DASL 102 - Navigation/Path-Planning

Instructor: Dylan Wallace
Email: wallad3@unlv.nevada.edu
Location: Drones and Autonomous Systems Lab
Course Time: TBD (2-3 hours per class), 1/week
Course Length: 6 weeks + Final Exam

Prerequisites: Due to the nature of this course, heavy use of data structures and mathematical analysis will be used. As such, a person taking this course should have experience with the following:

- C++ Programming, especially data structures and object-oriented programming
- The Robot Operating System (ROS) for use in simulation and visualization
- Vector calculus, especially gradients and vector fields
- Simple trigonometric mathematics, especially for non-holonomic constraints

Overview: This course will cover the basics of navigation and path-planning for mobile robots. The course will focus on many different navigation algorithms, including: Breadth-first Search, Depth-first Search, Dijkstra's Algorithm, A*, and Potential Fields. Many of these approaches follow a data structure implementation, and such the basics of these will also be covered. For simplicity and better visualization, simulation and virtual approaches will be used during most of the course. However, the course will cover how to obtain basic odometry data for implementation on a real robot platform (Robotino). Finally, the course will compare and contrast the different approaches as they are brought up, and will cover the challenges that come with navigation, especially in real-world applications. Attendance for this course is extremely important, as most of the course will be going over the algorithms in detail, both theoretically and in practice. Lecture notes will be available to the students, but will not be adequate enough to properly implement the algorithms covered.

Objectives: At the end of this course, students should be able to:

- Use data structures such as stacks, queues, and trees to organize data
- Utilize simple searching algorithms such as Breadth-first Search and Depth-first Search to find paths toward a goal
- Utilize more constrained searching algorithms such as Dijkstra's Algorithm and A*
- Obtain robot odometry data for robotic navigation
- Utilize vector-based approaches such as Potential Fields
- Know the differences between the approaches, in order to use the best one for a specific application
- Implement the different approaches on both real-world and simulated platforms

Homework & Testing: In order to gauge the student's progress throughout the course, quizzes will be given every week based on the previous lesson. In addition, a preliminary quiz will be given at the beginning of the course to gauge a student's preparation for the course, focusing mainly on C++ programming knowledge, but also on important mathematical concepts necessary for the course. This quiz will not be factored into the total grade, but will be used for the student to brush-up on knowledge for the course. After every lesson, a homework assignment will be given in order to practice the concepts covered that week. Each homework will be graded, and will take 2-6 hours depending on the student's ability. Finally, an exam will be given at the end of the course to test the overall knowledge gained during the course. This will help to gauge the student's actual ability to implement the material covered during the course. Below is a grade breakdown for the course:

- Homework - 50%
- Quizzes - 25%
- Final Exam - 25%

Course Schedule: The following is a schedule for what will be covered each week of the course, and the homework assignment for that week.

Week 1: Potential Fields

- Vector fields
- Potential points
- Potential Field calculations
- **Homework:** Matlab PF Navigation

Week 2: Data Structures Overview

- Linked-lists
- Stacks and queues
- Graph trees
- **Homework:** Family tree program

Week 3: Breadth-first Search

- BFS
- **Homework:** Simple BFS Navigation Program in ROS

Week 4: Depth-first Search

- DFS
- **Homework:** Simple DFS Navigation Program in ROS

Week 5: Dijkstra's Algorithm & A*

- Dijkstra's Algorithm
- A*
- **Homework:** City Navigation Program

Week 6: Odometry Data

- Encoders
- IMU data
- Kalman Filter
- Robotino Implementation
- **Homework:** Robotino BFS