ELEC/CENG 399 Design Project

Final Project Report

by

Kazu Arai, karai@uvic.ca

Rudolf Erasmus, rudolf@uvic.ca

url: www.web.uvic.ca/~karai

Supervisor: Dr. Colin Bradley, Dr. Alexandra Branzan Albu

Group: 3

Due: 3 August 2012

Dept. Electrical and Computer Engineering

University of Victoria

Table of Contents

Table of Figures

1 Goals

The goal of the ELEC/CENG 399 course for our team was to create a plan for an unmanned ground vehicle to compete in the Unmanned Systems Canada Student Competition in May of 2013 [1]. A secondary goal of this project is to garner more interest in the project to recruit more members. The secondary goal would serve 2 purposes, improvement of the engineering students in a hands-on robotics and mechatronics project and the formation of a new student group at UVic. Using the time between the end of the ELEC/CENG 399 course and the beginning of the ELEC/CENG 499 course, to gain financial support and order parts for the planned robot.

2 Project Overview

Project Title: Unmanned Ground Vehicle

Supervisor: Dr. Colin Bradley, Dr. Alexandra Branzan Albu

At the University of Victoria (UVic) there are two unmanned vehicle teams, AUVic (Autonomous Underwater Vehicle UVic) [2] and UVic AERO (Aeronautical Engineering Research Organization) [3]. Each team has an unmanned vehicle that represents UVic in competitions across North America. Unmanned Systems Canada, an association that promotes the development of unmanned vehicle systems, holds a competition every year for unmanned vehicles where other universities like Simon Fraser University and University of Alberta compete [4]. The goal is to utilize ELEC/CENG 399 to research and design an unmanned ground vehicle (UGV) and to use ELEC/CENG 499, in January of next year, to build the UGV. By doing this, the UGV will be ready by May next year to compete in the 2013 Unmanned Systems Canada UGV student competition [1].

Research will be very important during the initial development phase. The UGV has to meet all the physical, hardware and software requirements to be compatible with the simulation software utilized by Unmanned Systems Canada during the competition. The requirements for last year’s competition are presented in Appendix B. Other factors such as sensor compatibility, multi-core vs. multi-processor, and movement capabilities will need to be researched.

Once the initial requirements research has been completed, the minimum specifications of the UGV can be determined and research will continue into the design phase, marking the first major milestone. Research and design will then focus on the “How” and the “What” of the UGV, for example: circuit design, parts, sensors, and the robot design. The next milestone will be once all designs are complete.

The deliverables from this ELEC/CENG 399 course will include the research, testing and simulation of the UGV components. These designs will allow the UGV to be built in the ELEC/CENG 499 course starting in January of next year and the UGV will be able to compete in next years Unmanned Systems Canada UAV student competition.

3 Detailed Project Description

The purpose of this project is to design, build, and test a UGV which will be used to compete in a competition. The planned base of this UGV will be a MobileRobots Pioneer [5] shown below.

Figure 1: MobileRobots Pioneer

The core of the robot will be a laptop running Linux. The current choice is an Acer TravelMate 6594G which has an Intel Core i7 2.8GHz with 8 GB of DDR3 Memory. Attached to the laptop will be an Xbox Kinect for laser range finding and camera, and an Ardupilot 1.0 for GPS and inertial sensors. The laptop, running the control software, will control the Pioneer base by information gathered from the laser range finder, for path detection, and the GPS, used to stay within the set boundary. The control program will be using the Player [6] libraries which also interfaces with the Gazebo simulation software [7].

