

# Roboty monitorujące skażenie środowiska - symulator V-REP

Anna Wujek  
Łukasz Korpala  
Wiktor Ślęczka

10 czerwca 2016

# Spis treści

<b>1</b>	<b>Zadanie</b>	<b>3</b>
<b>2</b>	<b>Projekt systemu</b>	<b>3</b>
2.1	Założenia . . . . .	3
2.2	Scenariusz działania . . . . .	3
2.3	Struktura systemu . . . . .	4
2.3.1	Symulator przestrzeni roboczej . . . . .	4
2.3.2	Symulator chmury skażenia . . . . .	4
2.3.3	Symulator robota . . . . .	4
2.3.4	Komunikacja . . . . .	4
2.4	Algorytmy . . . . .	5
2.4.1	Algorytm poszukiwania chmury . . . . .	5
2.4.2	Algorytm bug 2 . . . . .	6
2.4.3	Algorytm otaczania chmury . . . . .	6
<b>3</b>	<b>Realizacja</b>	<b>6</b>
3.1	Komunikacja . . . . .	6
3.1.1	Roboty . . . . .	7
3.1.2	Chmura . . . . .	7
3.2	Roboty . . . . .	7
3.3	Przykładowe mapy środowiska . . . . .	9

# 1 Zadanie

Celem projektu było stworzenie symulatora mobilnej sieci ad hoc (MANET) do monitorowania skażenia środowiska naturalnego. Węzłami sieci są roboty mobilne, wyposażone w czujniki oraz komunikujące się między sobą. Zadaniem robotów jest lokalizacja chmury skażenia i otoczenie jej robotami, monitorującymi jej położenie i granice, przy założeniu utrzymania spójności sieci.

## 2 Projekt systemu

### 2.1 Założenia

- Roboty potrafią się na bieżąco lokalizować.
- System składa się z pewnej liczby robotów mobilnych.
- Roboty mobilne działają autonomicznie i potrafią same zorganizować sieć.
- Roboty są wyposażone w czujniki pozwalające im wykrywać poziom skażenia oraz przeszkody.
- Roboty są wyposażone w urządzenia pozwalające im się między sobą komunikować się między sobą.
- Chmura skażenia jest ogólnie określona i niezmienna w trakcie trwania symulacji.
- Nieznana jest mapa przestrzeni roboczej, ale znane są jej granice i kształt - prostokąt.

### 2.2 Scenariusz działania

Algorytm realizowany przez każdego robota:

1. Warunek początkowy: roboty znajdują się w dowolnym punkcie przestrzeni na mapie. Z punktu tego musi istnieć droga prowadząca do obszaru poszukiwań.
2. Przenieść się do punktu początkowego, zależnego od liczby robotów, wymiarów przestrzeni poszukiwań i zasięgu czujników, zgodnie ze wzorem:

$$max\_spaces = sensor\_range \cdot precision\_range \quad (1)$$

$$x = -length/2 \quad (2)$$

$$y = width/2 + distances * robot\_number + distances/2 \quad (3)$$

3. Wyznacz kolejny punkt docelowy, zgodnie ze wzorem:

$$fvdfv = vfdknvdk \quad (4)$$

4. Udaj się do punktu docelowego, monitorując skażenie oraz wykrywając przeszkody.

5. Jeśli wykryto przeszkodę, omiń ją algorytmem Bug2. Wyznacz następny punkt docelowy, jest aktualny został minięty.
6. Jeśli wykryto chmurę skażenia, zatrzymaj się.
7. Jeśli cały obszar poszukiwań sprawdzony, to koniec algorytmu; jeśli nie, to przejdź do punktu 3.

## 2.3 Struktura systemu

System prezentowany w projekcie składa się z następujących elementów:

- symulator przestrzeni roboczej
- symulator robotów
- symulator chmury skażenia
- komunikacja

zapewnia przekazywanie informacji pomiędzy poszczególnymi symulatorami.

