AC 2012-4343: SYSTEMS ENGINEERING EDUCATION THROUGH PARTICIPATION IN ENGINEERING COMPETITIONS

Dr. Fernando Garcia Gonzalez, Texas A&M International University

Fernando Gonzalez is an Assistant Professor of engineering at Texas A&M International University in Laredo, Texas. Previously, he was a technical staff member at Los Alamos National Laboratory and an Assistant Professor at the University of Central Florida in Orlando, Fla. Gonzalez holds a Ph.D. in electrical engineering from the University of Illinois, Urbana-Champaign. His research interests include intelligent control of autonomous systems, robotics, and modeling and simulation.

Systems Engineering Education through Participation in Engineering Competitions

There are many student competitions involving complex engineering designs and in particular involving robotics or robotic vehicles. In the Department of Defense's Intelligent Ground Vehicle Competition, IGVC, Students create autonomous vehicles that navigate an obstacle course using only its own systems with no human interaction. This results in the vehicle having very complex and sophisticated engineering systems including vision, a laser range finder, global positioning systems and many other sensory devises. It also involves very sophisticated software to perform the autonomous navigation using the sensor inputs. The complexity of the project necessitates the involvement of a relatively large group of students working together on different parts of the system.

The Robotics Laboratory at the University of Central Florida has been participating in the annual IGVC competition since 2002. In this project the students learn about most of the engineering disciplines that are typically included in a complex robotic project such as software design, computer vision, sensor data interpretation and fusion, robotic motion planning, vehicle navigation, vehicle design and construction, electric motor control, computer interfaces to various components and many others. Naturally this education compliments the core engineering education they receive during their course education. However one very important engineering discipline that they learn is systems engineering. The success of the project involves a strong systems engineering effort to integrate all the individual components, to design the overall system, to consider the complete life cycle, and to coordinate and oversee the project and its team members. The team generally shares the systems engineering responsibility as they are all involved in the higher level decisions that need to be made. This exposes all of the students in the team to systems engineering education which is especially beneficial when you consider that most engineering programs still do not include systems engineering.

Presented in this paper is how a student robotics competition involving the design and construction of a complex autonomous vehicle effectively gives the student team members real life systems engineering experience.

Introduction

As the need and importance of systems engineering becomes more apparent many engineering departments are creating programs in systems engineering. However they are still relatively rare especially at the undergraduate level. There are about 75 institutions that offer systems engineering programs in the US of which about 43 of them are available at the undergraduate level[1]. It is important that all engineers get some experience in systems engineering during their undergraduate education even if they are not targeting a systems engineering job. According to Satinderpaul et. al.[1] it takes about 10 to 15 years of hands-on experience before the systems engineer graduate can take on a lead role as a systems engineer. Under the NASA Exploration System Mission Directorate (ESMD) there is a program to introduce NASA engineering into the senior design courses at our nation's undergraduate institutions called the NASA Exploration Senior Design Projects[2]. In this program system engineering is highly emphasized and in their corresponding faculty workshop, they show how a lack of systems engineering is the main cause of many of NASA large and expensive engineering disasters. They even have the 2011 ESMD Space Grant Systems Engineering Paper Competition[3].

In addition to introducing systems engineering in the senior design course another way to introduce systems engineering to non-systems engineering majors is to add relevant courses into the curriculum however this is difficult since it will require replacing other courses. A simple way to give the students the systems engineering experience they will need is through the participation in one of the many competitions that are available to engineering students. The focus of this paper is not to imply that participation in engineering competitions will be a substitute to having a systems engineering program nor is it implied that this experience will be in any way complete. The focus is for engineering students that are not in systems engineering to learn some of the systems engineering skills and concepts that they otherwise will not get.

In this paper the focus will be on the IGCV competition because I have direct experience with that particular competition and the complexity of the vehicles required to perform in this competition make it a good fit to gain systems engineering experience. There are however many different types of competitions that require a complex system in which the students will gain some systems engineering experience. In Flint[4] they present 50 competitions just dealing with miniature autonomous vehicles.

The Robotics Laboratory at the University of Central Florida

While a faculty member at the University of Central Florida's Computer Engineering Program, in 2002 I created the Robotics Laboratory at the University of Central Florida[5] with the focus of building an autonomous vehicle to compete in the annual Intelligent Ground Vehicle Competition (IGVC)[6]. Throughout the years this program grew to include ground, underwater, surface, and aerial vehicles for different competitions. By the time I left it had 18 volunteer graduate and undergraduate students and was producing a highly sophisticated vehicle at a rate of one every six months.

