PROJECT PLAN

Intelligent Ground Vehicle Competition

Florida Tech Senior Team Members

Name	Major	Email
(Team Lead) William Nyffenegger	CSE Spring 2017	wnyffenegger2013@my.fit.edu
Adam Hill	CSE Spring 2017	ahill2013@my.fit.edu
Brent Allard	CSE Spring 2017	ballard2014@my.fit.edu
Chris Kocsis	CSE Spring 2017	ckocsis2007@my.fit.edu

JANUARY 18, 2017

FLORIDA INSTITUTE OF TECHNOLOGY

Florida Tech IGVC Team

Goals

The goal of Florida Tech IGVC remains winning the annual Intelligent Ground Vehicle Competition (IGVC) in Michigan. As a recap IGVC is an autonomous robot competition with random courses that include random obstacles, paths, and goals. The 2017 rules have slightly revised the competition. Instead of two separate courses, one basic and one advanced, the competition will feature a single intermediate course.

Winning the competition requires satisfying three requirements:

- Running the random course in the least time
- Developing an interoperable system that satisfies the Society of Automotive Engineers (SAE) JAUS standards
- Documenting and demonstrating an innovative hardware and software design

The long-term goals for Florida Tech are broader. The goal of this year is to create hardware and software, which may be extended to compete for the next five years. Additionally, the project must continue developing a relationship with FSU to develop opportunities for collaboration on new projects in the future.

With a semester's research and design, the goals regarding the software may be condensed to:

- Developing image processing capabilities using advanced cameras to detect lines
- Implementing planning and navigation algorithms to navigate an initially unknown space
- Writing and communicating with interactive motor control software
- Estimating position using GPS and IMU hardware
- Developing a communication framework allowing multi-process and multi-language communication
- Integrating hardware and software components
- Developing an extendable software framework built on CUDA, Java, and C++ 11
- Maintaining a logging and control framework that satisfies SAE's JAUS standards

Florida Tech is collaborating with Florida State to work on this robot. Florida State is responsible for most of the hardware oriented design and fabrication. To date Florida State has completed most of the fabrication, motor control, and emergency stop requirements. Additionally, Florida State is working with Lidar and GPS hardware.

Approach

Over the last six months, both universities' roles have crystallized. Florida Tech is specifically responsible for:

- Computer vision and image processing for line detection
- Implementing SBMPO for motion planning with additional support for navigation
- Writing the software framework and interoperability test client
- Creating the communication framework for independent software components
- Troubleshooting the INS (GPS & IMU sensor)

- Integrating the software components
- Controlling startup and logging behavior
- Writing the concurrency aspects of the software

Florida State will handle several software components including motor control, and Lidar.

Novel Features

Though the course has changed slightly, the competition has not.

The course features:

- An unknown, random map which cannot be previously mapped
- Obstacles and roads, which require object recognition and motion planning
- Software memory of a course
- Variable conditions for sensors
- Limited speed and capabilities

Additionally, the competition awards points to teams that provide interoperability. Interoperability, as defined by the JAUS standards proposed by SAE, focuses on the ability of an outside device to accomplish several tasks: communicate with the robot, monitor and log behavior of the robot, query the status of the robot's components, and control the behavior of the robot.

The robot must possess several advanced capabilities to compete coalescing around these ideas:

- GPU programming to optimize obstacle and line detection
- Navigating a space in real time using SBMPO planning
- A communication framework based on AMQP
- Position estimation accurate within one foot
- Experimental sensors and computer hardware with extraordinary capabilities

The novelty of these features manifests in their application. The experimental board used, an NVIDIA TX1, has extreme capabilities. These capabilities allow us to conduct image processing entirely on the GPU. This includes collecting data from sensors, filtering that data, and then running image processing algorithms to cluster points and identify shapes. The libraries and hardware available allow us to process many images a second at extremely high resolution. The conclusions we draw from such images then allows us to map the course.

SBMPO is a motion planning algorithm that leverages simple path planning algorithms combined with statistical knowledge of planning performance. The algorithm, instead of searching the known map, searches the possible movements the robot may make and then tests whether those movements are valid and/or optimal based on the map it possesses. The optimal movements also give you the optimal locations to travel to. This algorithm is demonstrably faster and simpler than other planning algorithms.

Using a communication framework is not original. However, using AMQP on a robot is original. AMQP (Advanced Message Queueing Protocols) allow many messages to be routed from many sources to many destinations. AMQP is typically used to program servers, websites, and apps. Often robots are programmed using custom communication libraries between each component of the system. AMQP presents a standardized way to conduct messaging between multiple languages and processes as if the processes were directly connected.

