Intelligent Ground Vehicle

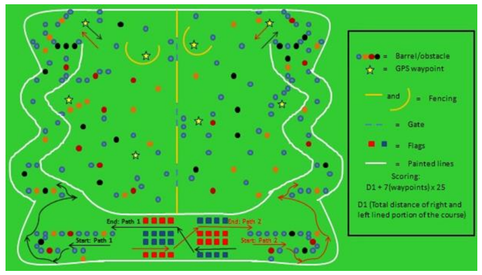
Brent Allard, Adam Hill, Chris Kocsis, Rohit Kumar, Will Nyffenegger, Matt Salfer-Hobbs, Kartik Sharma
Faculty Advisors: Dr. Marius Silaghi, Dept. of Computer Science, Florida Institute of Technology
Dr. Matthew Jensen, Dept. of Mechanical & Aerospace Engineering, Florida Institute of Technology

Objective

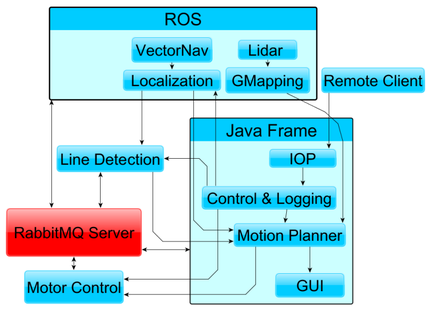
Develop an autonomous robot capable of competing in the Intelligent Ground Vehicle Competition (IGVC) in partnership with Florida State University.

FIT Tasks

- Identifying lanes & obstacles
- Navigating & mapping an unknown space
- Mechanical design
- Power requirements & RC
- Software / hardware integration



Software Architecture



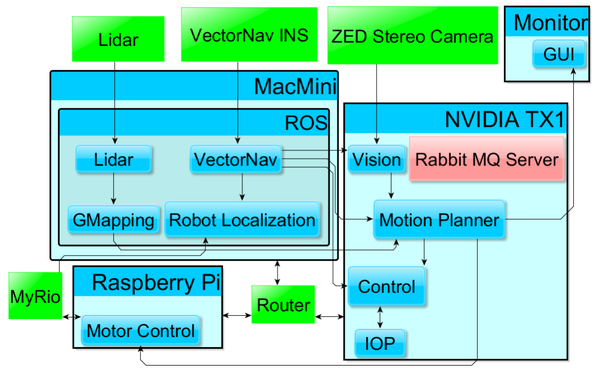
Major Tools
- RabbitMQ, ROS
- OpenCV, PCL
- Java, C++, CUDA
- JSON

Modular Architecture – supports multiple processes in different languages

Communication – RabbitMQ & ROS based framework allow language independent communication using JSON

Control & Interoperability (IOP) – SAE JAUS standards define interfaces to be used for attributes that satisfy design requirements

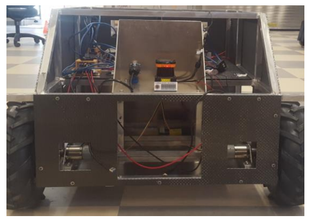
Design

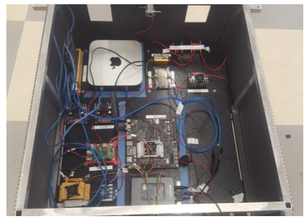


Steering – Differential steering with four wheels

Body – Aluminum frame & carbon fiber siding

Motor Control– MyRio & RoboClaw interfacing through a Raspberry Pi



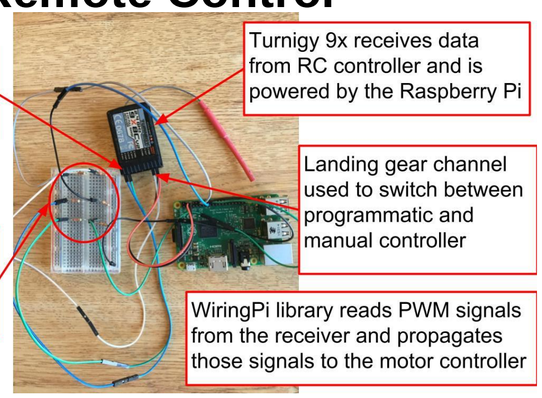


Line Detection – NVIDIA Jetson TX1 & ZED Stereoscopic camera for line detection

ROS – MacMini for ROS applications including Lidar and position estimation

Localization – VectorNav INS for position estimation

Remote Control

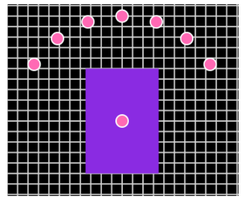


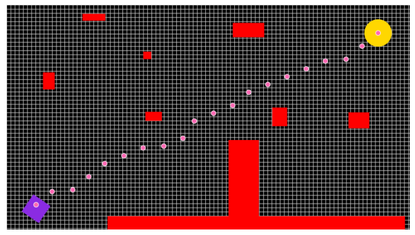
Navigation

Course Representation – Discretized empty map filled in as obstacles and lines are reported

SBMPO – A path is found by sampling the robot's control space against the map. As a result paths are sequences of commands.

D* Lite – Records the best path to individual squares thereby avoiding recalculating paths from scratch when new obstacles are detected





Line Detection

Filtering – Utilizes homomorphic filtering to normalize illumination across a frame followed by low pass filtering which removes noise

Edge Detection – Edge detection using Canny Edge detection algorithm in conjunction with Hough Lines to create lines

Image Processing – OpenCV provides GPU based implementations for image processing

Acknowledgements

NORTHROP GRUMMAN
Engineering & Science Student Design Showcase
at Florida Institute of Technology