ECE 445 | Project Proposal
Multi-Agent State Estimation in a Partially-Observable Environment

<u>Team 76</u>

Kourosh Arasteh

Junwon Choi

TA: Amr Martini

Introduction

Background

In applications of collaborative robotics, keeping track of the state of the environment is often attempted using individual agents that simultaneously perform localization and mapping onboard. To do so, these agents require powerful computation capabilities and constant communication with both GPS satellites and D-GPS towers. However, there are applications of robots like these in locations that are GPS-denied or contested to the point that extraneous long-range communication should be avoided or is unavailable entirely.

The inspiration behind our project and the problem we're trying to solve comes from the Army Corps of Engineers and the current challenge they face in trying to map and characterize a construction site prior to construction. Many military base locations and military construction sites in developing countries are often GPS-denied and/or contested environments[1]. For such construction projects, collaborative robots pose an attractive solution to the problem of distributed, large-scale construction. However, without a robust mapping process for the environment, it would be impossible to develop a plan for distributed autonomous construction. An accurate and dynamic mapping of the region(s) of interest is necessary before autonomous construction robots can be deployed. Therefore, our problem statement is as follows:

<u>How do we keep track of a large, sparse map between several ground agents, without using GPS or long-range localization technologies?</u>

Objective

Our research aims to solve the problem statement above by utilizing multiple unique agents, each tasked with one aspect of our multi-faceted mapping and localization objective. By utilizing 2 ground agents and 1 "eye-in-the-sky" agent, we'll be able to discretize mapping to our ground agents and localization to our sky agent. Our ground agents' sole objective is to move about our small-scale sandbox and capture small "patches" of map data, and our sky agent's sole objective is to determine the location and orientation of each ground agent. Ground agents relay their patches to a central controller agent (in our case, a PC) and the controller agent receives location data for each patch from the sky agent. Our controller then takes all patches of map data and their respective location data to accurately stitch them into one complete map of our entire environment. Our approach presents 2 novel advantages: considerably lower per-agent cost and significantly reduced computational resources required throughout the mapping process. We're able to observe these advantages because instead of having one intelligent agent performing both localization and mapping, we have multiple "dummy" agents each focused on one single task, thereby decreasing computational complexity, material cost, and overall development time. A further novel aspect could be achieved in one of our growth goals, utilizing agents of different designs and capabilities to map our environment in a scheme that plays to each agent's unique strengths.

