

Lokalizacja robota w hangarze

Michał Kozłowski

13-14 Marca 2022

Spis treści

1	Problem	2
2	Możliwe modele położenia	2
3	Możliwe rozwiązania systemu lokalizacji	2
3.1	Napędowo-ruchowy system lokalizacyjny	2
3.2	Wizualny system lokalizacyjny	3
3.3	Zewnętrzny system lokalizacyjny	3
4	Wybrana kombinacja systemów	4
5	Rozwiązanie	4
5.1	Osprzętowanie	4
5.1.1	Relacje i połączenia pomiędzy częściami fizycznymi	4
5.2	Software	5
5.2.1	Wykorzystanie hardware'u przez software	5
5.2.2	Struktury danych	5
5.2.3	Algorytmy	6
5.2.4	Przykładowa struktura pętli programu	6
5.2.5	Wpływ użytego modelu położenia na wyszukiwanie trasy	6

1 Problem

”Zaproponuj sposób lokalizowania się robota w hangarze”

Problem powyższego zadania polega na znalezieniu możliwie najlepszego systemu lokalizowania się robota poruszającego się po zamkniętym hangarze.

Ze względu na to, że w treści zadania nie zostały zawarte kryteria oceny systemu postaram się rozważyć jak najwięcej opcji skupiających się na optymalizacji w różnych kierunkach (np. niski koszt, niska złożoność, niezawodność) oraz na ich możliwych kombinacjach i postaram się wybrać system, który według mnie sprawdził by się najlepiej.

2 Możliwe modele położenia

By możliwa była lokalizacja robota w hangarze potrzebny będzie sposób opisanie w którym miejscu hangaru się znajduje.

1. Unikalnym identyfikatorem krawędzi/połączenia - ID (nie reprezentuje krawędzi dwoma punktami, ze względu na to, że w innych hangarach pomiędzy dwoma punktami może istnieć więcej niż jedno połączenie co uniemożliwiłoby taką reprezentację)
2. Odległością od początku krawędzi/połączenia (w tym celu krawędzie grafów są jednokierunkowe, by można było jednoznacznie oznaczyć jego położenie)

Przykładowe położenie może mieć postać: `std::pair<Node*, double>(nodes[45], 89.78)`, lub inaczej `node:46, distance:89.78m`

Zalety powyższego systemu reprezentacji lokalizacji:

- Odczytanie lokalizacji zapisanej w danym formacie jest jednoznaczne

Wady powyższego systemu:

- Zapisanie lokalizacji znajdującej się na skrzyżowaniu połączeń nie jest jednoznaczne, będą istniały dla każdego skrzyżowania przynajmniej dwa sposoby zapisania tej lokalizacji (np. dla skrzyżowania do z dwoma krawędziami: A, B wychodzącymi ze skrzyżowania można zapisać lokalizację jako `node:A, dist:0m` lub `node:B, dist:0m`)

Powyższy model może zostać rozszerzony np. o odchylenie od krawędzi (model uzględnia tylko odległość pomiędzy skrzyżowaniami w linii prostej). Odchylenie takie może być zapisane jedną liczbą reprezentującą odchylenie robota od linii prostej w prawo, wartości ujemne będą oznaczały odchylenie w lewo, a zerowe jego brak.

drugą opcją rozszerzenia powyższego modelu jest dodanie kąta odchylenia od osi krawędzi.

Drugą opcją reprezentacji lokalizacji robota będzie określenie jego położenia w hangarze za pomocą trzech wartości: x, y, θ - odpowiednio odległość od krawędzi dolnej, odległość od krawędzi lewej, kąt pomiędzy kierunkiem robota a krawędzią dolną hangaru.

Ten model będzie charakteryzował się dokładniejszym i prostszym lokalizowaniem robota. Zalecane będzie zamienienie algorytmu wyszukującego trasę z *algorytmu Dijkstry* na wariant algorytmu **A***

3 Możliwe rozwiązania systemu lokalizacji

3.1 Napędowo-ruchowy system lokalizacyjny

Napędowo-ruchowy system lokalizacyjny określałby położenie pojazdu w jednym z powyższych modeli na podstawie danych z czujników w systemie ruchowym robota, np. jeśli systemem poruszania się robota będą koła, dane będą czytywane z czujników na osiach napędowych i osiach sterowania. Na podstawie tych danych możliwe będzie cykliczne oszacowywanie pozycji robota.

Zalety powyższego systemu:

- Niski koszt oraz złożoność systemu
- Niskie zapotrzebowanie mocy obliczeniowej

Wady powyższego systemu:

- System ten jest w stanie tylko oszacować przybliżoną lokalizację
- Obliczenie aktualnej pozycji opiera się na poprzedniej obliczonej lokalizacji, przez co każda niedokładność w obliczeniach i/lub pomiarach będzie przenosiła się na kolejne obliczenia - błąd będzie rosł liniowo jeśli system będzie polegał tylko na własnych danych.

