Formal Methods for CPS & Robots
Final Project Proposal

Carl Hildebrandt & Melony Bennis

Problem Definition:
During development, deployment and maintenance of robotic applications both software and hardware is constantly changed, updated or replaced. The effect of these changes is generally difficult to isolate or even observe. We thus propose a project which constructs a set of test cases which identifies and isolates the differences in robot behaviors spurred by hardware changes.

Because this course entails understanding and constructing formal specifications for cyber physical systems, we chose to model hardware changes. This technique could also be easily applied to software systems, but for the scope of this assignment we have decided to apply it to the hardware system in order to more clearly observe any system changes.

Significance:
Each robot is equipped with a sensor that has a set of specific attributes. Changing this sensor will change or augment the set of attributes, ultimately causing the robot to exhibit a new set of behaviors. Creating test cases which identify the differences in attributes and exhibited behaviors will allow for a visualization of the differences of each model.

Novelty:
Identifying the behavioral differences in robotic changes is difficult. This projects novelty lies in creating a technique which formally models the differences allowing the generation of test cases which isolate and expose these differences.

Design Overview:

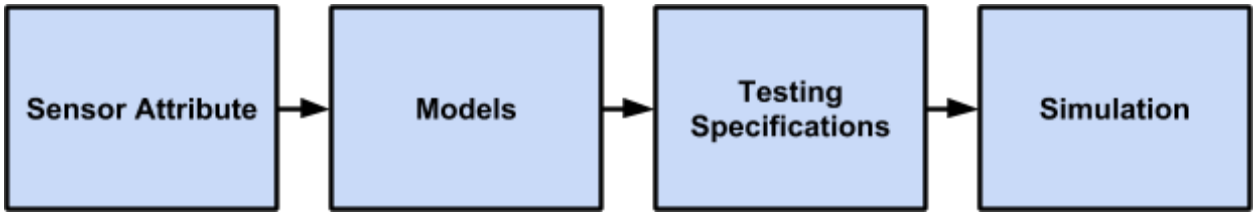

Figure 1: Depiction of project flow

For the project we will be using a Husky Robot[1]. The Husky robot use the Robotic Operating System(ROS)[3]. Specifically it uses the ROS navigation stack to do simultaneous localization and mapping (SLAM), path planning and obstacle avoidance[4]. Each Husky Robot is equipped with a laser scanner. Scanners are not limited by type or model. We will be monitoring changes to the laser scanner hardware.

Each of the laser scanners will have a set of attributes. For instance: range, minimum angle, maximum angle, and increment distance. We will replace the laser scanner on the robot, with a laser scanner containing a new set of attributes. From these attributes we will generate two models: one of the original robot and one of the new robot.

These models will be represented as a set of constraints. The constraints will be fed into a SMT solver in order to generate a set of environment coordinates. At each of these coordinates we will place an object. The object will have size and rotation calculated by the SMT solver. We will be using the Gazebo simulator[2] to simulate the robot and the environment. Both the original and the new robot will be run in the simulator and they will display different behaviors.

Sources

[1] https://www.clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/
[2] http://gazebosim.org/
[3] http://www.ros.org/
[4] https://wiki.ros.org/navigation