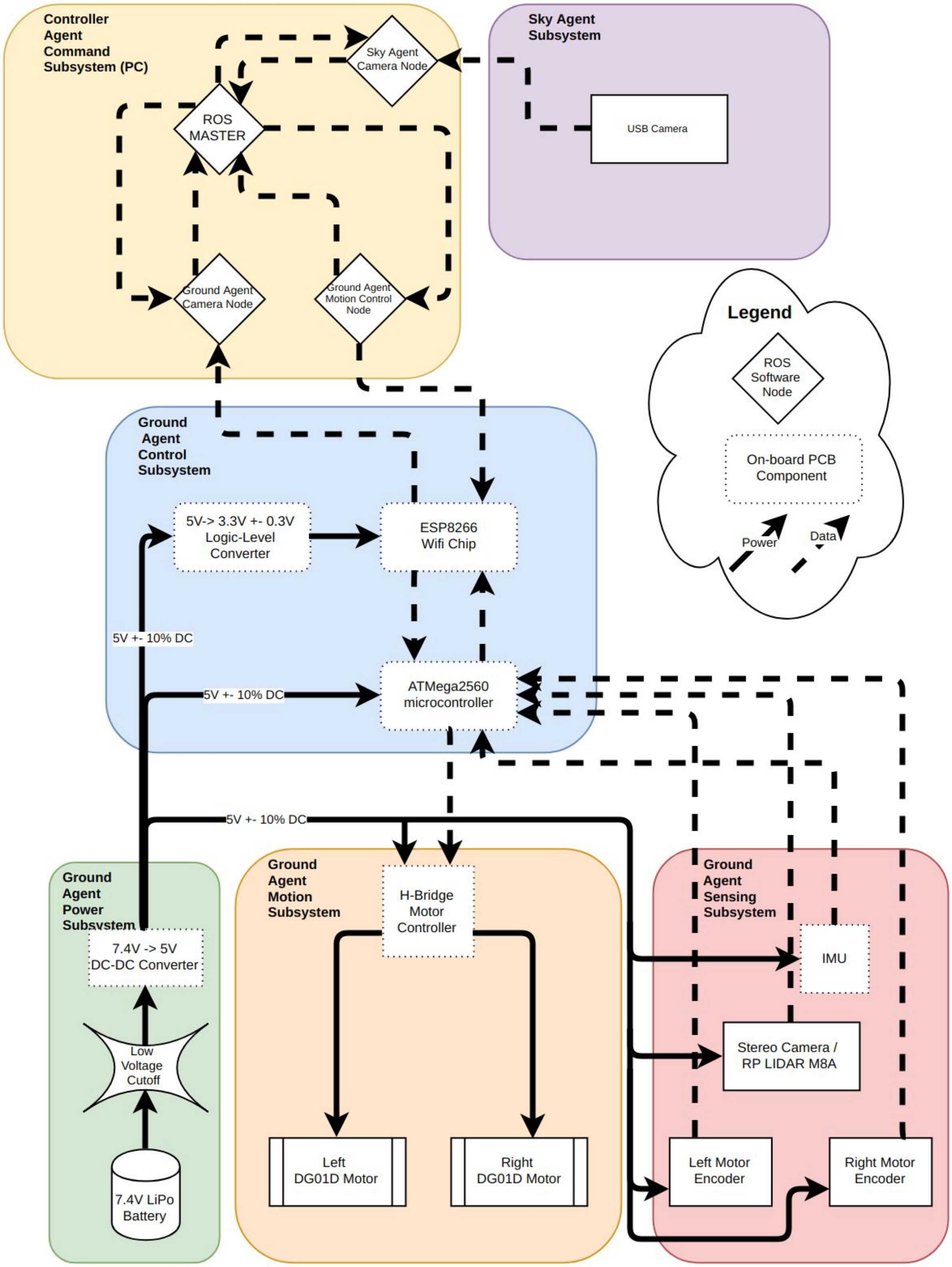
High-Level Requirements

1. System should be able to localize the ground agent position (±5cm) and orientation (±15°).

2. System should be able to take in patch maps of multiple ground agents and place them in the right location and orientation in the global map.

3. System should be able to detect and map all obstacles of size similar to the ground agent ($20cm^2$ to $0.1m^2$ top area)

Design
Block Diagram



Physical Design

Functional Overview

Each Ground Agent is made up of 4 main components: a power supply, a motor system, a sensor array, and a central control unit. The power supply provides a safe power source for the heavy draw of the LIDAR and unique needs of the control chips. The motor system moves the robot according to instructions. The sensing subsystem provides data about the Ground Agent state and the state of its immediate environment. The control unit passes the necessary data up to the controller agent, and receives instructions for new positions to move the motors.

The Sky Agent passes image data to the controller agent to aid in localization of the ground agents. It is essentially a USB camera.

The controller agent is responsible for maintaining the global map, fusing the Sky Agent image data with the Ground Agent mapping patches, and determining the next locations for the Ground Agents to explore.

Block Requirements

<u>1 Ground Agent Power Subsystem</u>

Power is required to run the motors on the ground agent, as well as to keep the microcontroller running and allow the sensors to send data to the microcontroller. This power subsystem should provide 5V to all of the other Ground Agent subsystems.

1.1) 7.4V LiPo Battery

The lithium-polymer battery must continuously provide power to the circuit until the environment is mapped.

Requirement: The battery must be able to provide 7.4V ±0.2V to the rest of the power subsystem for proper operation of the DC-DC converter.

1.2) Low Voltage Cutoff

This element must cut off the bridge from the LiPo battery to the DC-DC converter if the LiPo battery voltage drops below 6.0 V to prevent over-discharge of the LiPo battery.

Requirement: The cutoff device must open the circuit bridge between the LiPo battery and the DC-DC converter if power is in range 6.0V-6.5V.

1.3) 7.4V→5V DC-DC Converter

This DC-DC converter must convert battery voltage to usable 5.0V circuit voltage for the other subsystems of the circuit.
Requirement: This element must be able to convert battery voltages between 6.5V-8.0V to 5.0V ± 0.5V at peak current draw, 1500mA, at scan startup of the LIDAR.

<u>2 Ground Agent Motion Subsystem</u>

The motion subsystem is responsible for decoding motion instructions from the control subsystem and moving the robot in accordance with the instructions provided.

2.1) H-Bridge Motor Controller

This controller needs to convert switch polarity of voltage applied to drive both the left and right motors forwards and backwards at 5V.
Requirement: This element needs to be able to switch and pass 5V ±0.5V to both drive motors.

2.2) Left & Right DG01D DC Motor

These motors move the ground agent around the staged environment. They must operate in a range of 5V ±0.5V in both directions.
Requirement: Both motors must operate in both directions at 5V ±0.5V.