### 2.3.1 Symulator przestrzeni roboczej

Symulator ten jest odpowiedzialny za wizualizację całej symulacji - przestrzeni roboczej, przeszkód, robotów oraz chmury skażenia. Generuje także fizyczne oddziaływania między elementami, np. kolizje oraz odczyty czujników (oprócz czujników skażenia).

### 2.3.2 Symulator chmury skażenia

Symulator ten jest odpowiedzialny za generowanie odczytów czujników skażenia. Odczyt jest obliczany na podstawie założonego wcześniej kształtu chmury oraz sposobu, w jaki skażenie rośnie wgłąb chmury.

### 2.3.3 Symulator robota

W systemie symulator ten będzie zwielokrotniony, aby uzyskać grupę kilku robotów. Odpowiedzialny jest za wyznaczanie sterowania robotów w zależności od odczytów z czujników i wewnętrznego imperatywu, czyli zadania, jakie roboty realizują.

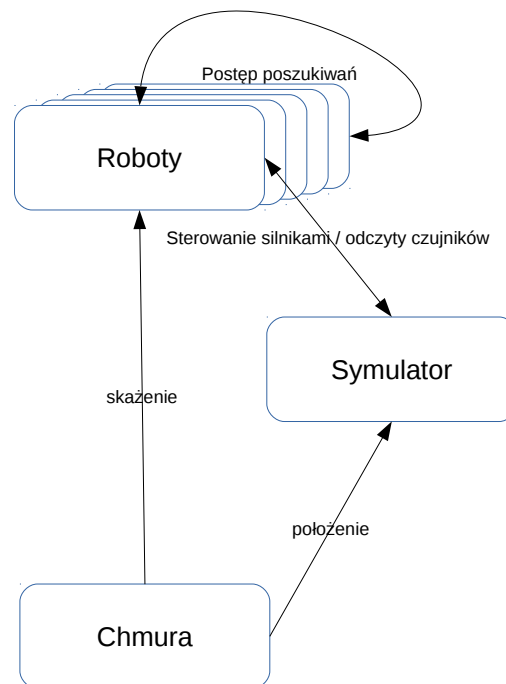
Każdy robot realizuje dokładnie ten sam algorytm

### 2.3.4 Komunikacja

Aby cały system działał poprawnie, poszczególne elementy muszą móc się ze sobą komunikować. W systemie rozróżniamy następujące kanały komunikacyjne i przesyłane informacje:

- Symulator przestrzeni roboczej  $\rightarrow$  Symulator robota:
  - odczyty z czujników (oprócz czujnika skażenia)
  - pozycja robota (położenie + orientacja)

- Symulator robota → Symulator przestrzeni roboczej:
  - sterowanie poszczególnymi silnikami
- Symulator chmury skażenia → Symulator robota:
  - odczyt czujnika skażenia
- Symulator robota → Symulator robota:
  - ostatni punkt docelowy, do którego udało się dotrzeć
  - informacja o wykrytej chmurze skażenia



## 2.4 Algorytmy

### 2.4.1 Algorytm poszukiwania chmury

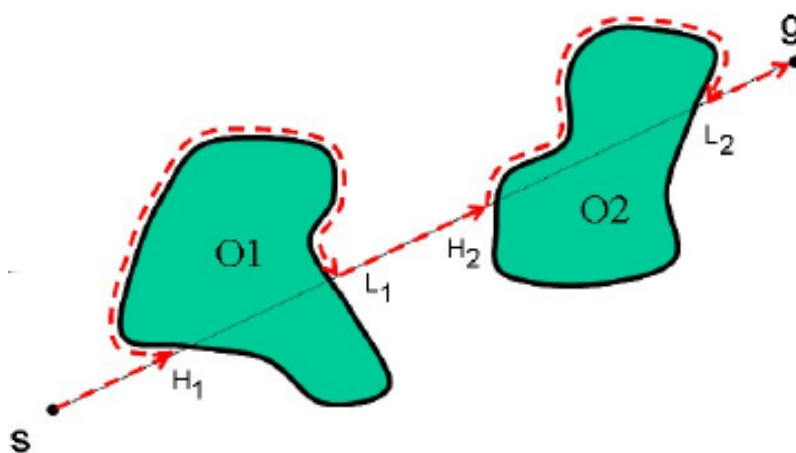
Roboty rozpoczynają pracę od ustawienia się w formacji linii. Następnie dla każdego robota wyznaczany jest następny punkt docelowy, do którego musi dojechać. Punkty docelowe wyznaczane są w taki sposób, aby roboty ponownie ustawiły się w linii. Po dojechaniu do punktu docelowego robot wysyła informację o tym zdarzeniu do innych robotów. Linia punktów docelowych jest jednocześnie miejscem synchronizacji - jeśli jakiś robot jechał wolniej (np. z powodu omijania przeszkody), to reszta na niego czeka. Powoduje to wolniejsze przeszukiwanie, ale zapewnia spójność sieci – robot nigdy nie jest oddalony od swoich sąsiadów bardziej, niż odległość do następnego punktu docelowego. Odległość ta dobrana jest w taki sposób, aby spójność sieci została zachowana.