Now celebrating their 10th anniversary the lab has recently won 2nd place at the IGVC among many other impressive rewards. Currently the lab is managed by a former student that started many years ago as an undergraduate student and is now a faculty member at the Institute for Simulation and Training (IST).

Each team is supervised by a student team member that takes on the role as team leader. This reduces the work load of the faculty advisor and gives the lead students an opportunity to gain managerial experience. The lab has around 5 teams at any given time. Furthermore one of the students was made responsible for the overall operation of the lab. This student supervises the team leaders and also interacts with any sponsor. All the students however participate in the actual development of the vehicles. The faculty advisor plays a relatively small role once the student managers were in place and the lab is fully operational. This took a few years to achieve but was well worth the time and effort.

The Intelligent Ground Vehicle Competition

The IGVC competition started as a way to increase participation in autonomous vehicle research at the undergraduate level. Today it has many sponsors and is in its 19th year. The first competition was in 1993 and had 7 teams. In 2011 they had 40 teams with many universities, including ours, bringing multiple teams to the competition. The competition offers the opportunity for students to gain experience in many areas of engineering including systems engineering. The following quote from the IGVC website summarizes this nicely.

> “The IGVC offers a design experience that is at the very cutting edge of engineering education. It is multidisciplinary, theory-based, hands-on, team implemented, outcome assessed, and based on product realization.” – IGVC website[6].

The competition consists of having vehicles run autonomously though an obstacle course. Since the vehicles must run with no human intervention, they require the use of technology such as vision and range finders to allow them to “see” and navigate through its environment. Global Positioning Systems (GPS) allows them to navigate to pre-specified way points. The use of other sensors such as touch, directional, speed etc. is also needed. The competition generally consists of 3 challenges, navigation through a path, navigation via waypoints and a design competition. The course for the navigation challenge consists of a path painted with white paint on the grass. See Figure 1.

Figure 1: The Autonomous challenge

The lines identifying the path have gaps that the vehicle needs to realize and there are large barrels in the way as well. The vehicles cannot touch any barrels. Although the navigation is timed, just making it to the end is considered a great success. The Navigation challenge consists of a list of Global Positioning System (GPS) waypoints the vehicle must visit. There are barrels and fences in the environment the vehicles must avoid. Sometimes a fence with a small entrance surrounds the way point and the vehicle needs to find the entrance. This challenge is timed as well and visiting all waypoints without touching an obstacle is quite a challenge. In the design challenge, see Figure 2, teams compete for the best design and the best paper which must include the existence of an engineering process they must follow.

Figure 2: The Design Challenge

The Autonomous Vehicles

The vehicles the lab produced were very sophisticated and included many different sensors, several computers and very complex software. This level of complexity is necessary for the vehicle to perform at the competition.

Our first vehicle, Black Knight[7], see Figure 3 is built on a Pride Mobility[8] scooter that was donated by Pride. It uses four cameras to gather information about the world around it. A path-planning algorithm is used for navigation on a map created from sensor input. An optical sensor, an electronic compass, digital GPS, and two magnetic encoders supply speed and position information to Black Knight. Figure 4 shows the software architecture that was spread over 2 computers. The vision system used a neural network system that after training could classify the parts of the image into areas that are grass, white lines, orange barrels, and other obstacles.

Data from all of these sensors allow Black Knight to avoid obstacles and navigate through varied terrain. It had two onboard computers that communicated with each other. The steering tiller was cut off and an electric motor was connected to the remaining shaft. A microprocessor controls the motor to steer the vehicle. The vision system uses the information from the 4 cameras to see the environment around it. A robotic path planning system was created using existing robot motion planning techniques. See Figure 5. A steel structure was built on top of the scooter platform and fiberglass was used to build a cover for the vehicle. The design paper and Gonzalez et. al[7]. highlights Black Knight's electronic, mechanical, computer, vision and software systems and included information on the design process, team and vehicle organization.

Figure 3: The Black Knight vehicle

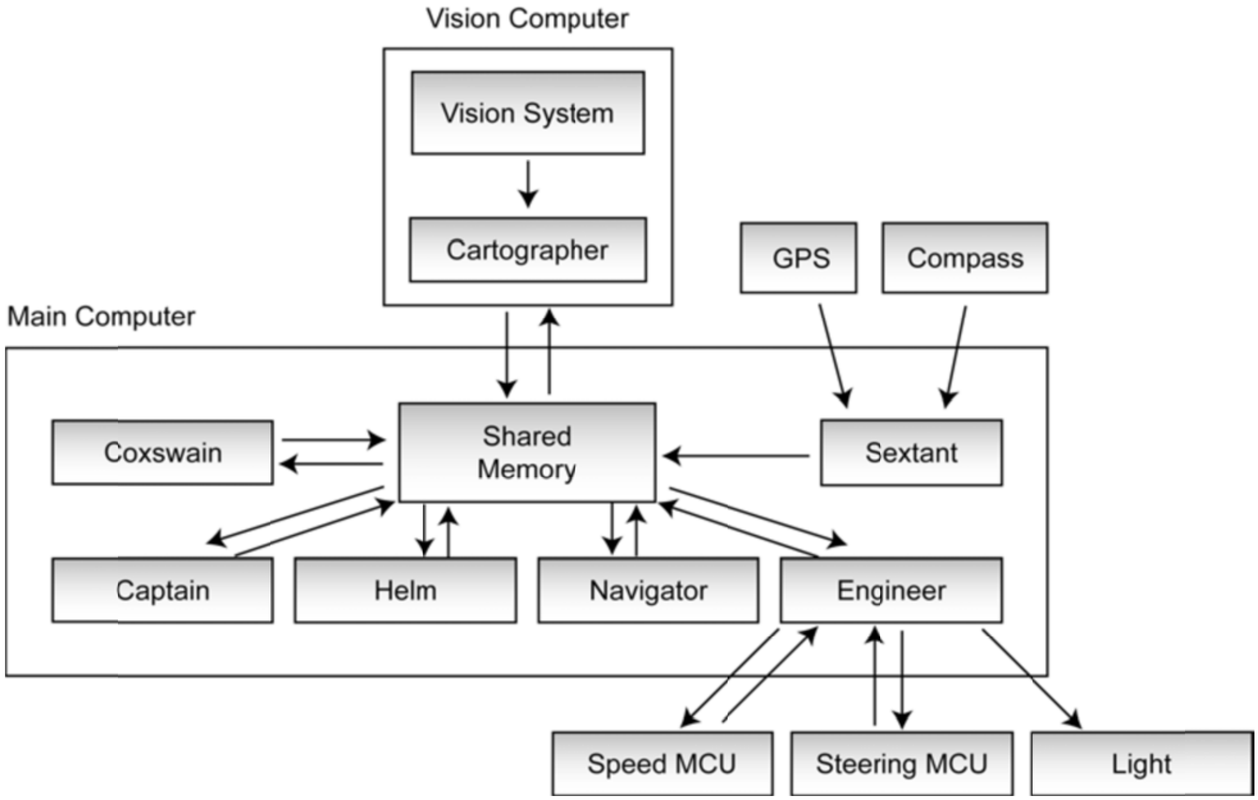

Figure 4: Black knight's software architecture

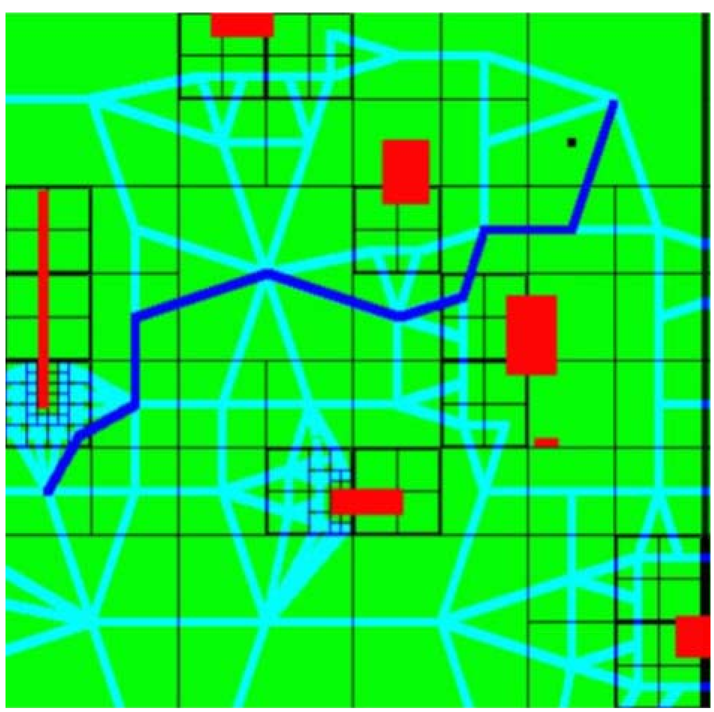

Figure 5: The path planning algorithm

Our second vehicle, Calculon[9], see Figure 6 was built using an electric wheelchair. Calculon's designed was to achieve an intelligent and able robotic platform which leverages an incremental design process, modular software design and use of commercial off-the-shelf products. The use of the wheelchair gave the vehicle a zero turning radius that we found was necessary considering the course it needed to travel. We added a SICK LMS Laser Range Finder which gives very precise location of obstacles. We changed the 1CCD to a 3CCD camera with higher color and pixel resolution that is better suited for the varied lighting conditions found at the course. The development of a rapid vision prototyping system called Discover Vision started with this vehicle and continued throughout the years. Figure 7 shows Calculon's electronic map including a list of devices attached and their location. Figure 8 is a diagram showing the system integration and the AutoCAD design is shown in Figure 9.

Figure 6: Calculon

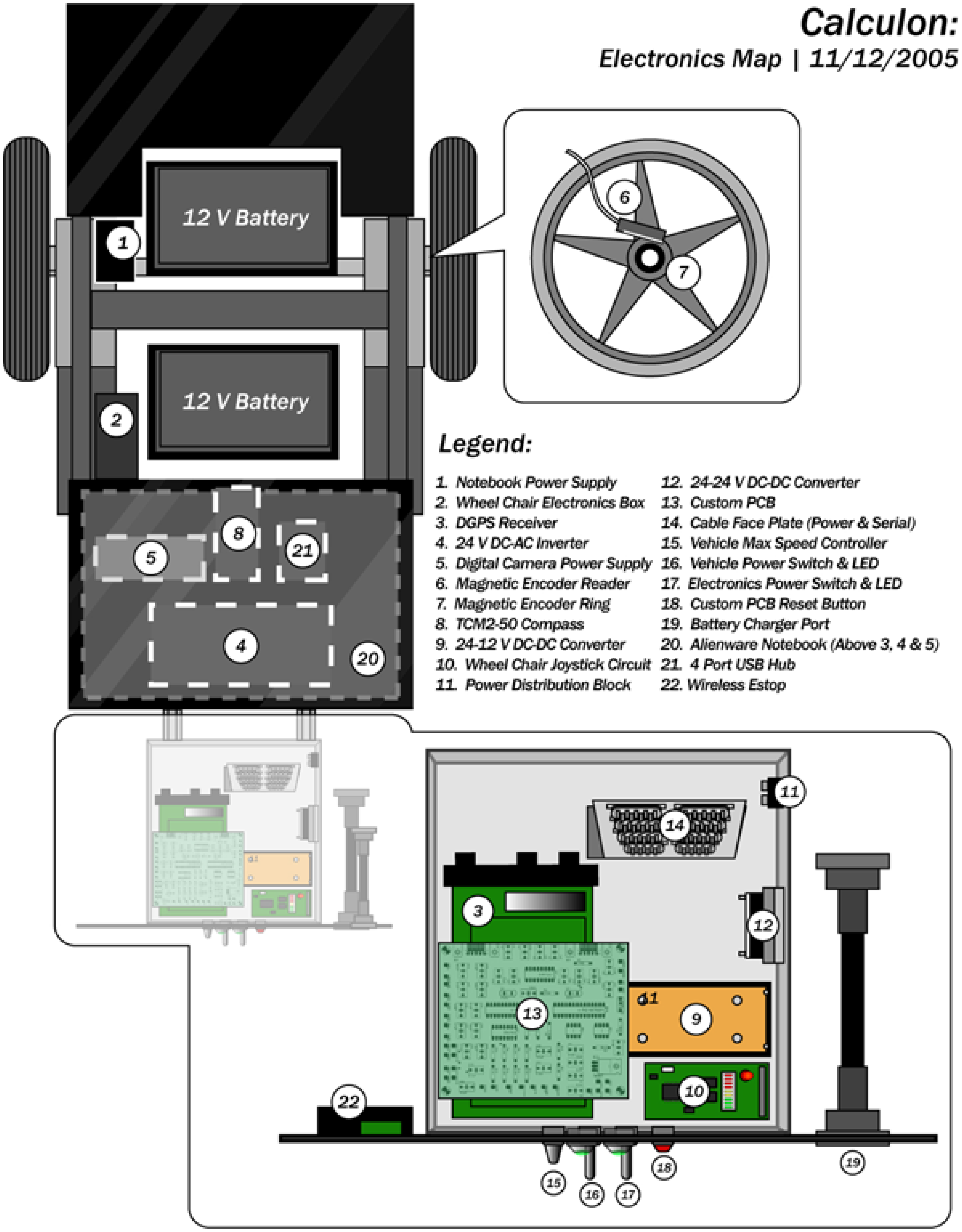

Figure 7: Calculon's electronics map

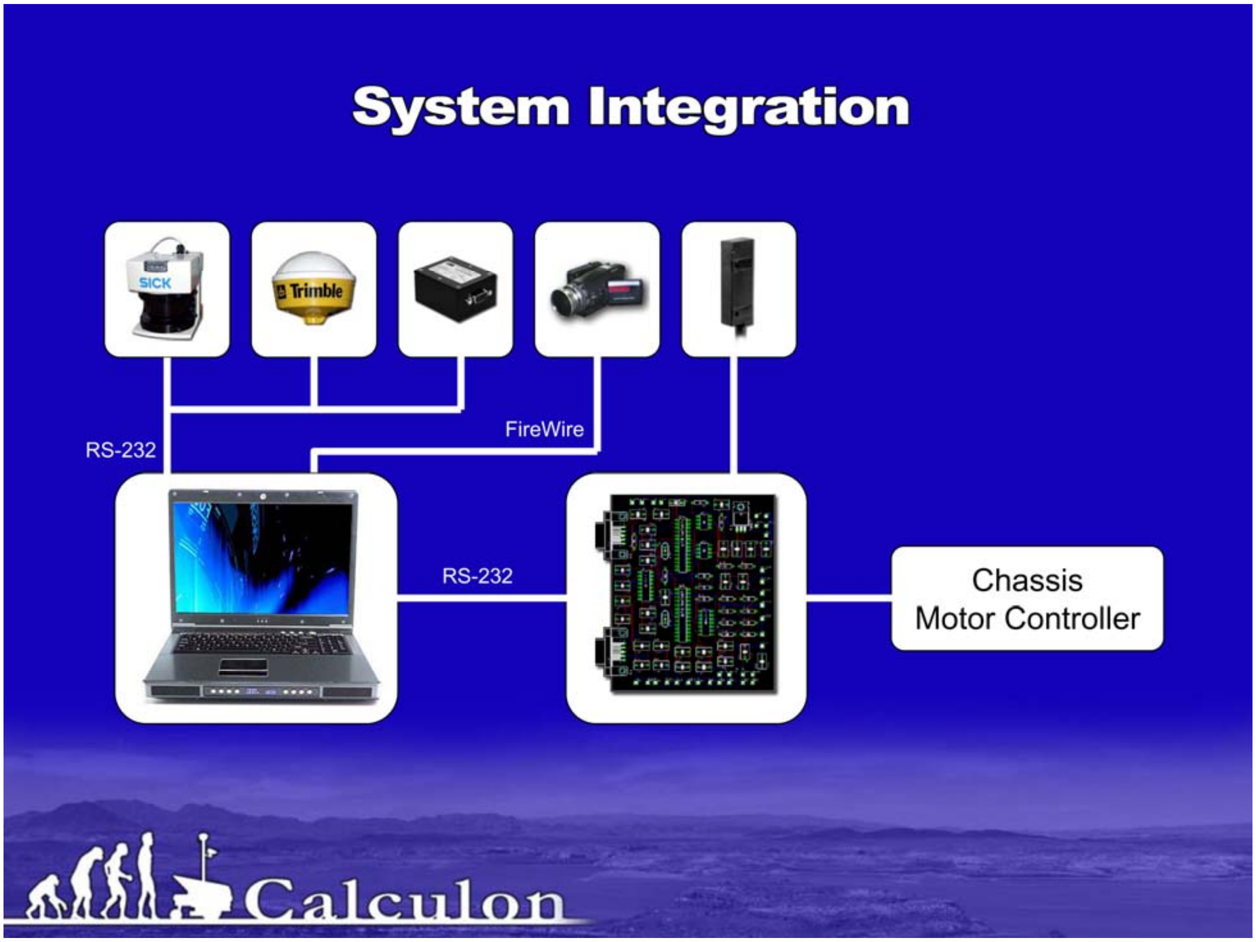

Figure 8: Calculon's system integration

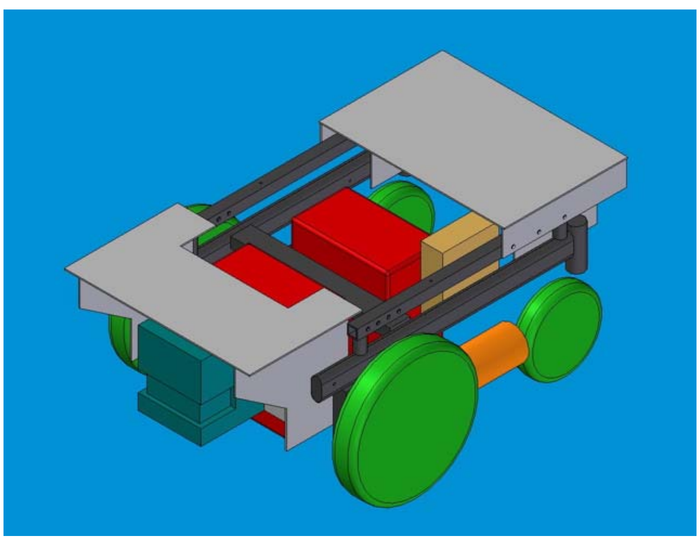

Figure 9: Calculon's AutoCAD design.

The Systems Engineering Experience

The complexity of the vehicle system necessitates the use of systems engineering to integrate all the individual components, to design the overall system, to consider the complete life cycle, and to coordinate and oversee the project and its team members. The experience gained is self-regulated in that if the team fails the systems engineering, the complexity of the vehicle is such that they will most likely perform poorly at the competition. Winning or performing well at the completion is what motivates the students to implement solid systems engineering principles. For example an inconsistency between the units used in two separate systems that interact will most likely cause the system not to perform the way it was designed. This was the cause of the first space shuttle accident. The system must be complex for this self-regulation concept to work.

The management hierarch developed for the lab gives the students systems engineering experience related to managing people in different disciplines. Since the vehicles involve mechanical systems, software, computer systems, vision systems, and electrical systems the teams must and in fact did include people from these different disciplines. No one person had experience in all these areas and therefore the leaders had to learn to manage the members of the team and coordinate their parts even though they are unfamiliar with their disciplines.

Every year the team made major improvements to the vehicle. This is actually a requirement for the competition. Usually this involved changes to the software and sensors. Every few years they also changed the vehicle platform and built a totally new vehicle. By redesigning components for each competition the team realized that if they consider the entire life cycle of their projects they could save a significant amount of effort. In the design of each vehicle system they considered how that effort can be used in the design of that system in following years. For example the vision system they built included a rapid prototyping system, Discover Vision, see Figure 10 that allows them to test new vision technologies quickly. They even created their own scripting language so that a vision script created by Discover Vision can be directly used in their future vision systems. They justified the effort for this system by knowing it will be reused many times.

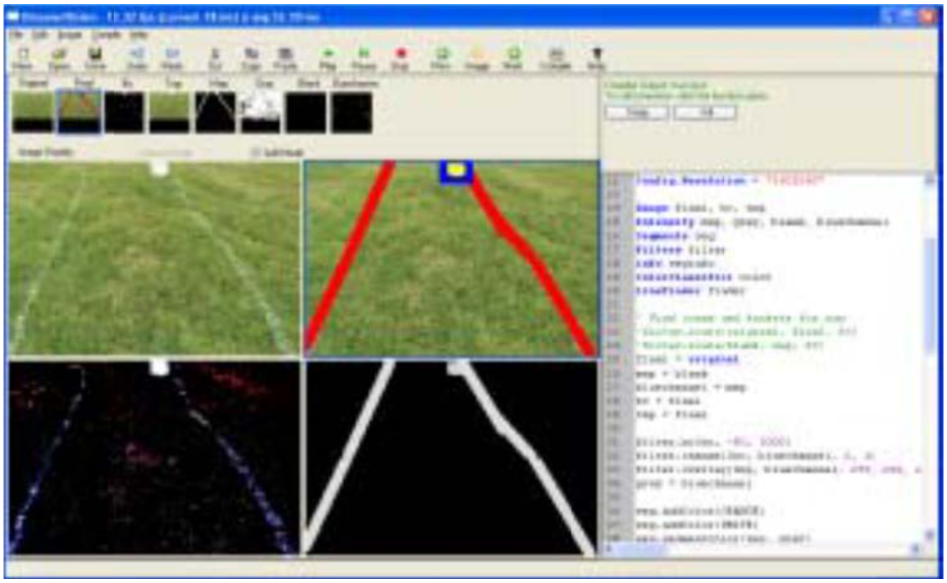

Figure 10: The Discover Vision window

The IGVC competition mandates that the team use some formal design paradigm. In the 2004 design we used the Bernd Bruegge and Allen H. Dutoit[9] Design Life Cycle Model. Later in 2005 we used an incremental design and development process, which allows for more flexibility when needed as well as for parallel development from various sub-groups which was essential to the team. Our incremental design flow followed many sequential ‘Validation V’ designs which then gave it a ‘W’ formation which represents the synchronization of multiple teams working together. The team found that the ‘W’ paradigm works exceedingly well for the level of modularity they were using. Design paradigms are generally taught in systems engineering courses however there is no better way for the student to learn than from real world experience like this competition. They were not looking at it from a theoretical point of view but rather from a practical and useful point of view. And they were not just learning the theory to pass a test but rather to achieve the development of a complex system. They knew that a poorly implemented design paradigm or the selection of the inappropriate paradigm will result in more overall development effort and possibly a poorly operating vehicle.

Documentation is critical when working on a large project whose development spans many years and involves many people. The vehicles mostly share software from year to year and so there is always the issue of bringing new members up to speed on the development. This applies to new members entering the project as well as to the older members that may have not worked on it for some time. I noticed that the documentation for the code as well as for the overall system kept getting better, more thorough and more organized with time. The teams realized the importance of good documentation and decided to put in place documentation standards that all team members must follow. In a programming class the students are required to include documentation in all programs however the students never really understand the importance of the documentation because their programs are generally too small and trivial, are created by a single person and are not maintained after they are turned in.

While the vehicles are complex enough to require systems engineering they are not so complex that the team members will only work in their discipline. The team members generally work in their discipline but also spend a lot of time interacting and helping the other team members. So for example the mechanical engineering student learns about software design by being involved with how the software interfaces with his mechanical design. The software programmer works with the computer vision students to produce the vision software and learns about computer vision in the process. This is very essential for today’s engineer.

While it may take several competition cycles for the team to realize many of the benefits implementing systems engineering principles may have, it’s not required for new entering team members to go through the same discovery phase. The team members that learned the value of systems engineering from experience will dictate to the new incoming team members the systems engineering rules and standards they have put in place. This may include such things as documentation standards, integration protocols, and adherence to a specific design paradigm they may have in place. While the incoming students initially do not learn by experience they do hear the need from existing team members and know that they are implementing these rules because of need and not solely for educational purposes. They trust that they are in fact needed since the rules are coming from their team members and not from an educator. Eventually as they work on the projects they realize why such rules are in place.

Finally the team members become accustom to using the systems engineering standards they have in place. For example they become accustomed to writing documentation on all the software they produce, on following a predetermined design paradigm and even to consider the entire life cycle. Because the time they spend on the project is probably the most hands-on work they will have done during their education, their experience will consists of mostly good habits. These habits will remain afterwards and become a standard by which they work.

The outcome of this learning through competition effort was not directly measured as this was a byproduct and not the intension of the project. It is however very obvious that the students learned a great amount of systems engineering by observing how the team improved in the competition over the years. The students have the option of using the part of the project they are assigned to as their senior design project however much to my surprise only a small number of students took advantage of this. Graduate students were able to perform research for the project and have it count towards their master’s thesis credits and project however this was uncommon as well.

Starting a Competition Project

Starting a lab for a competition requires some initial funding. We started out with $15,000 funded by the Navy and the donation of the scooter by Pride Mobility. The funds were exclusively used for material and travel to the competition. I had some previously allocated lab space that I used and it was just large enough to fit the vehicle and some students to work on it. While we did not qualify to enter the competition in the first year we were able to demonstrate the vehicle to the sponsor which then funded us at $25,000 the following year. In later years we received $50,000 per year from the Institute of Simulation and Training which also provided us with a large lab space. Since the equipment is reused from year to year and the labor is volunteer the funding we received in the later years allowed us to purchase new and better equipment, allowed for the funding of other competitions like the surface (boat), air and water (submarine) competitions and allowed us to purchase manufacturing equipment like a milling machine and a welder.

To start with less funding one can look for local competitions where the travel cost will be less. Also simpler competitions do not require as much equipment and can also result in less startup funding necessary. Finding used equipment that can be donated like the wheelchair we used for Calculon also reduces the need to such funding levels.

At the beginning the students needed a lot of support. The management hierarchy was not well defined and there were no students that had the experience to become an effective the team leader. The students did not receive any special or formal training in project or team management however they do get teamwork training in their senior design courses which some students would have had by the time they become a team leader. All student conflicts are handled by the team leaders and the lab leader and never made its way up to me. While the students work very hard at running the projects and the lab, I also give them a very high level of authority and flexibility in the way they wish to run the lab. Therefore the students running the lab have authority to handle members they see as needing conflict resolution. All the team members know that the leaders of the lab have authority to remove any student from the project and that it will be unlikely that I will override their decision. Therefore they learn that participation is a privilege they need to earn and can lose. As the faculty in charge of the lab, I had to take certain risks at giving this much authority to students however the students have so much time and effort invested and are so motivated that they will not abuse this authority and it was very apparent they they only wanted the best for the project.