<u>3 Ground Agent Sensing Subsystem</u>

The array of sensors on the Ground Agent is responsible for providing encoder values for both of the motors to the microcontroller, as well as providing the specific force and angular rate of the body. The sensor subsystem must also provide the local ‘patch’ of its environment, represented as a depth map, to the microcontroller.

3.1) Stereo Camera/RP LIDAR M8A

The sensing apparatus of the camera/LIDAR must be able to be queried at a very low rate, which we might cap at 1Hz. It will not need to provide continuous data, but only a single patch scan not to exceed 5Hz.
Requirement: The camera/LIDAR must be able to be queried at least at the sampling rate of 1-5Hz and operate indoors.

3.2) Left & Right Motor Encoders

Each Hall-Effect encoder must provide the position of its motor to the microcontroller in a high-speed control loop for accurate motor positioning, and operate in the 5V ±0.5V range.
Requirement: Each encoder must provide the orientation of its motor ±1° and be queried at a sampling rate of 500Hz.

4 Ground Agent Control Subsystem

The control subsystem is responsible for relaying the sensing data to the controller agent, and for relaying the motor commands from the controller agent to the motor controller H-bridge.

4.1) 5V→ 3.3V DC-DC Converter

This converter is necessary to power the ESP8266 Wi-Fi chip, which requires a specific 3.3V ±0.3V supply.
Requirement: This element must be able to handle input of 5V ±0.5V and output 3.3V ±0.3V.

4.2) ESP8266 Wi-Fi Chip

This chip must handle TCP/IP communications between the controller agent's network card and the microcontroller. It must operate in IEEE 802.11b/g/n standard and handle up to 100kBps packet send/receive rate to manage the scanned data from the ground agent's sensing systems.
Requirement: Provide an 802.11 b/g/n TCP/IP uplink to the controller agent network card to transfer(send/receive) sensor data and instructions at 200kBps.

4.3) ATMega2560 Microcontroller

This microcontroller must handle the data packaging from the sensor subsystems and pass these packets to the Wi-Fi chip for communication to the controller agent. It must also receive and unpack the packets received by the Wi-Fi chip, and decode these messages into motor commands, which it then must pass to the H-bridge motor controller.
Requirement: The microcontroller must be able to communicate over UART with the Wi-Fi chip at up to 200kBps. It must also intake up to 5.0V ±0.5V signals from the sensing subsystem components at a maximum bitrate of 180kBps.

5 Controller Agent Command Subsystem

The controller agent needs to process the image data from the sky agent and the patch information from the ground agent, maintain an updateable global map of the environment, and pass motion control commands down to the ground agents.

5.1) ROS Master

The ROS Master is responsible for registration and interfacing with all other nodes in the ROS structure.
Requirement: Connectivity with all other ROSnodes.

5.2) Ground Agent Camera Node

This Camera node needs to interface with the Wi-Fi chip on each of the Ground Agents over a ROStopic and provide imaging data to the ROS Master.
Requirement: Connectivity with Ground Agent Wi-Fi Chip Camera ROStopics. Connectivity with ROSMaster.

5.3) Ground Agent Motion Control Node

This Camera node needs to interface with the Wi-Fi chip on each of the Ground Agents over a ROStopic and relay instructions from the ROSMaster to the Ground Agents over dedicated ROSmsg/srv.
Requirement: Connectivity with Ground Agent Wi-Fi Chip Controls ROSmsg/srv. Connectivity with ROSMaster.

5.4) Sky Agent Camera Node

This Camera node needs to interface with the USB protocol camera in the sky agent over a ROSsrv service and provide imaging data to the ROS Master.
Requirement: Connectivity with sky agent USB camera over ROSsrv. Connectivity with ROSMaster.

6 Sky Agent Subsystem

This agent is essentially a USB camera that will be connected over hard-wire to the controller agent PC. It must be able to be queried at a speed not exceeding 5Hz.

6.1) USB Camera
This camera must be able to be queried by the controller agent ROSnodes at a maximum rate of 5Hz, and must be powered over USB.
Requirement: Must be able to provide image data at a rate of up to 5Hz, and must be powerable via USB.

Risk Analysis

A large issue that may risk the progress of our project is being able to successfully calibrate all of the sensors at play. Our ground agents will be equipped with LIDARs and the sky agent will be some form of a cheap RGB camera. It is possible we may run into issues combining location data with our map data because our ground agent and sky agent use two entirely different sensors. We can overcome this potential obstacle by including additional metadata about the data packets that our controller agent expects to receive. By doing so, the controller agent can better understand the how data from each sensor is related and provides the stitching algorithm with more context. This also reduces the complexity of our stitching algorithm.