Technical Challenges

Last semester the technical challenges the team faced focused on research and design. Areas that the team lacked experience and knowledge included:

- Inexperience using AMQP messaging in C++
- Inexperience with GPU programming
- Image processing algorithms
- Motion planning algorithms for navigating updating spaces
- Parallelizing any software components, some computationally intense
- Integrating high level software with low level sensors
- Accurate position estimation

Last semester's efforts eliminated many of the challenges. A communication framework for the project has been written. A motion planning algorithm has been researched and programmed. Our hardware and software are integrated individually. We have experience programming in CUDA. Finally, the NVIDIA TX1 board and the communication framework have eliminated most concerns about parallelization.

Though we have eliminated challenges, new challenges have arisen. This semester's challenges focus on integrating our software and optimizing the performance of our software. The most difficult challenges facing the team focus on optimizing the information gained from hardware.

Specifically, these challenges are:

- Determining an accurate position estimation using an advanced Inertial Navigation System (INS)
- Refining image processing algorithms to clearly define lines without false positives
- Integrating Lidar for obstacle detection
- Translating high volumes of XML requirements into Java code
- Implementing an interoperability testing framework
- Controlling and monitoring all software components

Image processing and accurate position estimation remain challenges. Despite extensively researching image processing algorithms, there is still an issue with accuracy when perceiving white lines on grass. GPS position estimation is not accurate enough for the competition. Instead, the team must combine GPS data and IMU data to get a more accurate position. We secured an INS, which may be capable of giving us that position; however, if the INS proves incapable a Kalman filter must be written.

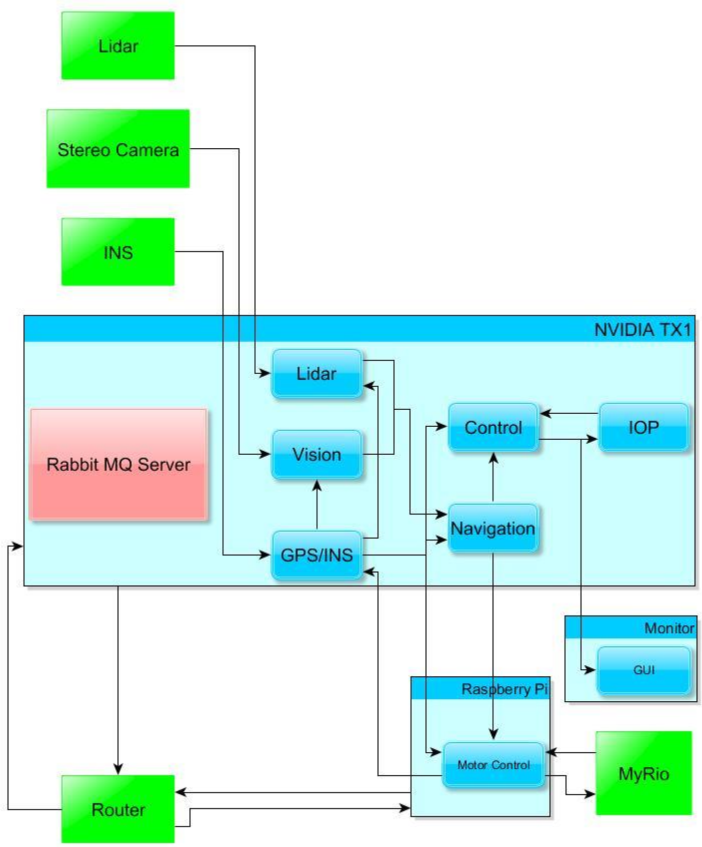
The competition rules also changed. The removal of flags from competition simplified the course. Removing flags leaves only generic obstacles. Instead of relying on a stereoscopic camera, which is much more intensive to program, a 2D Lidar will suffice. That Lidar needs a corresponding client that integrates it into line detection.

The competition awards points to teams that implement the JAUS interoperability standards for autonomous vehicles. In accordance, we have purchased the requirements for the interoperability standards and are now attempting to translate them from XML to Java. With those requirements translated we can then implement queries on the communication framework to retrieve or automatically store interesting information. The JAUS standards also require that the vehicle be operable remotely.

Finally, all data concerning the performance of each software component must be monitored or changed as necessary.

Design

The system architecture is still developing partially. Because of the change to Lidar for obstacle detection, another software component will be added. Furthermore, an interoperability unit will be required and a position estimation unit in addition to the INS unit may be required.



