

**Politechnika Warszawska**  
**Wydział Elektroniki i Technik Informacyjnych**  
Instytut Automatyki i Informatyki Stosowanej

## **Projekt SST i MISK**

System zarządzania magazynem wspomagany  
robotami mobilnymi

Raport pierwszej części realizacji projektu

**Zespół projektowy:**

Wiktor Gerstenstein  
Marek Dawidiuk  
Paweł Tymiński  
Robert Nebeluk

**Prowadzący:**

dr inż. Michał Karpowicz

Warszawa 2017

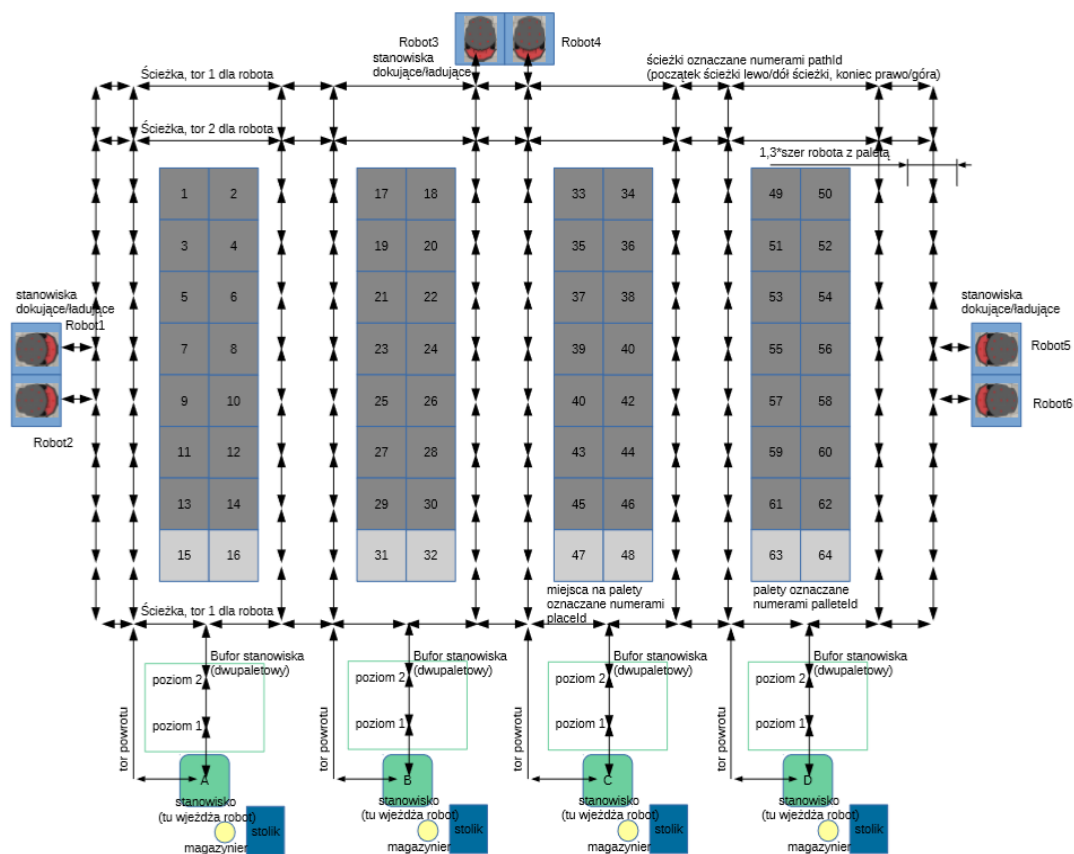
## 1. Pierwszy etap realizacji zadania projektowego

Prace nad systemem zarządzania magazynem przy wykorzystaniu robotów mobilnych rozpoczęto od przemyślenia i sformułowania koncepcji projektu jaki i zdefiniowania założeń projektowych. Niezbędne było opracowanie konspektów i opisów każdego z części projektu. Następnie przydzielone zostały zadania i przystąpiono do realizacji.