4 Workload Distribution and Achievements

The workload was split fairly evenly between the two members of the team. Looking at all of the sensors which were recommended for this project, we started to research possible options. Many of the recommended sensors were very expensive (over $1000 each) and therefore we started to research alternatives. Rudolf researched the laser rangefinders and cameras. Kazu researched GPS receivers and Inertial Sensors as well as possible robot platforms. As budget is always a concern for any student project, we found the most suitable parts in the materials that we already have. For the laser rangefinder and cameras, the Xbox Kinect serves as a low cost solution for this problem. Though the angle of view and the range is much smaller than those of dedicated laser rangefinders (dedicated rangefinders have an viewing angle of 180 degrees where the Kinect is only 57.8 degrees), with a few servos and slower movement, we believe it will suffice for this competition. As for GPS receivers and inertial sensors, again price was factored into its choice. In partnership with UVic AERO, we have decided on using an Ardupilot 1.0 board for its GPS, accelerometer and gyroscope. UVic AERO is moving towards an Ardupilot 2.0 meaning their existing Ardupilot 1.0 is no longer in use.

Along with the sensor packages, the frame and the software also had to be determined. While we were figuring out sensors, the software and the OS were also kept in mind. Because of the sensors we picked, it was possible to just use a laptop as the core of our design. This makes many of the interfacing problems disappear with the proper drivers for a well-supported OS such as Ubuntu Linux. Should the need arise for a smaller core (to reduce weight or size), many of the functions could be easily ported over to because of Ubuntu’s support. Here we reached a major milestone in that all of the sensors as well as the support software for the simulation software to run properly.

After the milestone, Kazu set out to get the simulation software running. Getting the system running with also the robot programming libraries proved more difficult than imagined and as a result was not completed successfully in the time remaining. The simulation software runs but the robot programming libraries were having difficulties interfacing with the simulation software.

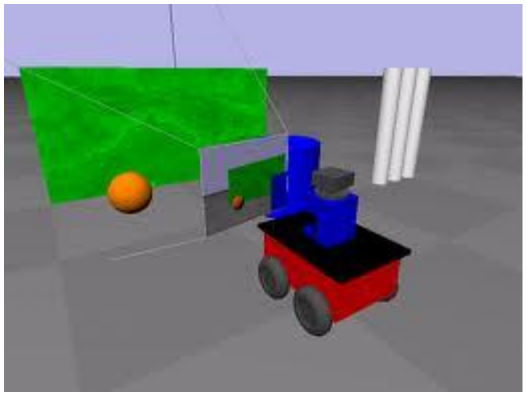

Figure 2: Gazebo Simulation

At the same time, Rudolf began working on getting the Kinect video and depth data back to the computer. A great resource for opening up access to the Kinect censors was found in the OpenKinect initiative [8]. The OpenKinect wiki contains the drivers necessary for interfacing with the Kinect and also a vast user base with existing code for many applications. In the figure below a sample of the Kinect output can be seen. The picture on the right is the video output and the picture on the left is the depth map returned from the Kinect. In the depth map, dark blue represents far distances and bright red represents close distances.

