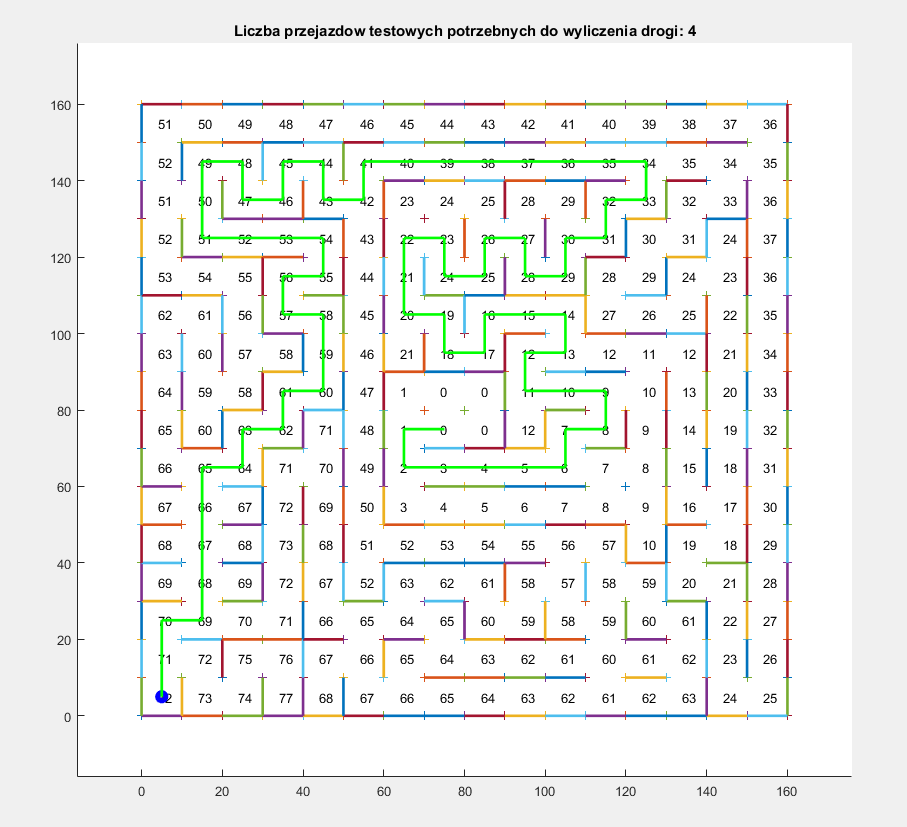
Jerzy Baranowski

Bartosz Grabczak

# Projekt SST:

# Optymalizacja trasy robota typu „micromouse” w przestrzeni dwuwymiarowej

W ramach przygotowania do projektu został napisany skrypt symulujący rozwiązywanie labiryntu o wymiarach 16 x 16 komórek przez robota typu micromouse. W środowisku MATLAB wygenerowana została mapa labiryntu ze zdefiniowanymi ścianami pomiędzy odpowiednimi komórkami, która jest rysowana w momencie wywołania skryptu, wraz z zaznaczoną na zielono najkrótszą trasą przejazdu.

Rys. 1 Efekt działania symulatora

Do rozwiązania labiryntu został użyty algorytm „flood-fill”. Oprócz ilustracji graficznej efektem działania symulatora jest zwracana tablica zawierająca indeksy kolejnych komórek labiryntu, które musi pokonać robot w celu dojechania do punktu końcowego. Tablica indeksów komórek jest punktem wyjścia dla algorytmu optymalizującego krzywą przejazdu ze względu na czas.

Obecne rozwiązanie polega na tłumaczeniu tablicy indeksów komórek na zestaw rozkazów, wykonywanych przez robota, są to ruchy: do przodu o n komórek , w lewo, w prawo, po skosie w lewo o n komórek oraz po skosie w prawo o n komórek. Rozkazy te są przechowywane w kolejce FIFO. Podczas pracy robota po wykonaniu danego rozkazu pobierany jest z kolejki kolejny. Każdy rozkaz ruchu tłumaczony jest na parametryczną krzywą Beziera. System sterowania iteracyjnie generuje kolejne punkty krzywej i dba o to by robot przejechał przez nie. Rozwiązanie to jest efektywne, krzywa przejazdu zostaje wygładzona, lecz nie zapewnia najszybszej, optymalnej trasy dla dowolnej sekwencji komórek do przejechania.

Celem optymalizacji jest wygenerowanie gładkiej krzywej prowadzonej przez wyznaczone komórki w celu uzyskania najkrótszego czasu przejazdu przy ograniczeniach geometrycznych labiryntu oraz samego robota. Najkrótszy czas przejazdu jest równoznaczny z warunkiem najmniejszej średniej krzywizny danej krzywej, gdyż robot może pokonać taką trasę z większą prędkością. Stworzony skrypt może symulować czas przejazdu robota po krzywej, dzięki czemu będzie można porównać efekty przeprowadzonej optymalizacji. Finalnym etapem projektu będzie implementacja nowego w robocie.