W tej części utworzono strukturę magazynu w środowisku V-REP. Do realizacji zadań w magazynie wybrano roboty o nazwie „Pioneer” z biblioteki symulatora. Następnie przystąpiono do nawiązania komunikacji z wskazanymi jednostkami za pomocą skryptów w języku Python. Po wykonaniu pierwszych testów rozpoczęto pracę nad docelowym algorytmem pozwalającym na wysterowanie robotów tak, aby wykonywały powierzone im zadania zgodnie z założeniami projektowymi. W kolejnych rozdziałach raportu zostały opisane wyniki pracy wykonane w ramach tej części projektu.

## 2. Projekt magazynu

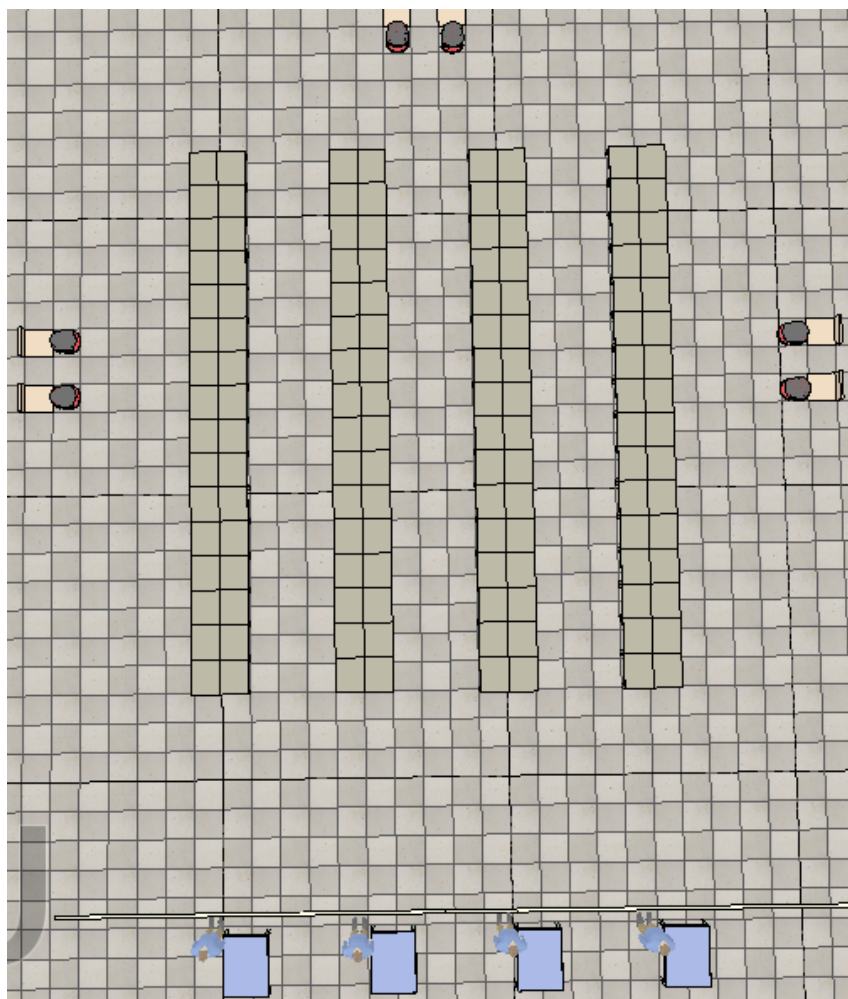
Magazyn składa się z 6 stanowisk ładujących, gdzie dokują roboty, 128 miejsc na palety z towarem oraz 4 stanowisk obsługi palet z 2 paletowymi buforami przy każdym stanowisku. Każdy z robotów ma za zadanie dojechać do wskazanego miejsca, gdzie obecnie znajduje się paleta z załadowanym towarem, pobrania jej, dostarczenia do określonego stanowiska i powrotu do stanowiska dokującego lub rozpoczęcia wykonywania kolejnego zadania. Na stanowiskach obsługi mogą być wykonane operacje: pobierz, połóż i sprawdź towar przywieziony przez robota,. Robot może się poruszać po wskazanych ścieżkach i powinien wybrać tę, która zapewni mu najkrótszy czas realizacji celu. Na rysunku poniżej pokazana jest omówiona struktura magazynu.