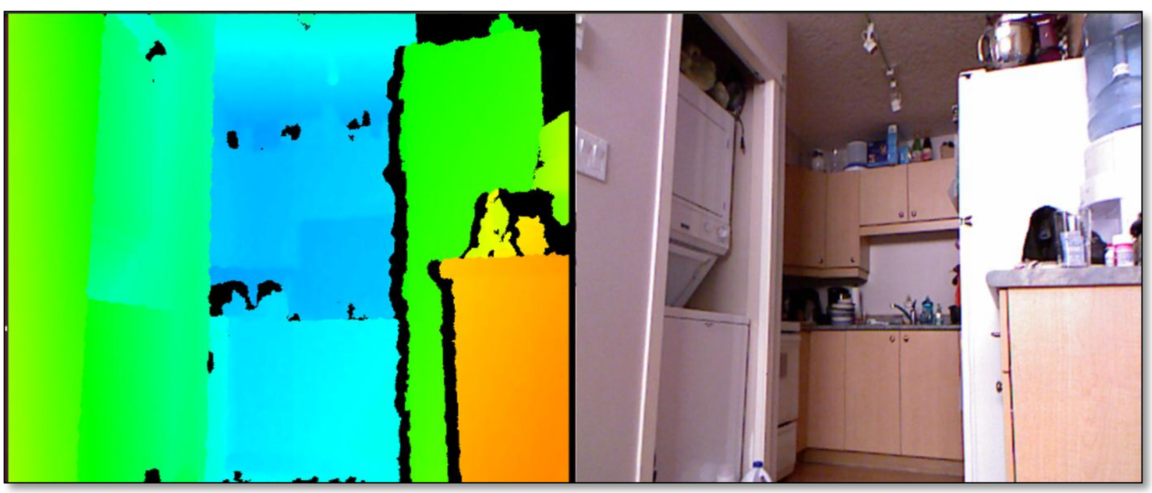

Figure 3 - Depth map and video from Kinect

The next milestone would be to get the control code working within the simulation software as well as processing data from the Kinect. The other sensors would be added till all of them are being used. This will be done in preparation for ELEC/CENG 499.

5 Project Discussion

In this project, there were many compromises that needed to be made in terms of cost and effectiveness. Though it would be nice to create a very high end version of a ground vehicle robot, the cost would prevent us from doing so. Using lower cost options means that we could do a proof of concept design. The sensor interfaces and the use of a Linux platform also make the project very modular. This modularity will allow each successive UGV group to upgrade parts of the project without affecting other parts.

This project would also benefit from a few more people. The image processing and the robot control algorithm each could require a team to properly design and program. The software component is relatively large in this project meaning that it would benefit greatly from a few software engineers.

6 Summary and Future Works

During the ELEC/CENG 399 course, we have managed to determine the major components that would be required for the UGV. Preliminary work on image capture and simulated environment software has also been achieved. By the end of December, both image recognition and simulated environment work will need to be finished as well as gaining financial support for the components that will need to be ordered. We hope to have all required components bought and ordered by this time as well so that in January, for the ELEC/CENG 499 course, we can begin building and testing the actual UGV. By the end of April, the UGV should be fully built and tested so that it is ready for the competition put on by Unmanned Systems Canada in May. After this competition, new designs will be made for the next competition building off of this year’s design. With more financial support, better sensors could be used to improve the overall design of the UGV to further the team in the competition.

Appendix A – Textbook Review

Part Four – Implementation Phase

Moving into implementation

Implementation starts off by the project manager assigning tasks to the programmers. Regular meetings must be held to ensure that all the correct standards are being followed and that the schedule is being adhered to. Scope creep and minor delays are the biggest reasons for not staying on schedule.

When a set of code is finished testing must be performed that examines different parts of the code because the longer a bug goes unfixed the more expensive it is to fix. There are several forms of tests:

- Unit tests which tests modules or programs based off of program specification;
- Integration tests which tests how modules work together based off of interface design, scenarios, DFDs;
- System tests which tests the system as a whole from system design, structure design and other tests; and
- Acceptance tests which are done by users as alpha or beta tests.

Other major part of the implementation to keep in mind is the documentation, which is increasingly done in a digital format. User and system documentation are the two main types of documentation. User documentation is further divided into:

- Reference: Used to find way to perform specific function

- Procedure: Describes how to perform a business task
- Tutorial: Teaches how to use the system

Documentation navigation controls are helpful for finding the appropriate documentation on a specific topic and are recommended for large systems.

Transition to the new system

When moving from an existing system to a new system *Systems Analysis & Design* suggest using the Lewin's three-step model: unfreeze, move, and refreeze. Unfreeze involves detaching from the existing system. A migration plan is then followed and once ready support and maintenance of the new system helps refreeze the new system into place.

The migration plan, which sets the procedure between the unfreeze and freeze stages, is comprised of many strategies all involving getting different parts of the organization is prepared for the new system. The move must not only be prepared but also the acceptance of the new system. Management policies, motivation techniques and training, and understanding the sources of resistance will help with the adoption of the new system and the migration plan.

There are many activities that need to be performed after the new system has been implemented. Support staff must handle question about the new system and, depending on their experience, help determine customer problems and generate bug fix requests if needed. Maintenance staff fix bugs and improve shortcomings of the system through assessments which can then be used in the next system implementation. The project team reviews team activities and generates documentation on any lessons learned.

