PROGRESS REVIEW #2

INDIVIDUAL LAB REPORT [ILR03]

OCTOBER 30, 2015

TUSHAR CHUGH

TEAM B-AUTOPIRATES
SHIYU DONG, BIKRAMJOT HANZRA, TAE-HYUNG KIM, WILLIAM SETO

PROGRESS REVIEW #2

INDIVIDUAL PROGRESS

This week I worked on following things: graph search algorithms, SBPL Library, CAD assignment, PCB conceptual design and CODR.

Path Planning

a. Graph Search Algorithm

Graph search algorithm is the crux of the path planning section. Although we would using library (SBPL) to implement graph search algorithm but it is very important to understand how these algorithms work. I studied the implementation of Dijkstra's, A* and D* algorithms. I referred to the the A* code which has one of the most optimized performance. I built it on Visual C++ and explored how it has been implemented. Here I learnt how to create an environment in the form of occupancy grid map (matrix) and how to use graph search algorithm to plan the path. Figure 1 shows the test run of the code which finds the path from Node (1,7) to (14,0) in 23 search steps. Node are the positions in the occupancy grid map. I also met Chiyu (Ph.D. student under Prof. Dolan) to understand how we generate occupancy grid maps of fixed size and orientation, generate occupancy grid maps by defining a section of a river with a polynomial and run path planning algorithm on the generated occupancy grid map of the environment.

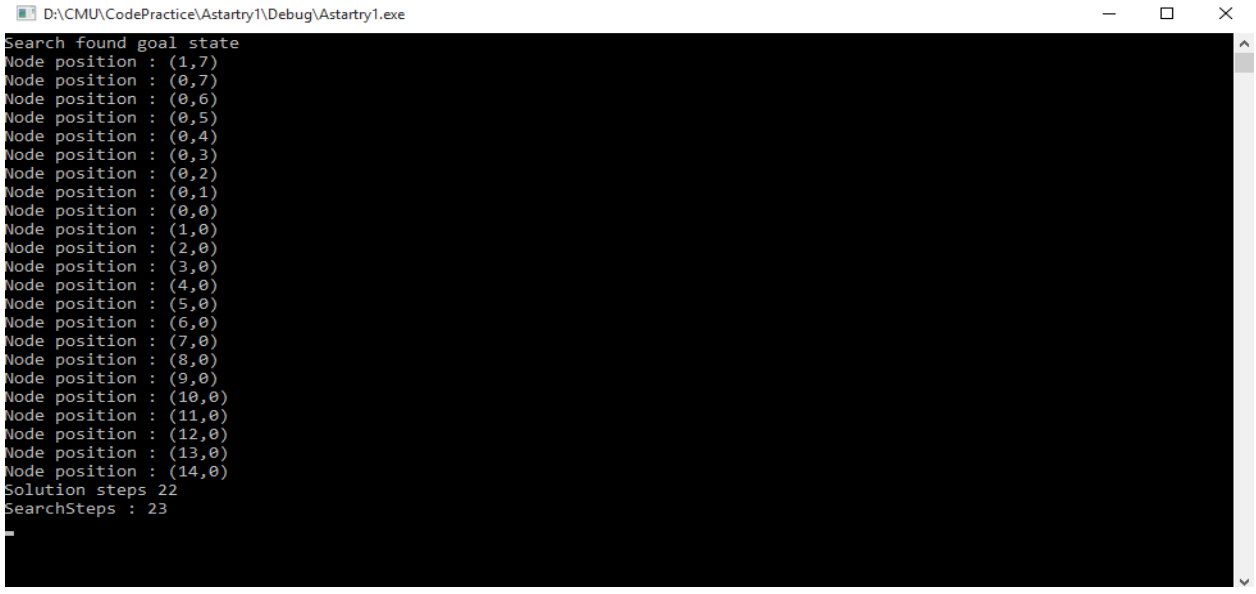

Figure1: A Sample Code Run*

b. SBPL Library

I worked with Bikram to build and explore the sample codes from Search Based Planning Library. The code takes input as object of the environment configuration file(in this case binary data converted from the depth data of willow garage floor), the planner algorithm

(like ARA*, AD*), dimensions of robot, start location and the goal. The output is the planned path from start to the goal and the visualization of the path.

c. CAD Assignment

I created 10 parts and assembly of a toy car which I submitted for the CAD assignment. I learnt through tutorials about using linear and circular patterns, various types mating and creating exploded view.

d. PCB conceptual design

I suggested the idea to the team that we would be designing PCB to create a new visualization/alarm system of our REC boat. The system will indicate when obstacles are near to the boat using red LED's and an alarm. Technically, the PCB would be built on ATMEGA 328 microcontroller with Arduino bootloader which will receive serial commands from ROS (ROS topic will publish obstacle presence for Arduino). The circuit would be connected to the same laptop (through USB 2.0) which will run our perception and path planning algorithm.