Ilustracja 1: Model koncepcyjny magazynu

Ilość miejsc składowujących została zwiększona, aby osiągnąć lepsze wyniki optymalizacji w wyniku zwiększonej liczby ścieżek oraz odległości od poszczególnych stanowisk.

Scena magazynu w symulatorze V-REP została utworzona w oparciu o przygotowany wcześniej model i jest przedstawiona poniżej.



*Ilustracja 2: Scena magazynu (widok z góry)*

Części magazynu takie jak stanowiska, palety zostały utworzone z wykorzystaniem oprogramowania Solidworks, a następnie dodane do sceny.

### 3. Implementacja niezbędnych klas

Na potrzeby pierwszej fazy projektu zostały utworzone następujące klasy:

- **klasa *Palette*** zawiera położenie początkowe palety, położenie aktualne palety oraz jej nazwę w symulatorze vrep. Metody tej klasy pozwalają na aktualizację położenia oraz otrzymanie jego wartości.
- **klasa *PaletteAction*** definiuje akcje jakie można wykonać w związku z paletą w magazynie. Są to: paleta załadowana na robota, przenoszona, na stanowisku obsługi, w buforze lub gotowa do wzięcia.
- **klasa *Path*** zawiera pozycję początku ścieżki oraz pozycję końca ścieżki, wyliczoną długość ścieżki.

- **klasa PathWay** definiuje drogę jaką porusza się robot zawiera pojedyncze ścieżki (listę klas *Path*).
- **klasa Place** zawiera aktualną pozycję palety oraz identyfikator ścieżki prowadzącej do tego miejsca.
- **klasa Station** zawiera bufor (obiekt klasy *StationBuffer*), pozycję stanowiska, identyfikator ścieżki prowadzącej do stanowiska od bufora.
- **klasa DockStation** definiuje stację ładującą dla robota, zawiera pozycję stacji dokującego, gdzie ładowany jest robot oraz identyfikator ścieżki prowadzącej do niej.
- **klasa StationBuffer** definiuje bufor magazynu, zawiera pozycję miejsc w buforze i ścieżki prowadzące do danej pozycji jak i aktualną ilość palet w buforze.
- **klasa MoveRobot** implementuje algorytm jazdy do przodu na określoną odległość, obrót robota w miejscu na określony kąt w stopniach, pozwala na ustawienie prędkości poruszania oraz pobranie aktualnej pozycji. Zawiera również mechanizm zliczania czasu pracy robota, który to będzie czynnikiem podlegającym optymalizacji.
- **klasa RobotInStorehouse** jest klasą następną w hierarchii. Definiuje poruszanie się robota po wybranej ścieżce i drodze. Pozwala na interakcję z obiektami w magazynie (np. paletami – przyłączenie/odłączenie). Pozwala na wkroczenie na ścieżkę, jazdę do końca ścieżki, przyczepienie i transport palety, odłożenie palety.
- **klasa MoveRobotsInStorehouse** jest klasą najwyższą w hierarchii, która pozwala wykonywać zadania wykonywane na magazynie przez roboty. Klasa pozwala na ustawianie prędkości poruszania się robota, pobranie palety z magazynu, przewiezienia palety do stanowiska obsługi lub odłożenie palety, a także powrót robota do stacji ładującej.
- **klasa Storehouse** przechowuje wszystkie informacje odnośnie ustawienia obiektów w magazynie.  
Przechowuje pozycje:
  - a. miejsc w magazynie dostępnych pod *placeId*,
  - b. aktualną pozycję palet dostępnych pod *paletteId*,
  - c. pozycje stanowisk dostępnych pod *stationName*
  - d. pozycje stacji dokujących dostępnych pod *stationName*,
  - e. tablice ścieżek *Paths* przechowująca wszystkie dostępne ścieżki dostępnych pod *pathId*

#### 4. Opis algorytmu poruszania się po wytyczonych ścieżkach we współrzędnych globalnych

Wszystkie dostępne ścieżki robotów zostały zdefiniowane i otrzymały unikalny identyfikator. Algorytm będzie wybierał na bieżąco najbardziej optymalne ścieżki odwołując się do zdefiniowanego numeru identyfikator i budował drogę do celu. Ze względu na to, że roboty będą pracowały jednocześnie wykonując przypisane im zadania magazynowe to algorytm ma zapewnić: minimalizację czasu wykonywania tych zadań w tym minimalizację przestojów w buforze stanowiska innymi słowy wybranie najmniej obciążonego stanowiska lub wykonanie innego zadania w tym czasie, kompaktowanie miejsca w magazynie na podstawie prognozowanych zamówień, wykrywanie kolizji z przenoszonymi paletami przed wykonaniem decydującego ruchu co w rezultacie uzyskujemy poruszanie się robotów wytyczonymi antykolizyjnymi ścieżkami, liczenie opóźnień realizowanych zadań.