Możliwe użycia systemu:

- System może zostać użyty w połączeniu z jednym z innych systemów, ze względu na rosnący błąd przy korzystaniu tylko z własnych obliczeń. Drugi system minimalizowałby błąd do stałego ze względu na ciągle dostarczanie niezależnych danych.

Wymagane komponenty:

- Czujniki na elementach odpowiadających za poruszanie się robota
- Jednostka obliczeniowa

3.2 Wizualny system lokalizacyjny

Wizualny system lokalizacyjny bazowałby na przetwarzaniu danych z kamer zamontowanych na robocie. Najlepszym przypadkiem użycia tego systemu jest praca robota w zamkniętym pomieszczeniu ze specjalnymi oznaczeniami na ścianach (bądź innych ograniczeniach) pozwalających na dokładną lokalizację robota.

Oznaczenia na ścianach mogą mieć postać unikalnych figur geometrycznych (np. kodów QR). Robot posiadałby zapisane informacje o położeniu każdego takiego oznaczenia i na podstawie wykrycia go będzie w stanie określić swoją pozycję.

Do wykrywania oznaczeń może zostać użyty system opisany w pracy "Rapid Object Detection using a Boosted Cascade of Simple Features" - Viola-Jones. Pozwala on na bardzo szybkie rozpoznawanie schematów między innymi w bitmapach.

Zalety powyższego systemu:

- Bardzo dokładna lokalizacja

Wady powyższego systemu:

- Potrzebne oznaczenia na ścianach co ogranicza możliwości zastosowania
- Wymagane specjalne komponenty
- Wyższe wymagania mocy obliczeniowej w porównaniu do poprzedniego systemu

Wymagane komponenty:

- Kamery wideo
- Jednostka obliczeniowa

3.3 Zewnętrzny system lokalizacyjny

Zewnętrzny system lokalizacyjny polegałby na określaniu pozycji robota korzystając z zewnętrznych czujników.

Można to osiągnąć korzystając między innymi z systemu przedstawionego w punkcie 3.2 z kilkoma modyfikacjami. Kamery/czujniki musiałby zostać ustawione nad przestrzenią w której będzie poruszał się robot, a znaczniki byłyby zlokalizowane w widocznym dla kamer miejscu na robocie. Lokalizacja będzie przekazywana do robota z systemu zewnętrznego z pomocą połączenia bezprzewodowego.

Rozwiązanie to pozwala na monitorowanie wielu robotów w systemie jednocześnie co pozwala na koordynację grupy i skorzystanie z algorytmu pozwalającego na optymalizację przemieszczania się w grupie.

W tym systemie zalecałbym użycie drugiego systemu reprezentacji lokalizacji.

Zalety powyższego systemu:

- Wysoka dokładność lokalizacji
- Możliwość obsługi wielu robotów jednocześnie i ich koordynacji

Wady powyższego systemu:

- Wysoki koszt
- Wysoka złożoność systemu

Potrzebne komponenty:

- Oznaczenia na robocie/robotach
- Jednostka łączności w robocie/robotach
- Kamery w hangarze
- Jednostki łączności w hangarze

4 Wybrana kombinacja systemów

W tej sekcji postaram się przedstawić system lokalizacyjny, zoptymalizowany pod względem dokładności oraz złożoności.

System reprezentacji lokalizacji: system nr.2 - reprezentacja na płaszczyźnie

Systemy lokalizacji robota:

- Wizualny system lokalizacyjny
- Napędowo-ruchowy system lokalizacyjny

Zaprojektowany przeze mnie system będzie opierał się na dwóch wyżej wymienionych systemach lokalizacyjnych. Napędowo-ruchowy system lokalizacyjny będzie wspierany przez wizualny system lokalizacyjny w celu minimalizacji błędów określania lokalizacji.

System ten może zostać wyposażony w awaryjne czujniki odległości by uniknąć zderzenia z innymi obiektami zarówno należącymi do struktury hangaru jak i obiektami w niej się nie zawierającym.

5 Rozwiązanie

Jako rozwiązanie proponuję rozwiązanie w którym hangar będzie reprezentowany kombinacją dwóch wyżej zaprezentowanych systemów, a lokalizacja będzie określana za pomocą kombinacji systemów: Wizualny oraz Napędowo-Ruchowy

5.1 Osprzętowanie

W potrzebnym osprzętowaniu znajdują się:

- **Czujniki odległości**
3 - 4 sztuki zamontowane na froncie, lub 6 - 8 jeśli robot będzie mógł poruszać się wstecz
- **Kamery**
2 - 3 sztuki zamontowane w równych odległościach kątowych od siebie by pokryć jak największą powierzchnię
- **Jednostka sterująca**
Jednostka obliczeniowa odpowiadająca za odbieranie i przetwarzanie danych z czujników
- **Czujniki na systemie poruszania się**
Czujniki rejestrujące jakie ruchy wykonuje robot na podstawie ruchu części odpowiadających za przemieszczanie się robota

5.1.1 Relacje i połączenia pomiędzy częściami fizycznymi

Wszystkie wyżej wymienione komponenty będą połączone bezpośrednio z jednostką sterującą. Istnieje możliwość połączenia jednostki sterującej bezpośrednio do systemu hamulcowego z powodów bezpieczeństwa.