Roboty w ten sposób poruszają się do przodu do momentu dotarcia do przeszkody, toksycznej chmury lub granicy obszaru poszukiwań. W przypadku powyższych wydarzeń, uruchamiane są odpowiednie algorytmy.

Gdy roboty dotrą do końca obszaru poszukiwań, zatrzymują się, przesuwiają w bok i kontynuują poszukiwania w drugą stronę, granicząc z poprzednim obszarem poszukiwań – w ten sposób szerokimi pasami przeszukują cały obszar mapy.

#### 2.4.2 Algorytm bug 2

Algorytm ten jest stosowany w przypadku napotkania przeszkody przez robota. Polega on na podążaniu wzdłuż ściany przeszkody do momentu osiągnięcia punktu leżącego za napotkaną ścianą, na przedłużeniu początkowej ścieżki ruchu. Jego działanie zostało przedstawione na rysunku 1.



Rysunek 1: Działanie algorytmu bug 2

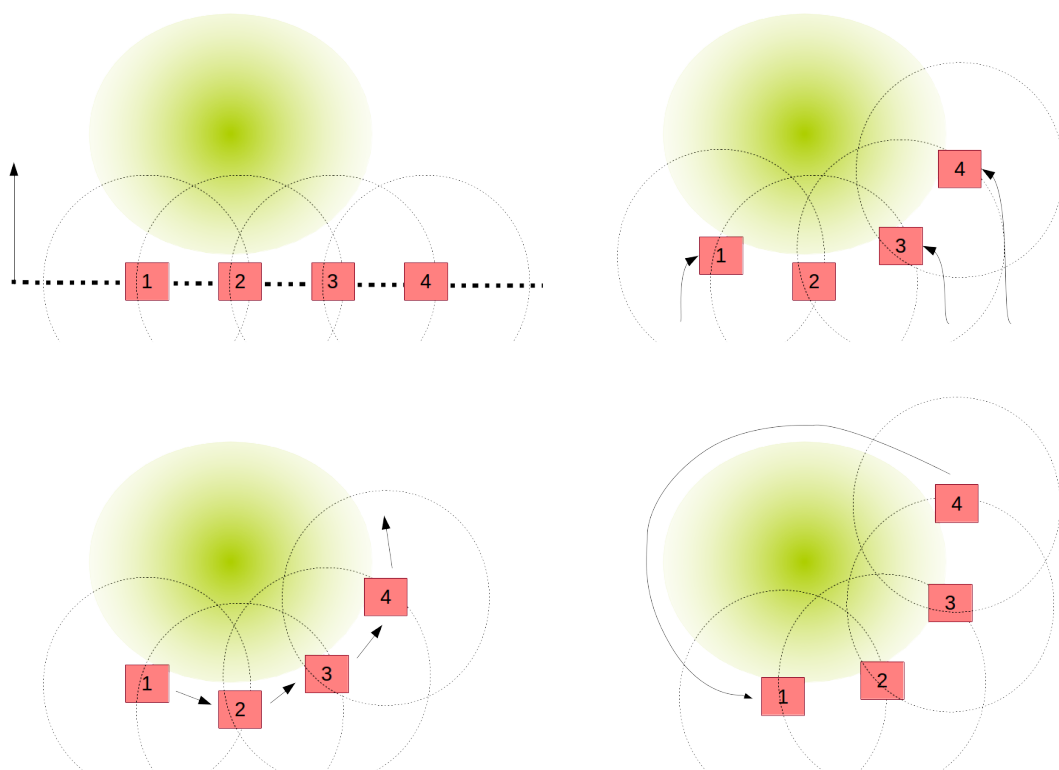
#### 2.4.3 Algorytm otaczania chmury

W momencie napotkania chmury przez któregokolwiek robota, zatrzymuje się on (po osiągnięciu natężenia toksyn określonego arbitralnie). Pozostałe roboty jadą dalej, po łuku – tak, aby nie przerwać połączenia z siecią. Gdy napotkają określone stężenie chmury, zatrzymują się. Gdy wszystkie roboty zatrzymały się, zwracają się w jedną stronę i zaczynają okrążyć chmurę, utrzymując się na granicy stężenia oraz zachowując spójność sieci. Jego działanie przedstawiono na rysunku 2.

## 3 Realizacja

### 3.1 Komunikacja

Komunikacja przebiega na zasadzie ogłaszania aktualnych informacji na swój temat przez roboty (oraz chmurę). Informacje przesyłane są za pomocą struktury JSON do gniazd, przypisywanych każdemu modułowi osobno. Wiadomości opatrzone są nagłówkiem, pozwalającym sklasyfikować komunikat. Dane tworzone są w postaci usystematyzowanych słowników, których struktura została ogólnie ustalona. Komunikaty są wysyłane i odbierane asynchronicznie.



Rysunek 2: Działanie algorytmu otaczania chmury. Robot 2 napotyka chmurę, pozostałe roboty zbliżają się do chmury aż do wartości granicznej, następnie cała sieć zaczyna krążyć wokół chmury, monitorując jej kształt i położenie.

### 3.1.1 Roboty

Poniżej przedstawiono przykład

```
def broadcast_info(self):
    #print("Sending info!" + str(self.number))
    message = str(self.robot._name)+str(self.substage)
    message = dict()
    message["robot"] = self.robot.name
    message["stage"] = self.stage
    message["substage"] = self.substage
    message["type"] = "progress"
    self.robot.commutron.broadcast(json.dumps(message))
```