The known software components as follows are:

- Vision: GPU image processing and message communication

- Navigation: SBMPO motion planning and navigation
- GUI: robot state display
- Control: log and monitor state of all other components and communicate with the IOP unit
- IOP: receive all JAUS queries and execute them, possibly assume control of the robot
- GPS/INS: publish data concerning the location and heading of the robot
- Motor Control: execute commands from navigation and report supplementary data on the location of the robot
- Position Estimation*: estimate position using all known data including encoder values, IMU data, and GPS position data
- Lidar: locate obstacles relative to the robot and report those obstacles to navigation
- Communication: a RabbitMQ-Server supporting a communication framework on a local network

Design Summary

The software is housed on two hardware devices: an NVIDIA TX1, and a Raspberry Pi 2. Most of the software resides on the TX1. RabbitMQ Server is installed on the TX1 with client code written in Java and C++. The computer vision section of the code depends on the TX1 for GPU capabilities as well as the driver software for the stereoscopic camera.

A router sets up a local network between both devices that allows messages to be passed through the server between software units. Besides RabbitMQ and the vision aspects of the code, none of the code must be on the TX1. Currently, we are planning to place everything except motor control on the TX1. The Raspberry Pi provides pins that are needed to communicate with a MyRio controlling the motors. However, if the TX1 has insufficient pins to support all devices, more of the hardware will be moved to the Raspberry Pi and correspondingly more of the software units.

Progress Summary

Our completion percentage is also a function of FSU. Florida State has been unable to program certain aspects of the project so our workload has increased. Furthermore, much of the work is still integration.

Module/Feature	**Completion %**	**To Do**
Line Detection	67%	Remove false positives
Motion Planning	67%	Add units, finish D*, and add buffer for time to turn and stop
Communication	80%	Add messages
IOP	20%	Parse XML to Java and implement Control
Control	20%	Add logging messages for status of C++ units
Motor Control	67%	Set comm. protocol
GPS/INS	25%	Fix INS errors on Linux
Lidar & Obstacle Detection	0%	Obstacle and Lidar detection are synonymous

Spring Milestones

Milestone 4

This milestone reflects a milestone set by our advisor for late January. The milestone will be a demo of the robot navigating a spray-painted grass road. To meet this milestone several software components must be functional including the vision component, the motion planning component, the motor control component, the communication framework, and the control component.

Task 1

Line detection depends on the ability of our software to remove noise from the filtered images collected. We have implemented hardware fixes (polarization film), but now need to enter software fixes to further reduce noise. These fixes focus on luminescence, image coarseness, and clustering techniques.

Task 2

All software components must be combined into one software system. The communication framework will be the glue that holds components together. Components that need to be integrated include motor control, vision, motion planning, control, and GPS.

Task 3

Motion planning is almost complete. Currently, generic units are used to navigate spaces; those units will be standardized with the GPS, vision, and motor units. Additionally, the motion planning algorithm still needs several optimizations to speed calculation, including incorporation of the incremental pathfinding algorithm D* Lite.

Task 4

FSU has tested the IMU capabilities of the INS on Windows. However, FSU is unable to collect data from the GPS and IMU sensors on the device. The issue revolves around drivers, which is in FIT's wheelhouse. The INS will be shipped to Melbourne where several coders shall work on producing data.

Task 5

The communication framework is complete. Messages may be sent from Java and C++ processes encoded as JSON. The tools used on both sides to serialize and de-serialize JSON are RapidJSON and Gson. Now specific messages between software components must be coded.

These messages are:

- Lists of lines
- Lists of obstacles
- Lists of commands
- Motor controller data
- GPS data
- Logging messages
- State change data (commands from users)

Each message must be encoded in a data structure in both Java and C++.

Task 6

Communication between the MyRio and the TX1 must be standardized to allow 2-way communication and to forward commands as necessary. This involves concurrency tasks which may be outside the scope of FSU's capability.

Task 7

With the communication framework completed, starting independent processes, and logging their performance, may now be completed. A Control software component will be completed in Java that allows logging messages sent on the framework, behaviors performed, and errors. The component will also be capable of changing other software component's states.

Task 8

Finalizing the GPS unit for the demo must be completed. We will be using a GPS with a base station because position estimation issues will not be resolved by then. The data must be translated and broadcast on the communication framework.