CoDR Revision

For this week, I updated the Fall Validation Experimentation section of path planning and the work breakdown for path planning. Figure 2 shows Fall Validation experiments. The major change from previous is that now we have separated generating environment map, demo of path planning and obstacle avoidance into different sections. In addition to it, we would be restricting the scope of demonstration of FVE only to the simulator.

Fall Validation Experiments: Path Planning and Simulator

Step ID	Step Description	Success Condition	Associated Requirements
PL.1	Interfacing with IMU/GPS	Display logged data from IMU/GPS which we would be collecting during a field test	MF.6
PL.2	Generate Environment Map	Display occupancy grid map for 2 miles with filled blocks depicting obstacles (For one of the rivers of Pittsburgh)	MF.5
PL.3	Demonstrate Path Planning	Boat successfully navigates from Point A to Point B in simulator which are 1 mile apart	MF.2, MNF.3
PL.4	Show obstacle avoidance	Boat successfully avoids obstacles in simulator in the above task. Obstacles are static and are simulated through occupancy grip map	MF.5

Figure2: Updated Fall Validation Experiments for path planning

Below are the tasks that we need to do in fall. Green represents the tasks we have completed. Yellow are the ongoing tasks and Red are the one which we haven't started.

Generating Environment Map

1. Getting started with SBPL
    i. Build example codes
2. Understand how an environment is built
    i. Get way points from Open-Street Map
    ii. Understand how grids are created
        i. Grids of static size YxY size
        ii. Representing boundaries of river with a polynomial
3. Create Environment
    i. Identify location for where we need to create environment map
    ii. Create a grid of size Y x Y
    iii. Represent environment outside shores as obstacles

Path Planning

1. How to take input map as the parameter
2. Understand path planning algorithms
    a. A*,D*, ARA*
    b. Run sample codes for A* (standalone in c++/c#)
    c. Run sample code from SBPL Library
    d. Get accustomed with SBPL Library
    e. Study ROS Navigation Package, move_base, Stage Simulator
3. Run Planning code on input environment
    a. Try various inbuilt algorithm with our environment as input
    b. Analyze performance for A*, ARA* and Anytime D*

Sensors Interfacing

1. Interface IMU with ROS Node
2. Record IMU data during test run

CHALLENGES

a. CAD assignment

Initially, I was finding it hard to get started with the CAD assignment as the only experience I had with solidworks was the first two assignments. I went through the tutorial on youtube which helped me in understanding some of the very important features of Solidworks (linear and circular patterns, etc).

b. SBPL Library

There are no good tutorials available for SBPL library online. So, we have to meet the engineer (Andrew) from SBPL team to clarify our doubts.

TEAM WORK

a. **Shiyu Dong:** Shiyu researched on the data format of radar and preprocessing of data. He also revised the risk management and fall validation for perception sections of CoDR.
b. **Bikram Hanzra:** Bikram created the model of the boat in solidworks. He also worked with me to explore functionalities of SBPL. In addition to it, he simulated world of dummy environment on gazebo.
c. **Tae-Hyung Kim:** Taehyung worked separately to build sample code of SBPL library. We approached engineer from SBPL team to understand the library better.
d. **William Seto:** William worked on the integration of OpenCPN and ROS. Also, he worked with the team to revise CoDR (FVE, WBS and critical path)

FUTURE WORK

a. **Path Planning**
   - For this week my work would be focused towards generating occupancy grid map of the environment. This includes the following:
     i. Get coordinates of location from OpenCPN to create occupancy grid map
     ii. Create a grid of size Y x Y (square boxes of occupancy grid map)
     iii. Work to see how this grid can be used with SBPL code

b. **PCB Assignment**
   - I would be working with the team to create the PCB. My primary role would be to document BOM and to work on the schematics of the PCB.
c. **Primary Design Review**
   - PDR is due next week. I would be working on revising the software architecture of the project.