5.2 Software

Fragmenty software'u będą zaprezentowane w C++.

5.2.1 Wykorzystanie hardware'u przez software

- **Czujniki odległości**

Czujniki odległości będą wykorzystywane przez poboczny wątek odpowiadający za kontrolowanie odległości. W przypadku, w którym odległość do przeszkody przekroczy próg krytyczny zostanie aktywowany system hamulcowy.

- **Kamery**

Kamery będą wykorzystywane przez algorytm odpowiadający za określanie aktualnej pozycji w hangarze.

- **Czujniki systemu poruszania się**

Czujniki systemu poruszania się będą wykorzystywane przez ten sam algorytm co Kamery, który określa aktualną pozycję robota.

- **Jednostka sterująca**

Jednostka sterująca będzie centralną częścią systemu, na której wykonywane będą wszystkie obliczenia.

5.2.2 Struktury danych

- **Schemat hangaru**

Schemat hangaru będzie przechowywany jako struktura składająca się z poniższych elementów:

- **Lista skrzyżowań hangaru - Crossings:**
Będą w niej przechowywane skruktury reprezentujące skrzyżowania (przedstawione poniżej)
- **Struktura obiektów ograniczających - Restr:**
Struktura określająca w jakich miejscach hangaru znajdują się nieprzekraczalne struktury i inne miejsca przez które robot nie może się poruszać. Może zostać zaimplementowana jako bitmapa, lub zbiór figur geometrycznych w zależności od potrzeb.
- **Lista połączeń - Connections:**
Lista przechowująca informacje o połączeniach, która pozwoli na szybsze wyszukiwanie najtańszej trasy.

Skrzyżowanie hangaru będzie zapisane za pomocą poniższych elementów:

- **ID skrzyżowania - ID:**
Opcjonalny element pomagający identyfikację skrzyżowania
- **Lokalizacja - Location:**
Element reprezentujący w jakim miejscu w hangarze znajduje się skrzyżowanie - para liczb zmiennoprzecinkowych.
- **Połączenia - Links:**
Struktura reprezentująca połączenia z innymi skrzyżowaniami

Symbole ścienne będą zapisywane za pomocą ich modelu używanego w Algorytmie *Viola-Jones* oraz ich położenia w hangarze.

5.2.3 Algorytmy

W programie sterującym będą wykorzystywane poniższe algorytmy:

- **Viola-Jones algorithm**
Wykorzystywany do wyszukiwania i rozpoznawania lokalizacyjnych symboli ściennych.
- **Dijkstra algorithm**
Wykorzystywany do wyszukiwania najkrótszej trasy do celu.

5.2.4 Przykładowa struktura pętli programu

```
void compute_location(Location* prev, std::time_t dt, int* iteration){
    std::time_t cycle_s_t = std::time(0);
    if(iteration % EXTERNAL_UPDATE_RATE){
        prev = get_location_from_system_1();
    } else {
        prev = get_location_from_system_2();
    }
    std::time_t cycle_d_t = std::time(0) - cycle_s_t;
    cycle_balancer(cycle_d_t);
    iteration++;
}
```

Powyższa funkcja wywoływana cyklicznie będzie szacowała dane na podstawie odczytów z układu ruchowego co każdą iterację i co kilka iteracji będzie precyzowała ją za pomocą wizualnego systemu lokalizacji. W ten sposób osiągniemy balans pomiędzy dokładnością a wymaganą mocą obliczeniową.

5.2.5 Wpływ użytego modelu położenia na wyszukiwanie trasy

Zaprezentowany powyżej model położenia robota w hangarze dzięki temu, że jest rozszerzonym grafem ważonym umożliwia wyszukiwanie trasy przy bardzo małym nakładzie mocy obliczeniowej. Wyszukiwanie to można wykonać za pomocą algorytmu Dijkstry lub innego algorytmu wyszukiwania trasy dla grafów nie-kierunkowych z wagami nieujemnymi. Wystarczy dodać do grafu nowe skrzyżowanie dzielące połączenie, na którym aktualnie znajduje się robot. Wagi nowych połączeń można obliczyć na podstawie zapisanych w modelu położenia współrzędnych skrzyżowań oraz współrzędnych robota uzyskanych z zaprezentowanego wyżej programu.