Another risk factor is the stitching algorithm itself. The completion of our environment map relies on our controller agent's ability to stitch patches of map data together using location and orientation data provided by the sky agent. Developing an image processing algorithm to accurately stitch all of our ground agent patches together may prove to be difficult, especially if incoming data from each respective agent isn't necessarily uniform.

The wireless connection aspect of our project could also pose a risk because of the size-intensive nature of LIDAR pointclouds. Hardware on our ground agents could pose limitations on how fast these LIDAR pointclouds are relayed back to our controller agent. We will have to make sure that the pointclouds generated for each patch of map data are as small and distinct as possible. By doing so, we can guarantee that we're passing the smallest pointcloud necessary to form a complete and comprehensive map.

Safety & Ethics

Our project includes several safety hazards that should be clearly identified. Most notably, our ground agents are equipped with LIDARs which are laser devices. The inclusion of LIDARs pose risk of vision damage should an individual look directly into the area where lasers are emitted. The RPLidar A2M8 has been certified by the American National Safety Institute (ANSI) as an FDA Class 1 Laser[2]. This means that it cannot emit laser radiation at known hazard levels (Class 1 lasers are less harmful than barcode scanners). However, Class 1 lasers can still become harmful if viewed through optical aids (such as binoculars or magnifying glasses) for extended periods of time. As a result, the LIDAR should never be brought to eye-level or looked directly into when powered on.

Another safety hazard involves the lithium-polymer batteries used to power our ground agents. Lithium-polymer batteries are necessary as a power source because of the significant amount of power consumed by the LIDAR for mapping. As a result, misuse of these batteries or carelessness can lead to extremely dangerous explosions if overcharged or introduced to extremely hot temperatures. Furthermore, over-discharging these batteries can lead to unpredictable behavior, and even explosions. We need to maintain charging temperatures between 0°C and 60°C and make sure to utilize our aforementioned low-voltage cutoff elements throughout the testing and implementation process.[3][4]

Our project has one main risk in ethics violation, and that's regarding privacy. At a high level, our system can map and localize any given area using multiple ground agents and an eye-in-the-sky. Unfortunately, it is possible that an individual, or a group of individuals, with malicious intent can use our technology to acquire localized map data of a property or region without consent. This is a direct violation of principle 1.6 in ACM's Code of Ethics, which states that computing professionals have a responsibility to respect the privacy of the public and other professionals. [7] Misuse of our technology is also a violation of IEEE's Ethics Code #1, which states that engineers should hold the welfare of the public paramount, and strive to comply with ethical design. [5] This is reiterated in the IEEECS Code of Ethics section 3.12, mentioning that we must 'Work to develop software and related documents that respect the privacy of those who will be affected by that software. [6] To avoid misuse of the technologies introduced in our project, we plan to only run our project in a lab environment and avoid any case of mapping private property. Furthermore, we will maintain this technology as a research project, not for commercial distribution, which will avoid the issue of the public using the technology to invade privacy.

References

[1] “Case study: Military operations in GPS-denied environments,” *CRFS - Spectrum Monitoring and Geolocation*. [Online]. Available: https://www.crfs.com/casestudies/case-study-military-operations-in-gps-denied-environments/. [Accessed: 08-Feb-2019].

[2] FDA.gov. (2019) *Laser Products and Instruments*.[online] Available at: https://www.fda.gov/Radiation-EmittingProducts/RadiationEmittingProductsandProcedures/HomeBusinessandEntertainment/LaserProductsandInstruments/default.htm. [Accessed 06-Feb-2019].

[3] B. Schneider, Brian, and Roger, “A Guide to Understanding LiPo Batteries,” *Roger's Hobby Center*. [Online]. Available: https://rogershobbycenter.com/lipoguide/. [Accessed: 06-Feb-2019].

[4]“Lipo Battery Safety Guide NZ,” *iCharger NZ*. [Online]. Available: https://www.icharger.co.nz/buying/resources-faq/lipo-lithium-battery-safety-guide/. [Accessed: 08-Feb-2019].

[5]“IEEE Code of Ethics,” IEEE - *Advancing Technology for Humanity.* [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 08-Feb-2019].

[6]“Software Engineering Code of Ethics,” *content - IEEECS*. [Online]. Available: https://www.computer.org/web/education/code-of-ethics. [Accessed: 08-Feb-2019].

[7]“ACM Code of Ethics,” *Association for Computing Machinery*. [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed: 08-Feb-2019].