ECE 4011/ECE 4012 Project Summary

Project Title	Semi-Autonomous Salamander Robot Disaster-Response System
Team Members (names and majors)	Alex Popescu, EE Shashwat Sitesh, EE Calvin Yao, EE
Advisor / Section	Dr. Linda Milor / L7A
Semester	Fall 2016, Intermediate (ECE 4011)
Project Abstract (250-300 words)	Robots have the potential to serve as extensions of emergency-response professionals. During the immediate response to a disaster, robots can provide valuable real-time data to help assess and monitor a situation and potentially save lives. In environments with hazardous materials or precarious structures, robots can navigate to places humans cannot. The goal of our project is to develop a semi-autonomous smart salamander robot for such disaster-response scenarios. The robot should be able to traverse a disaster area and perform tasks such as search, reconnaissance, mapping, and structural inspection. Salamander robots have certain properties that make them ideal candidates for disaster-response situations: ● Amphibious capability - can walk and swim ● Mobility - 4 segmented legs, flexible spine ● Stability - low center of gravity ● Portability - small size and weight The current state-of-the-art salamander robot is called Pleurobot. It was developed by Biorobotics Lab at EPFL (École Polytechnique Fédérale De Lausanne) and uses a simple manual operator control mechanism. Our semi-autonomous salamander platform builds upon this design; it will augment traditional operator control with autonomous path planning capabilities and a more user-friendly human-machine interface (HMI). Streamlining and automating the operator interface will improve efficiency and thus overall disaster operation response of the robot. Moreover, we plan to optimize the robot's physical attributes to improve performance in rubble-filled disaster terrain. For our demonstration, we plan to build a prototype model of the streamlined smart-salamander disaster-response system. The prototype salamander robot will demonstrate the robot's semi-autonomous operation and novel HMI by navigating around an unknown mock-disaster-zone commanded by a human operator utilizing our operator-assistance features. Ideally, it will demonstrate environment mapping and inspection.

Project Title	Semi-Autonomous Salamander Robot Disaster-Response System

List **codes** and **standards** that significantly affect your project. Briefly describe how they influenced your design.	● IEEE 802.11n: This wifi wireless local area network standard will be used to communicate between an operator control station and the mobile robot, if it is untethered. This standard supports high-bandwidth data transfer which can be used for transmitting real-time video. ● USB: Universal serial bus may be used to send motor commands from the single board computer to the motor controller, which will then send PWM signals to servo motors. USB may also be used to send information to the robot from the operator control station, if it is tethered. ● PWM: A pulse-width modulated control signal is typically used to send rotation commands to servo motors. The frequency is usually around 50Hz. ● LVDS: Used to send camera data from camera to single-board computer for processing, at very high data rates. ● ROS: Robotic operating system will be utilized to run and coordinate motor control and data collection tasks in a modular fashion on the single-board computer. ● C++: Object-oriented programming language used to create ROS modules.
List at least two significant **realistic design constraints** that applied to your project. Briefly describe how they affected your design.	● Degrees of freedom: The number of motors needs to be sufficient to mimic the articulation of a salamander. Too few motors will create an inaccurate model that may not move how we want it to, and too many motors will be redundant and costly. ● Accuracy of gait: Replicating the gait of an actual salamander for the control system will require recorded data from the movement of salamanders or construction of the walk cycle from scratch, which may be time-consuming and inaccurate when programming each motor.
Briefly explain two **significant trade-offs** considered in your design, including options considered and the solution chosen.	● Battery vs. power tether: An on-board battery would significantly increase robot size and volume, but would free the robot from the distance-limitations and excess drag force of a power tether. A power tether would be beneficial because it provides unlimited operating life and reduced on-board weight. The weight contributions between an on-board battery vs. pulling a long tether are nearly equal, and both limit operating range, so battery was chosen to eliminate tether snags when traversing the targeted rough terrain. ● On-board SLAM vs. overhead camera tracking: On-board SLAM eliminates an overhead camera, yet requires increased sensor payloads (weight) and more computational power. On the other hand, overhead camera tracking limits the environment size yet is an easy way to localize the robot. Overhead camera tracking was chosen to reduce robot weight and localization complexity, so more resources can be spent on robot control rather than localization.
Briefly describe the **computing aspects** of your projects, specifically identifying **hardware-software** tradeoffs, interfaces, and/or interactions. *Complete if applicable; required if team includes CmpE majors.*	Hardware/Software Interfaces: ● ROS (Robotic Operating System) ○ Command the gait and control systems of the robot ○ Read data from onboard camera(s) ○ Receive remote commands from a human controller via gesture-based or joystick-based input ● SBC (Single-board computer) ○ Receive wireless or tethered control signals, process via ROS and send commands to motor controller ○ Receive sensory input from cameras and transmit wirelessly or over tether to operator control station ● HMI (Human-machine interface) ○ Analog controller utilized by operator's hands ○ Converts analog control signals to digital inputs at operator control station, which are transmitted to robot

	Hardware/Software Tradeoffs: ● SLAM (simultaneous location and mapping) Accuracy ○ Adding many on-board hardware sensors, such as ultrasonic sensors, LIDAR (Light Detection And Ranging), GPS (global positioning system), or IMU (inertial measurement unit), will improve localization accuracy. However, these hardware components can be expensive. In comparison, an overhead camera utilizing simple software image processing, can achieve rudimentary SLAM for our robot.