Management determines the costs and benefits of the new system and how they match with the predictions when the project was initiated so that future predictions are improved.

The movement to objects

Object oriented programming techniques have become very important due to way in which objects (person, place, or thing) can be specified. The object-oriented approach helps minimize the development costs which are often much more than the computers used. Each object has attributes which describe the object and methods which specify what the object can do. Objects that share common attributes are grouped into classes. Classes can also be arranged in a hierarchical tree based on common subclasses and attributes below, and methods in superclasses above. The UML standard allows complex problems to be deconstructed into smaller components using a set notation that everyone can understand.

Works Cited

[1] "Student Competition," 25 November 2011. [Online]. Available: http://www.unmannedsystems.ca/content.php?doc=123. [Accessed 27 May 2012].

[2] 2012. [Online]. Available: http://auvic.uvic.ca/. [Accessed 27 May 2012].

[3] 2012. [Online]. Available: http://www.aero.uvic.ca. [Accessed 27 May 2012].

[4] 14 May 2012. [Online]. Available: http://www.unmannedsystems.ca/index.php. [Accessed 27 May 2012].

[5] "Adept Mobilerobots," [Online]. Available: http://www.mobilerobots.com/ResearchRobots/P3AT.aspx.

[6] W. R. Dennis, Systems Analysis & Design, Wiley, 2012.

[7] "Limitations of the Kinect," 17 December 2010. [Online]. Available: http://www.iheartrobotics.com/2010/12/limitations-of-kinect.html. [Accessed 24 June 2012].

[8] "SICK LMS 291 Laser RangeFinder," 14 02 2009. [Online]. Available: http://www.hizook.com/projects/sick-lms-291-laser-rangefinder-lidar. [Accessed 20 06 2012].

[9] "The Player Project," [Online]. Available: http://playerstage.sourceforge.net/.

[10] "Gazebo," [Online]. Available: http://gazebosim.org.

[11] "Orca," Orca Robotics, [Online]. Available: http://orca-robotics.sourceforge.net/.

[12] "Miro," [Online]. Available: http://miro-middleware.berlios.de/.

[13] "Yarp," [Online]. Available: http://eris.liralab.it/yarp/.

[14] "ROS," [Online]. Available: http://www.ros.org/wiki/.

[15] "Clearpath Robotics," [Online]. Available: http://www.clearpathrobotics.com/husky.

[16] "Eclipse Project," [Online]. Available: http://www.eclipse.org.

[17] "DIY Drones," [Online]. Available: http://diydrones.com/profiles/blogs/ardupilot-main-page.

[18] "Sokkia," [Online]. Available: http://www.sokkia.com.

[19] "MicroStrain," [Online]. Available: http://www.microstrain.com/inertial/3DM-GX1.

[20] "MTEK GPS3329," [Online]. Available: store.jdrones.com/jD_MediaTEK_GPS_p/jdmtek01.htm. [Accessed 02 07 2012].

ELEC/CENG 399 Design Project I Final Project Report Evaluation Form

To be filled by students:

Project title:____

Group #:____

Group members:____

Supervisor(s):____

To be filled by the supervisor(s):		
Progress report distributed to the supervisors for grading: Friday, August 3, 2012		
Please complete the progress report grading by: **Friday, August 17, 2012**		
Please refer to the rubric for grading.		
Topics		Grade [%]
[5%]Chapter 1, Goals :		
[10%]Chapter 2, Progress Overview:		
[25%]Chapter 3, Detailed Project Description:		
[25%]Chapter 4, Workload Distribution and Achievements:		
[10%]Chapter 5, Project Discussion:		
[5%]Chapter 6, Summary and Future Works:		
Subtotal [80%]:		**0.0**
To be filled by the instructor:		
[20%]Appendix A: Textbook Review	[10%]Write the textbook review in a clear	
	[10%]Meet minimum page requirement (2 pages):	
	Subtotal [20%]:	**0.0**
Total [100%]:		**0.0**

Comments by the supervisor: