

Katedra Automatyki i Robotyki Podstawy Robotyki z Kinematyką Ćwiczenie nr 5 – Robot mobilny Khepera III		
L.p.	Imię i nazwisko	Data i godzina
1.	Jakub Szczypek	21.04.22r. godz. 11:15 – 12:45 (czwartek)
2.	Agata Słonka	
3.	Aleksandra Stachniak	

1. Cel ćwiczenia

Celem zajęć jest zapoznanie się z budową i możliwościami robota Khepera III oraz stworzenie aplikacji umożliwiających realizację prostych trajektorii i wykrywania przeszkód przez tego robota.

2. Przebieg ćwiczenia

Laboratorium rozpoczęło od połączenia się z robotem za pomocą komendy „ref=kopen('COM3')” napisanej w command window w programie Matlab. Następnie zbadano zachowanie robota zadając mu podstawowe rozkazy. Po sprawdzeniu zachowania się robota napisaliśmy skrypt umożliwiający poruszanie się robotowi po zamkniętym prostokącie. W niekończącej się pętli, z wykorzystaniem instrukcji warunkowych, instrukcji kSetSpeed (nadawanie prędkości), kStop (zatrzymanie robota) oraz kReadPos (odczyt współrzędnych), nadawano robotowi rozkaz poruszania się prosto na określoną odległość. Następnie w kolejnej pętli nakazywano robotowi obrót i powyższe czynności powtarzano aż robot przejechał drogę po prostokącie. Poniżej przedstawiam stworzony przez nas skrypt.

```

1      a = kReadPos(ref);
2      kSetSpeed(ref,10000,10000)
3
4      while 1
5          b = kReadPos(ref);
6
7
8          if (b(1) >= a(1) + 8000) && (b(2) >= a(2) + 8000)
9              kStop(ref)
10             break
11         end
12     end
13
14
15
16     a = kReadPos(ref);
17     kSetSpeed(ref,5000,-5000)
18
19     while 1
20         b = kReadPos(ref);|
21
22         if (b(1) >= a(1) + 1040) && (b(2) <= a(2) - 1360)
23             kStop(ref)
24             break
25         end
26     end
27
28
29     a = kReadPos(ref);
30     kSetSpeed(ref,10000,10000)
31

```

```

32 while 1
33     b = kReadPos(ref);
34
35
36     if (b(1) >= a(1) + 5000) && (b(2) >= a(2) + 5000)
37         kStop(ref)
38         break
39     end
40 end
41
42 a = kReadPos(ref);
43 kSetSpeed(ref,5000,-5000)
44
45 while 1
46     b = kReadPos(ref);
47
48     if (b(1) >= a(1) + 1040) && (b(2) <= a(2) - 1360)
49         kStop(ref)
50         break
51     end
52 end
53
54 a = kReadPos(ref);
55 kSetSpeed(ref,10000,10000)
56
57 while 1
58     b = kReadPos(ref);
59

```

```

59
60
61     if (b(1) >= a(1) + 8000) && (b(2) >= a(2) + 8000)
62         kStop(ref)
63         break
64     end
65 end
66
67
68 a = kReadPos(ref);
69 kSetSpeed(ref,5000,-5000)
70
71 while 1
72     b = kReadPos(ref);
73
74     if (b(1) >= a(1) + 1040) && (b(2) <= a(2) - 1360)
75         kStop(ref)
76         break
77     end
78 end
79
80
81 a = kReadPos(ref);
82 kSetSpeed(ref,10000,10000)
83
84 while 1
85     b = kReadPos(ref);
86

```

```

86
87
88     if (b(1) >= a(1) + 5000) && (b(2) >= a(2) + 5000)
89         kStop(ref)
90         break
91     end
92 end
93
94 a = kReadPos(ref);
95 kSetSpeed(ref,5000,-5000)
96
97 while 1
98     b = kReadPos(ref);
99
100     if (b(1) >= a(1) + 1040) && (b(2) <= a(2) - 1360)
101         kStop(ref)
102         break
103     end
104 end
105
106

```

W drugiej części zadania napisano skrypt odpowiadający za omijanie napotkanej przeszkody na drodze robota. Skrypt ten polegał na tym że ustawiano robota na wprost przeszkody, nadawano mu prędkość a gdy ten zbliżał się do przeszkody, za pomocą czujników wykrywał ją, zatrzymywał się, a następnie wykonywał obrót o 90 stopni i jechał dalej. Komenda odpowiedzialna za odczyt z czujników to kProximity. Poniżej przedstawiam stworzony przez nas skrypt.

```

1  a = kReadPos(ref);
2  kSetSpeed(ref,10000,10000)
3
4  while 1
5      b = kProximity(ref)
6      if (b(4) > 220 && b(5) > 340)
7          kStop(ref)
8          break
9      end
10 end
11
12
13
14 a = kReadPos(ref);
15 kSetSpeed(ref,5000,-5000)
16
17 while 1
18     b = kReadPos(ref);
19
20     if (b(1) >= a(1) + 1040) && (b(2) <= a(2) - 1360)
21         kStop(ref)
22         break
23     end
24 end
25
26 a = kReadPos(ref);
27 kSetSpeed(ref,10000,10000)
28
29 while 1
30     b = kProximity(ref)
31     if (b(4) > 220 && b(5) > 340)
32         kStop(ref)
33         break
34     end
35 end
36

```

3. Wnioski

Uważamy że za pomocą tak prostych skryptów w Matlabie udało nam się stworzyć dość trudne trajektorie, a do tego wszystko w dość małej ilości czasu. Stwierdzamy zatem, że przy odpowiedniej ilości czasu, można by stworzyć bardzo skomplikowany algorytm obsługujący ciekawe sekwencje ruchów. W trakcie ćwiczenia zauważyliśmy, że sprawdzanie położenia robota za pomocą `kReadPos(ref)` często różni się mimo tego, że robot jest w tej samej pozycji. Z powyższego faktu wnioskujemy iż programy te są przeznaczone dla niezbyt wielkiego obszaru roboczego. W trakcie ćwiczenia pojawiły się także problemy z łącznością z robotem co skutkowało koniecznością resetowania programu Matlab. Pomimo powyższych wad stwierdzamy, że ćwiczenie było nad wyraz ciekawe i pouczające. Pierwszy raz w trakcie trwania naszych studiów mieliśmy możliwość samodzielnie napisać kod i zmusić jakiegoś robota do poruszania się co na pewno wywołało w nas wiele pozytywnych emocji.