

# Sprawozdanie

## Roboty przemysłowe

### Igor Cena

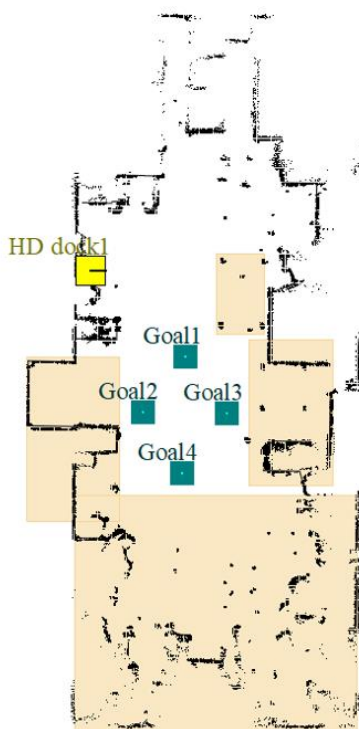
### IME - III rok

Sprawozdanie powinno zawierać opis wykonanych prac, uzyskany wykres z trasą robota (z integracji, z enkoderów oraz z czujnika laserowego), kod programu, zestawienie wyników dokładności i powtarzalności lokalizacji oraz wnioski z ćwiczenia.

#### 1. Opis wykonanych prac

Na początku zeskanowaliśmy i utworzyliśmy mapę poprzez przejechanie za pomocą manualnego sterowania joystickiem i umożliwienie „zobaczenia” przez robota całego pomieszczenia. W kolejnym kroku na wygenerowaną mapę 2D dodaliśmy obszary zakazane i cele, które musiał osiągnąć robot. Następnie stworzyliśmy makro, które uwzględniło przejazd robota przez wszystkie dodane cele oraz zbudowaliśmy trasę dodając w niej również komendy wait pomiędzy każdorazowym wykonaniem jednego okrążenia. Wgraliśmy trasę do robota i po wykonaniu przez niego programu otrzymaliśmy plik .csv zawierający dane takie jak pozycja i orientacja estymowane na podstawie odczytów z enkoderów i czujników laserowych robota z całej trasy. Później wyodrębniliśmy otrzymane dane na trzy rodzaje: uzyskane na podstawie samych enkoderów, uzyskane na podstawie samego czujnika LiDAR oraz zintegrowaną pozycję i wystaliśmy je do osoby prowadzącej zajęcia, aby poprawiła Ona orientację danych z enkoderów. Po uzyskaniu nowych danych wgraliśmy je w miejsce starych i wyświetliliśmy zarejestrowane ścieżki na jednym wykresie. Ostatnim zadaniem było obliczenie dokładności, w których jako referencyjną pozycję przyjęliśmy tą uzyskaną z zintegrowanych czujników oraz powtarzalność uzyskiwania punktów, w których robot czekał sekundę po każdorazowym wykonaniu okrążenia.

#### 2. Wygenerowana mapa 2D



### 3. Program trasy wraz z pokazaną zawartością makr



### 4. Kod programu

```
import os
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import matplotlib
matplotlib.use('TkAgg')

# Ścieżka do folderu z plikiem csv
file_folder = "C:\\Users\\Igor\\Desktop\\studia\\jkoszyk omron pycharm"
# Nazwa pliku
file_name1 = "dataLog_Omron-LD_241119_153731.csv"
file_path1 = os.path.join(file_folder, file_name1)

# Odczytanie pliku
df1 = pd.read_csv(file_path1)

# Ścieżka do folderu z plikiem csv
file_folder1 = "C:\\Users\\Igor\\Desktop\\studia\\jkoszyk omron pycharm"
# Nazwa pliku
file_name2 = "Dane_z_encodera1.csv"
file_path2 = os.path.join(file_folder1, file_name2)

# Odczytanie pliku
df2 = pd.read_csv(file_path2)
# Wyświetlenie nazw kolumn
print(df2.columns)

# Wyświetlenie wymiarów datasetu
print(df1.shape)

# Wyświetlenie 5 pierwszych linijek datasetu
print(df1.head())
```

```

# Wyświetlenie nazw kolumn
print(df1.columns)

# Wybranie wartości robota z przejazdu Trasy 1
df1 = df1[df1["ModeName (string)"] == "Patrolling route DASJ_3 once"]

# Wyświetlenie wymiarów datasetu
print(df1.shape)

x_robot = np.array(df1['RobotX (mm)']).reshape((-1, 1))
y_robot = np.array(df1['RobotY (mm)']).reshape((-1, 1))
th_robot = np.array(df1['RobotTh (degrees)']).reshape((-1, 1))

arr_xy_robot=np.hstack((x_robot,y_robot))
df_robot=pd.DataFrame(arr_xy_robot)
df_robot.to_csv('Dane z robota')

x_encoder = np.array(df2['//X']).reshape((-1, 1))
y_encoder = np.array(df2['Y']).reshape((-1, 1))
th_encoder = np.array(df2['Z']).reshape((-1, 1))

arr_xy_encoder=np.hstack((x_encoder,y_encoder))
df_encoder=pd.DataFrame(arr_xy_encoder)
df_encoder.to_csv('Dane z encodera')

x_laser = np.array(df1['LaserLocalization_X (mm)']).reshape((-1, 1))
y_laser = np.array(df1['LaserLocalization_Y (mm)']).reshape((-1, 1))
th_laser = np.array(df1['LaserLocalization_Th (degrees)']).reshape((-1, 1))

arr_xy_laser=np.hstack((x_laser,y_laser))
df_laser=pd.DataFrame(arr_xy_laser)
df_laser.to_csv('Dane z lasera')

# Wyodrębnienie danych, gdzie robot wykonywał polecenie "Wait"
waiting_data = df1[df1["ModeStatus (string)"] == "Waiting"]
x_wait=np.array(waiting_data['RobotX (mm)']).reshape((-1, 1))
y_wait=np.array(waiting_data['RobotY (mm)']).reshape((-1, 1))

# Wyświetlenie danych na jednym wykresie
plt.ion
plt.title('Ruch robota i punkty "Wait"')
plt.xlabel('x [mm]')
plt.ylabel('y [mm]')
plt.plot(x_encoder, y_encoder, c="g", label="Ścieżka encodera")
plt.plot(x_laser, y_laser, c="b", label="Ścieżka lasera")
plt.plot(x_robot, y_robot, c="y", label="Ścieżka robota")
plt.plot(x_wait, y_wait, 'ro', label="Punkty 'Wait'")
plt.legend()
plt.show()

# Obliczenie dokładności między trasą zarejestrowaną przez robota, a trasą
zarejestrowaną przez encoder
distances_with_encoder = np.sqrt((x_robot - x_encoder)**2 + (y_robot -
y_encoder)**2)
accuracy_with_encoder = np.mean(distances_with_encoder)
round(accuracy_with_encoder, 2)
print("Średnia dokładność porównując z trasą encodera:",
round(accuracy_with_encoder, 3),"mm")
distances_with_laser = np.sqrt((x_robot - x_laser)**2 + (y_robot -

