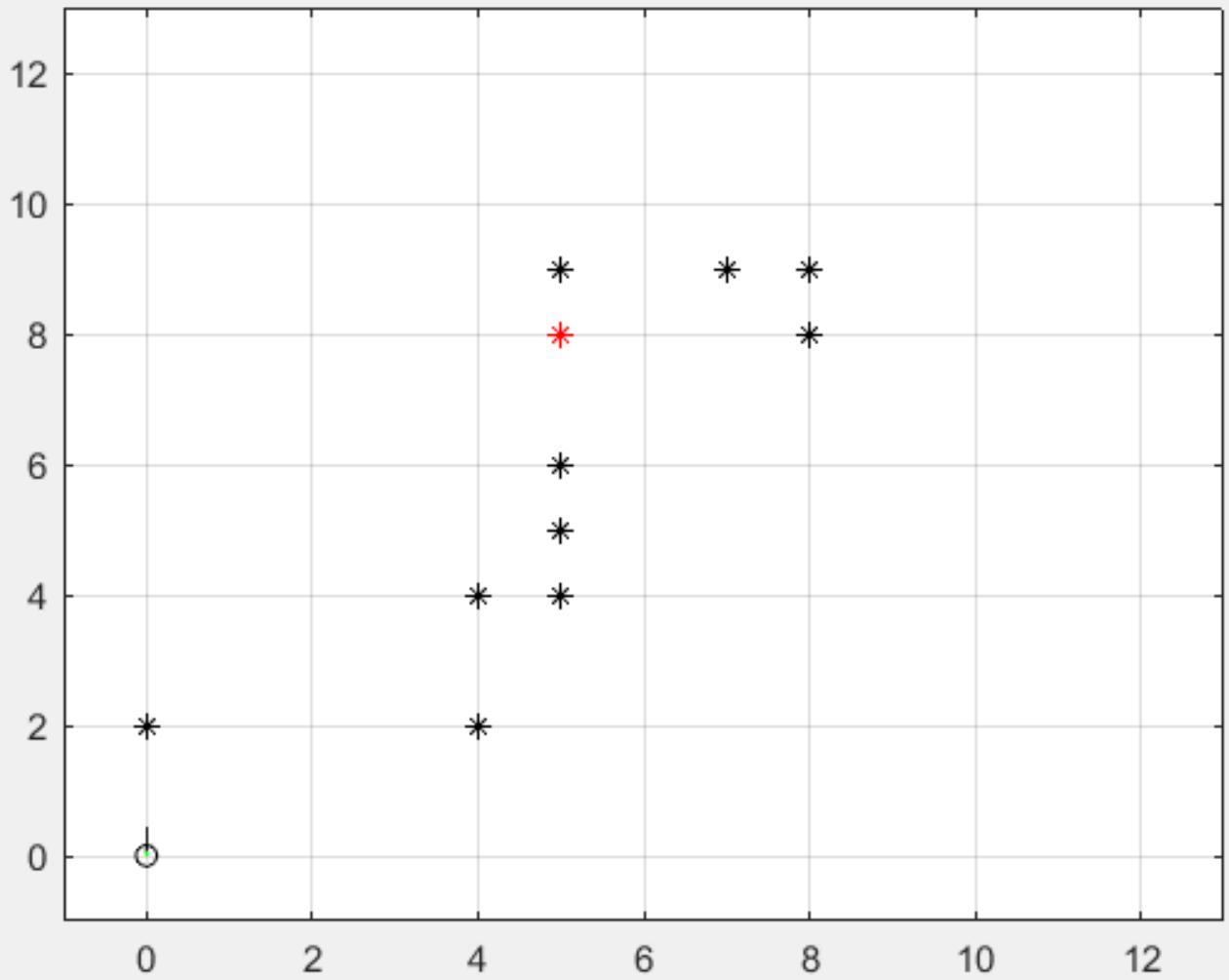
# DWA – Przykład 1




Wygenerowano na bazie:
Autor: Atsushi Sakai
Link:
https://github.com/AtsushiSakai/MATLABRobotics/blob/master/PathPlanning/DynamicWindowApproach/DynamicWindowApproachSample.m