Task Matrix

#	Task	Brent Allard	Adam Hill	Chris Kocsis	Will Nyff.
1	Line Detection	0%	0%	100%	0%
2	Software Integration	15%	25%	15%	45%
3	Motion Planning	70%	30%	0%	0%
4	INS Troubleshooting	0%	0%	50%	50%
5	AMQP Setup	0%	0%	0%	100%
6	Motor Control Comm.	0%	0%	0%	100%
7	Startup, Control, Logging	10%	60%	0%	30%
8	GPS (for Demo)	0%	0%	0%	100%

Milestone 5

Task 1

Line detection will be optimized another software component using Lidar to detect obstacles will be constructed. Though being CPU based, the Lidar component will basically be a clone of the Vision component. Similar algorithms from OpenCV will be used to identify obstacles which will then report to motion planning.

Task 2

Team members shall collaborate to incorporate software components into the system and debug that system. Specifically, we shall be interested in maintaining an accurate time across the system and logging the behavior of the system. Attention shall be paid to enhancing reliability for this milestone.

Task 3

Motion planning will continue to be refined. Particularly, compensation for the time necessary to change trajectory will be accounted for as well as accumulated error in position estimation.

Assuming pathfinding is found to be fast and effective, efforts will be directed towards fine tuning and tweaking for the competition course.

Task 4

The INS will be connected to a Linux machine and data retrieved and tested for precision and accuracy. That data will then be used to replace the GPS unit. If necessary, a position estimation unit will begin.

Task 5

A GUI displaying data known to the Control unit of the software will be constructed to improve testing capabilities and provide information on the performance of the robot. This GUI will be constructed in Java and rely on the communication framework for reporting changes.

Task 6 & 8

Controlling the robot relies on understanding the state of the robot. So logging messages and state control have to be implemented. IOP will also use messages collected in the control unit to relate the robot's state to the user and if necessary change the robot's state. The Control component built for Milestone 4 will be expanded in Milestone 5 to accommodate the performance the needs of IOP. Specifically, functionality will be added to dictate where the robot moves.

Task 7

Incidental additions to communication frame will be made as the Lidar module is added. Additional messages will also be incorporated.

Task 8

The team shall collaborate to produce a poster show casing the robot's capabilities. The poster will reflect the interdisciplinary nature of the project. Additional materials will be produced to provide more information on the code.

Task Matrix

#	Task	Brent Allard	Adam Hill	Chris Kocsis	Will Nyff.
1	Lidar & Lines	0%	0%	70%	30%
2	Software Integration	25%	25%	25%	25%
3	Motion Planning	70%	30%	0%	0%
4	INS Setup	0%	0%	50%	50%
5	GUI	100%	0%	0%	0%
6	Startup, Control, Logging	10%	60%	0%	30%
7	Comm. Maintenance	20%	20%	20%	40%
8	IOP	20%	70%	0%	10%
9	Create Poster	25%	25%	25%	25%

Milestone 6

Task 1

Line detection and obstacle detection should be complete. Focus shall be on improving the accuracy of the reported location and shape of obstacles and lines. A major factor in that will be finding an accurate location.

Task 2

Software integration shall focus on testing and performance evaluation. Software units are expected to function together before the last milestone. Additional behavior, especially behavior focused on position estimation will be thoroughly tested.

Task 3

Any desired additional features will be added to the GUI. Any remaining motion planning edge cases will be dealt with. Documentation of the GUI and pathfinding sections of code will be complete.

Task 4

Extensive work will continue. Specifically, the heading and location of the robot will always be known by the motor control, Lidar, vision, and motion planning aspects of the robot. This will be the critical piece of the software

Task 5 & 7

Compliance with the JAUS standards for IOP will be achieved. The robot will be able to start and stop components upon failure.

Task 6

Messages will be added to the communication framework as necessary.

Task 8 & 9

The team will work to produce a user manual for running the robot as well as understanding the software. The software manual will be created using Doxygen and contain notations on how to affect the robot's behavior

Task 10

Competition documentation demonstrating the innovative aspects of the robot will be drafted. Finished versions of these documents will be presented as part of the competition.

Task Matrix

#	Task	Brent Allard	Adam Hill	Chris Kocsis	Will Nyff.
1	Lidar & Lines Refining	0%	0%	70%	30%
2	Software Integration	25%	25%	25%	25%
3	Motion & GUI Cleanup	70%	30%	0%	0%
4	INS & Position Est.	0%	0%	50%	50%
5	Startup, Control, Logging	10%	60%	0%	30%
6	Comm. Maintenance	20%	20%	20%	40%
7	IOP	20%	70%	0%	10%

8	Create user manual	25%	25%	25%	25%
9	Create demo video	25%	25%	25%	25%
10	Create Competition Documentation	25%	25%	25%	25%

Faculty Sponsor: Dr. Silaghi

I have discussed with the team and approve this project plan. I will evaluate the progress and assign a grade for each of the three milestones.

Signature: ______________________________ Date: __________________