```

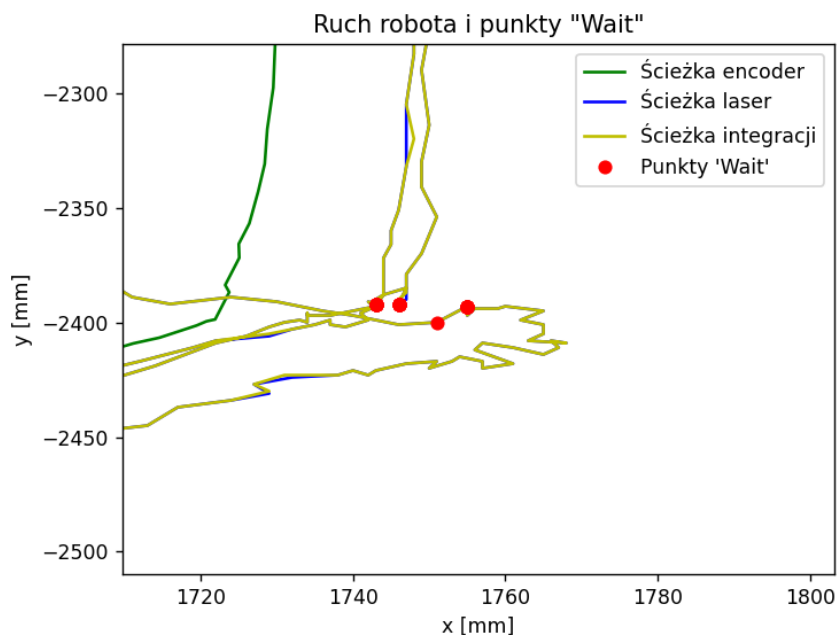
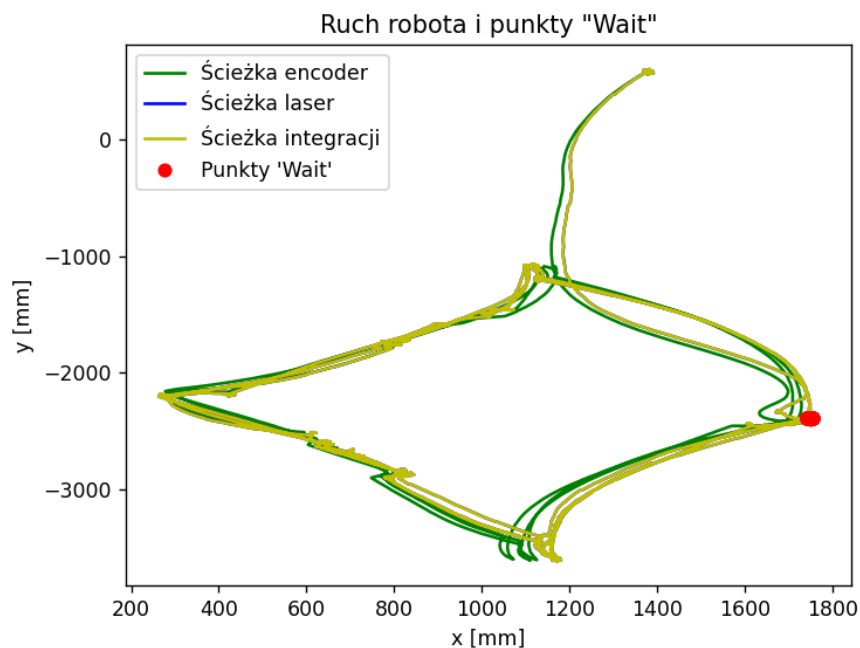
```

y_laser)**2)
accuracy_with_laser = np.mean(distances_with_laser)
round(accuracy_with_laser, 2)
print("Średnia dokładność porównując z trasą lasera:
",round(accuracy_with_laser, 3),"mm")

# Obliczenie powtarzalności
# Obliczam współrzędne środka ciężkości między punktami wait
center_x = np.mean(x_wait)
center_y = np.mean(y_wait)
# Obliczam odległości pomiędzy punktami a środkiem ciężkości
distances = np.sqrt((x_wait - center_x)**2 + (y_wait - center_y)**2)
repeatability = np.std(distances)
print("Uzyskana powtarzalność: ",round(repeatability,3),"mm")

```

## 5. Wykres trasy robota i punkty „Wait”



## 6. Zestawienie wyników dokładności i powtarzalności lokalizacji

```
Średnia dokładność porównując z trasą encodera: 42.504 mm  
Średnia dokładność porównując z trasą lasera: 0.035 mm  
Uzyskana powtarzalność: 2.169 mm
```

## 7. Wnioski

Po wynikach możemy wywnioskować, że czujniki laserowe są o wiele bardziej dokładne w porównaniu do enkoderów, jeżeli jako pozycję referencyjną uznajemy pozycję uzyskaną z zintegrowanych czujników. Widać to również na wykresie - ścieżka uzyskana z czujników laserowych pokrywa się niemal idealnie z ścieżką uzyskaną z czujników zintegrowanych, gdzie z drugiej strony ścieżka uzyskana z enkoderów w wielu odcinkach odstaje. Uzyskana powtarzalność wynosi jedynie ok. 2 mm, co jest dobrym wynikiem i oznacza, że robot trafia za każdym razem w prawie to samo miejsce.