W celu minimalizacji czasu przejazdu algorytm będzie uwzględniał drogę ze stacji do palety i wybranego stanowiska. Wybranie odpowiedniego stanowiska będzie uzależnione od obciążenia, czyli ilości zadań w kolejce stanowiska a także od drogi do stanowiska oraz możliwości

zakończenia obsługi palety przez stanowisko, do którego może być kierowana paleta w czasie dojazdu do stanowiska.

## **5. Przygotowanie do sterowanie robotem *pioneer*, modyfikacja skryptu robota**

W celu poprawy poruszania robotem, tak aby jak najdokładniej osiągnął określonej pozycji wykorzystano możliwość sterowania pozycją za pomocą samodzielnie zdefiniowanego regulatora PD bezpośrednio w symulatorze. Pozwoli to na osiągnięcie lepszych efektów sterowania robotem niż w przypadku sterowania z poziomu skryptu w Pythonie. Musiałby on oczekiwać na odpowiedź od symulatora, a przez to opóźnienia powodował zbyt późne zatrzymanie robota. Podczas realizacji zastosowano zmodyfikowany regulator PD, którego działanie polega na zerowaniu wartości części proporcjonalnej, jeśli wartość uchybu jest mniejsza niż 0.01 radiana. Bez tej modyfikacji silniki po osiągnięciu zadanych pozycji powoli obracały się w przypadkowym kierunku.

## **6. Organizacja pracy i zarządzanie projektem**

Ważnym zadaniem podczas rozpoczęcia pracy nad nowym projektem jest poznanie umiejętności zespołu oraz uzupełnienie ewentualnych braków przez członków zespołu pod kątem powierzanych im zadań. Do tego celu zostały przygotowane instrukcje i znalezione odpowiednie poradniki internetowe.

Kolejnym ważnym zadaniem jest podział obowiązków i wyznaczenie osób odpowiedzialnych za każdą z części projektu. Projekt zdekomponowano na mniejsze zadania i przydzielono odpowiednim osobą. Do tego celu posłużyła aplikacja webowe Trello oparta na paradygmacie Kanban należącym do technik Agile. Pozwala w łatwy sposób rozdzielać zadania między członków zespołu.

Trudnością jaką napotyka się przy współpracy zdalnej jest komunikacja. Pomocne w rozwiązaniu tej kwestii okazało się narzędzie Slack, które jest platforma stworzona specjalnie do komunikacji zespołowej. Ma zaawansowany system powiadomień, który pozwala na monitorowanie kanałów i grup.

Istotną kwestią jest przechowywanie efektów pracy zespołu, tak aby każdy z członków miał łatwy dostęp do stworzonych zasobów. Niezbędne w tym celu było stworzenie repozytorium. Zdecydowano się na skorzystanie z GitHuba, który jest prostym w użyciu i popularnym narzędziem do realizacji projektów, odpowiednim dla potrzeb naszego zespołu.

## **7. Podsumowanie**

W ramach pierwszej części realizacji projektu zostało przygotowane środowisko symulacyjne pozwalające na wizualizację zaimplementowanego algorytmu sterowania robotami mobilnymi. Stworzona została struktura klasowa programu z dokładnym wyznaczeniem zadań dla każdej z klas. Zaproponowany został algorytm szukania ścieżek przez roboty przy wykorzystaniu zmiennych globalnych. Na tym etapie już natknięto się na problemy związane ze sterowaniem dostępnych jednostek i zaproponowano rozwiązanie w postaci zmodyfikowanego regulatora PD. Wykonana praca pozwala na przystąpienie do kolejnego kroku realizacji projektu jakim jest wdrożenie algorytmu optymalizacyjnego dla opisanego systemu zarządzania magazynem.