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

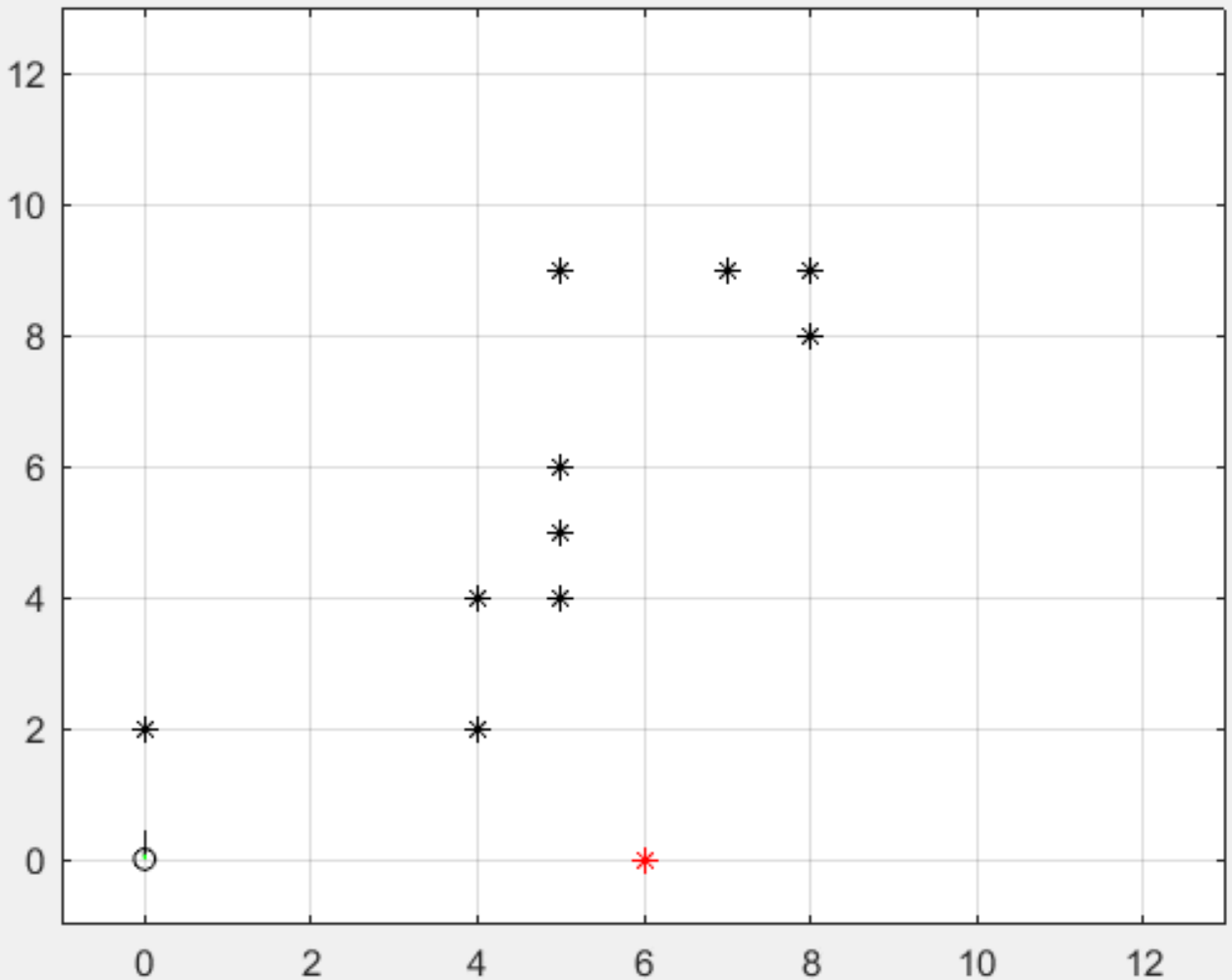
# DWA – Przykład 2




Wygenerowano na bazie:
Autor: Atsushi Sakai
Link:
https://github.com/AtsushiSakai/MATLABRobotics/blob/master/PathPlanning/DynamicWindowApproach/DynamicWindowApproachSample.m

