Florida Tech IGVC Milestone 2 Report

Brent Allard <ballard2014@my.fit.edu>
Adam Hill <ahill2013@my.fit.edu>
Chris Kocsis <ckocsis2007@my.fit.edu>
William Nyffenegger <wnyffenegger2013@my.fit.edu>
Dr. Marius Silaghi <msilaghi@cs.fit.edu>

Milestone Two Task Completion Matrix

#	Task	Percentage	Will	Adam	Chris	Brent	To do
1	Finished Navigation GUI	100	0	0	0	100	none
2	Prototype Navigation	100	0	30	0	70	none
3	RabbitMQ C++ Clients	80	80	0	20	0	debug
4	Line following	66	0	0	100	0	final implementation
5	Prototype Control and IOP	20	20	80	0	0	get document
6	Performance Testing ZED	100	0	0	100	0	none
7	Equipment Specs	100	100	0	0	0	none
8	Evaluate INS Driver	100	50	0	50	0	none

Milestone Two Discussion

Task 1 & 2

SBMPO navigation has been implemented with a supplemental GUI written to test that implementation. The GUI allows the user to design basic simulations to test the navigation algorithm and provides metrics for its performance. Obstacle navigation has been achieved. The GUI will be extended to encompass the entire robot as additional pieces are brought online. Specifically, position, velocity, and control status will all be displayed. Other peripheral tasks that navigation needs to proceed forward include the implementation of a Java client and definition of the type and structure messages sent between components of the robot.

Task 3

Extensive effort has been put into producing a standardized C++ client that will work for all components of the robot coded in C++. The client is mostly done including a subscribing event loop with callbacks. Details regarding the protection of data will be made regarding each component, because several threads may share data. The Java standardized client has been debugged and is ready for use.

Task 4 & 6

Line following algorithms are still being explored. Issues with noise caused by the lighting conditions, weather conditions, and course conditions have contributed to concerns about the performance of some of the algorithms selected. Testing using multiple libraries including OpenCV and PCL has proceeded. While the ZED stereoscopic camera provides specifications, we are doing our own testing of the ZED's performance to determine where the camera should be placed on the robot and what kind of frame rate for data we will be able to achieve.

Task 5

Unfortunately, the lack of progress on this milestone is out of the team's control. The team awaits a clarification from the competition judges on what JAUS standard (by year) should be used. have asked the competition to specify what SAE JAUS standard (by year) our code needs to satisfy. The competition judges have not yet replied to multiple requests. If no clarification is received by the beginning of November, then we will select the 2009 standard for use.

Task 7

The equipment specifications regarding the routers, INS, and communication between FSU's motor controllers and our GPU board have for the most part been determined. Specifically, the router will be part of a local network

Task 8

FSU's INS (inertial navigation system) has a driver which must be ported from the QNX operating system, which focuses on parallelization, to Arch Linux version of Ubuntu. Fortunately, through evaluation of the drivers and libraries, the driver should be directly portable with the installation of a couple libraries. FSU will set up a machine which the INS can be remotely tested on.

Contributions

1. William Nyffenegger: Contributed to building the C++ communication framework, porting the INS driver to Ubuntu, and figuring out the equipment specifications. Researching how JSON works with C++ and how to minimize issues.
2. Chris Kocsis: Developing image processing code for line following to avoid CPU bottleneck. Helped debug communication library for AMQP. Chris also began performance testing the ZED.
3. Brent Allard: Implemented navigation algorithm and GUI to test behavior of algorithm. Simulations may be built and recorded to test time, steps, and reliability of the implementation
4. Adam Hill: Working on IOP implementation and navigation algorithms. Adam has been stuck waiting for documentation to really begin implementing IOP and Control components.

Milestone Three Task Completion Matrix

#	Task	Will	Adam	Chris	Brent
1	Finished GUI	15	15	0	70
2	Optimized Navigation Algorithm	0	30	0	70
3	RabbitMQ Clients for each software component	100	0	0	0
4	Finished Line Following	0	0	100	0
5	LIDAR Integration	25	0	75	0
6	IOP Test Client	0	100	0	0
7	IOP Nav Platform	0	100	0	0
8	Control Component	50	50	0	0
9	Integration Testing MQ Clients and IOP	50	50	0	0

10	Integration Testing Components	25	25	25	25

Milestone Three Summaries

Task 1 & 2

Currently, a GUI has been built to test navigation. That GUI will be extended and modified to display information concerning the performance of the whole robot. This information includes position, image, IOP, and navigation information. The GUI will be attached to either the control unit or navigation unit. Its placement within the structure of that component will be determined.

The navigation algorithm will be optimized using known optimizations and then extended based on its performance with simulations. A client will be written for the navigation unit with the capability to receive obstacle data.

Task 3

Java and C++ clients will be written for every component of the robot. Messaging standards will be attempted wherever possible to complete as many of those clients as possible. The initial focus of the clients will be on responses to requested messages from the control unit as well as communication between the control and IOP unit.

Task 4 & 5

Line following shall be completed and LIDAR integrated into the line following to form a vision unit. The focus for the milestone will be line following primarily. The relative difficulty of implementing LIDAR is simple. Testing of the vision unit shall focus on line following in various conditions.

Task 6 & 7 & 8

Regardless of whether the team receives clarification from the competition judges, progress will be made towards satisfying the IOP challenge. The navigation and management platform required to satisfy the challenge shall be implemented as well as a test client to verify its performance. The development of the IOP client will proceed in parallel with the development of the Control component. Messaging between the Control and IOP component shall be completed as well as setting up the framework for messaging between the Control component and other components on the machine.

Task 9 & 10

Integration testing the communication clients will proceed between languages particularly focusing on translating messages into and from JSON in a standardized manner that accommodates for the Unicode specifications concerning JSON. The overall goal is to end the semester with a working vision client that communicates with a navigation component. The navigation component will map the course.

Faculty Sponsor Signature

Sponsor Signature

Evaluation:

William Nyffenegger	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Chris Kocsis	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Brent Allard	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Adam Hill	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10