### 3.1.2 Chmura

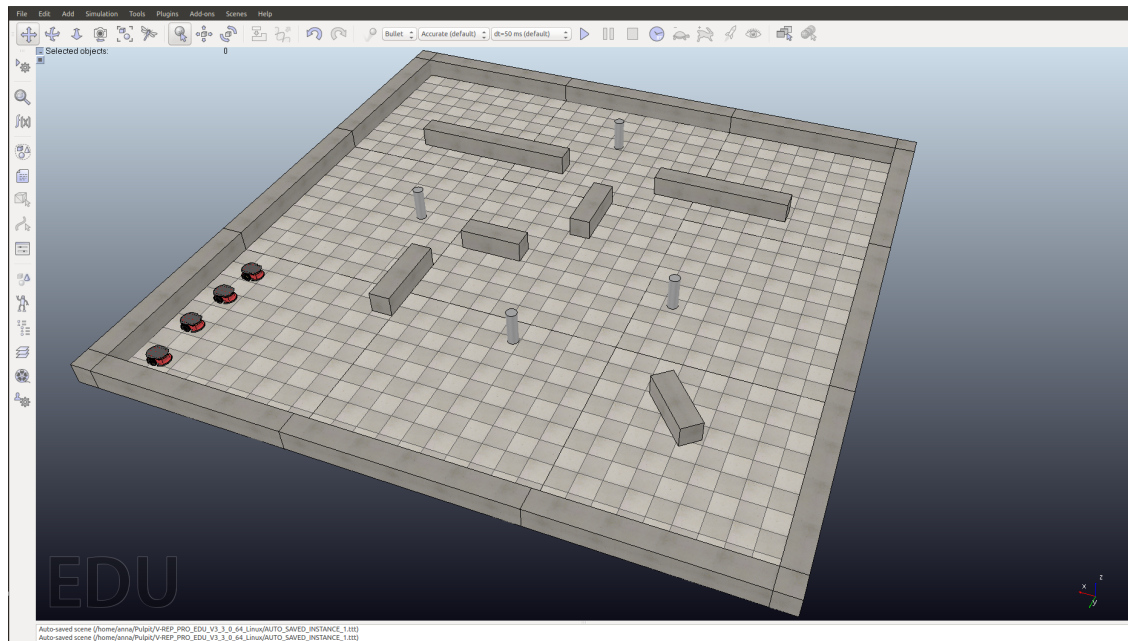
## 3.2 Roboty

Wykorzystywane w realizacji zadania roboty Pioneer 3-DX wyposażone w ultradźwiękowe czujniki odległości - będą one zapewnione przez symulator V-Rep. Posiadają one napęd różnicowy, pozwalający na dużą swobodę poruszania się po symulowanej przestrzeni. Są przystosowane do rozbudowy o dodatkowe urządzenia

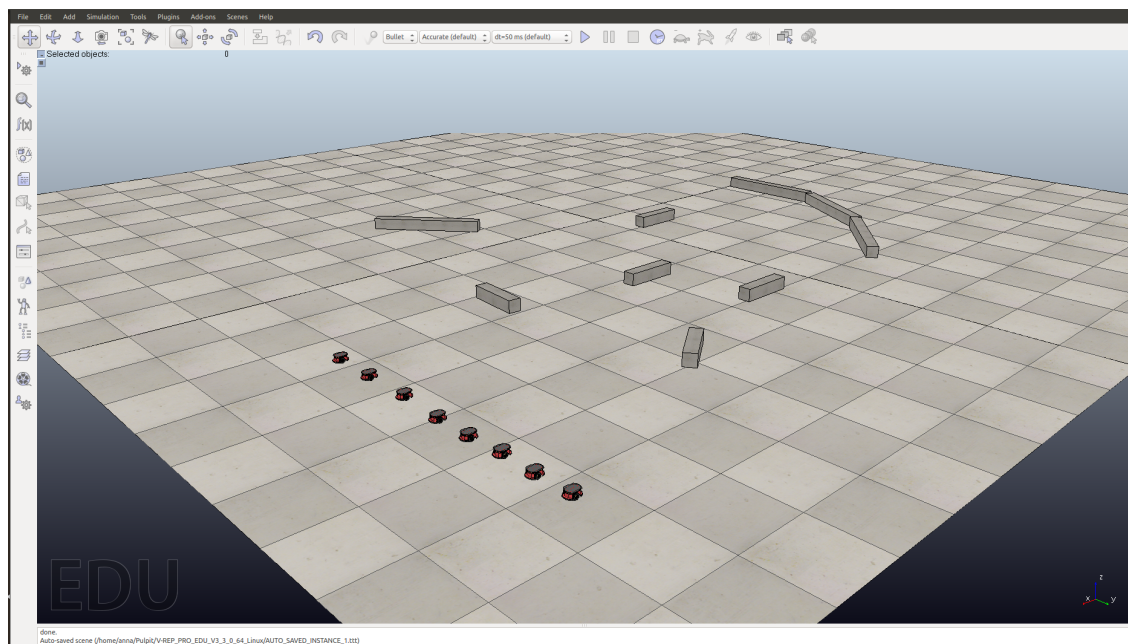
– dzięki czemu mogły by być łatwo dostosowane do realizacji zadania lokalizacji chmury. Dodatkowo, czujniki stężenia toksycznych substancji będą realizowane w skryptach Pythona, ułatwiając jej symulację. Informacje o jej położeniu i kształcie będą przekazywane do Vrepa, gdzie nastąpi ich wizualizacja.



### 3.3 Przykładowe mapy środowiska



(a) Mały teren



(b) Duży teren

Rysunek 3: Przykładowe mapy środowiska