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

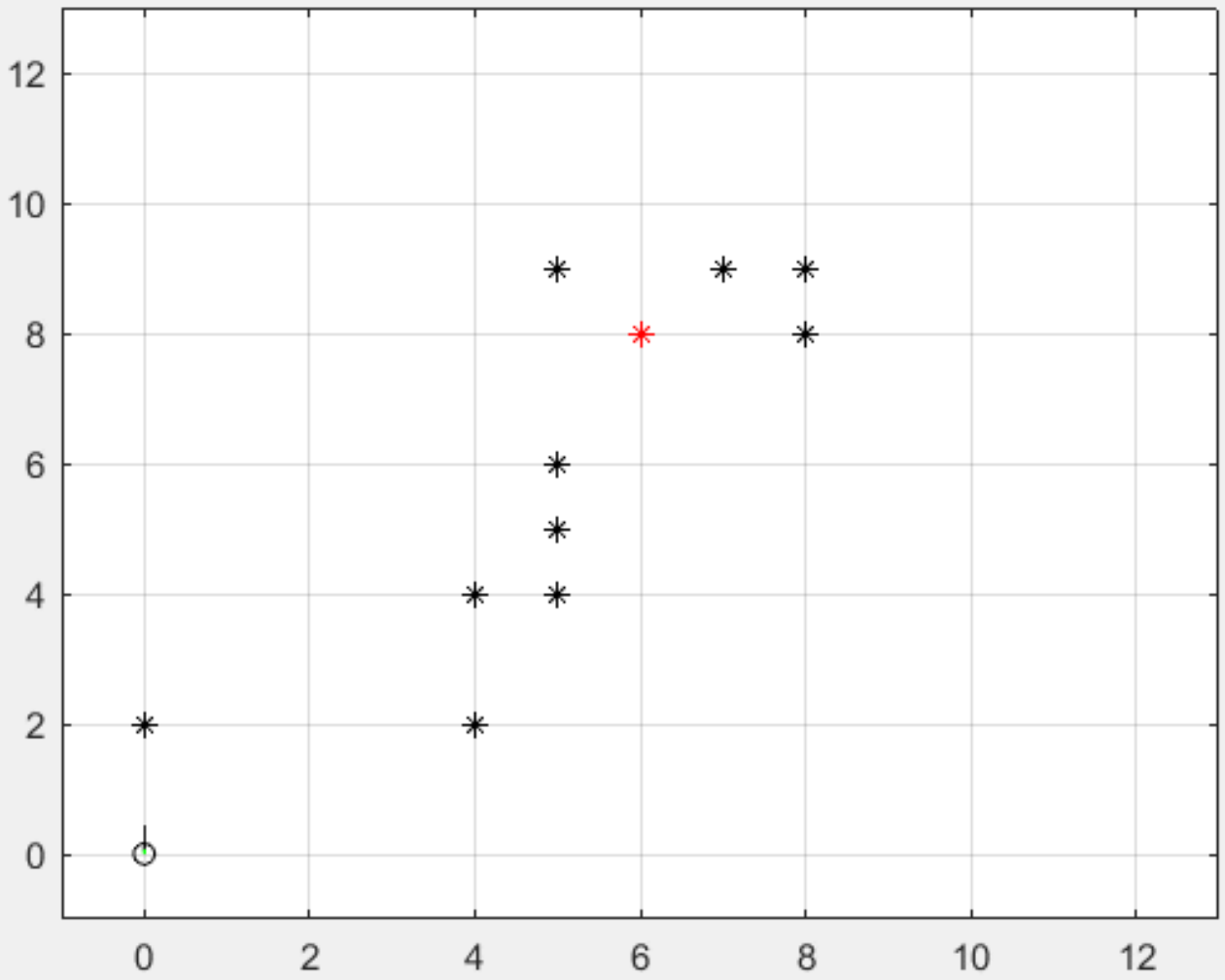
# DWA – Przykład 3




Wygenerowano na bazie:
Autor: Atsushi Sakai
Link:
https://github.com/AtsushiSakai/MATLABRobotics/blob/master/PathPlanning/DynamicWindowApproach/DynamicWindowApproachSample.m

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

# DWA – Przykład 4




Wygenerowano na bazie:
Autor: Atsushi Sakai
Link:
https://github.com/AtsushiSakai/MATLABRobotics/blob/master/PathPlanning/DynamicWindowApproach/DynamicWindowApproachSample.m