From a technical point of view the students did not have the technical expertise. The demand on the faculty is high for about the first 2 competition cycles. While I had a good group of students willing to work hard to create this lab I still found myself giving the students technical knowledge. I offered a project type course on autonomous vehicles which I used to get the project moving and to recruit more students. After the class about 20% remained with the project. All it takes is a few really sharp and hardworking students to push the project forward. The good news is that in the second year the team needed a lot less support from me. In the third year I was almost not needed other than to give technical advice and pointing the student in the right direction. After that the team had a well-defined and strong management hierarchy, had the technical skills they needed and were experienced enough that they could move the lab forward even going beyond my own technical knowledge to the level of publishable research for example as in[10]. They became an expert in every area they needed. They became experts at computer vision, path planning, software engineering and they even became an expert with the electronics which is impressive considering there were no electrical engineering students on the team. At this time the role of the faculty advisor is minimal and the lab runs entirely by the students themselves.

Throughout the development of the lab the students participated in securing the funding and lab space we needed, recruiting new student, training the new students, teaching themselves the latest technologies, and dealing with all of the issues that arose. It was truly a joint effort between the students and me. On the down side, it will be very difficult to pursue such a venture without the help of some very motivated students willing to put in lots of time and effort. A graduate program will allow the students to remain participating in the lab past their graduation and it's these students that generally become the team leaders and that run the lab.

For the systems engineering experience, as a faculty advisor one can teach them the systems engineering they will need to know. While they will not learn it or even believe it at first, it will set them up to be prepared for when they realize they need it. They will at least know what the theory is. Then with time they will realize the need and begin to implement systems engineering principles little by little. After a few competition cycles they will have the system engineering in place and they would have learned it very well from experience. They will teach the new incoming students what they need. So like the rest of the lab it just takes some time at the beginning to get things moving and it will then run by itself.

Conclusions

The participation in an engineering competition results in the students gaining hands-on systems engineering experience that they generally do not get in a classroom unless they are actually in such a program. They learn some relevant systems engineering theory and its importance not by studying or listening to a lecture but by experiencing it firsthand. In a sense they learn the hard way but I am sure it will stick much longer. In reality it's a by-product as the students set out to participate out of their desire to learn more engineering in the discipline and have fun with the competition. The systems engineering is added to the other engineering concepts they also learn.

The competition concept is especially useful if there is no systems engineering courses or program in place. The time the students spend on this project is not taken from their curriculum but rather from their own spare time. The advantages to the students including all they learned in the discipline, plus all the systems engineering they learned and the experience they can put in their resume is well worth the time they invest. And the students know this and are what motivates them to recruit more students.

References

1. Satinderpaul Devgan, Saleh Zein-Sabatto, Sachin Shetty, 2010, "Integrating Systems Engineering in Main Stream Engineering Disciplines," Proceedings of the 2010 ASEE Annual Conference & Exposition.
2. http://education.ksc.nasa.gov/esmdspacegrant/Faculty_Training.htm
3. http://education.ksc.nasa.gov/esmdspacegrant/SystemsEngineering.htm
4. Jason Flint, Shawn Cheney, Blake Hale, C. Richard G. Helps, 2003, "Analysis of Competitions for Mechatronic Design Using Autonomous Mobile Robots," Proceedings of the 2003 ASEE Annual Conference & Exposition.
5. http://robotics.ucf.edu/
6. http://www.igvc.org/
7. F. G. Gonzalez, R. Andres, D. Deal, F. Goergen, M. Rhodes, T. Roberts, G. Stein, J. Wilson, S. Wong, 2004, "Black Knight: An autonomous vehicle for competition," Journal of Robotic Systems, Vol. 21 No. 9, pp. 451-460, September 2004.
8. http://www.pridemobility.com/
9. F. G. Gonzalez, T. Roberts, D. Barber, B. Becker, A. Kiriwas, 2005, "Calculon: An Autonomous Ground Vehicle for the 13th Annual Intelligent Ground Vehicle Competition," Proceedings of the 5th Annual Intelligent Vehicle Systems Symposium & Exhibition.
10. Bernd Bruegge, Allen H. Dutoit, 2000, "Object-Oriented Software Engineering: Conquering Complex and Changing Systems," Prentice Hall,
11. F. G. Gonzalez, T. Roberts, D. Barber, B. Becker, A. Kiriwas, 2005, "Software Design for an Autonomous Ground Vehicle for the 13th Annual Intelligent Ground Vehicle Competition," SPIE - The International Society for Optical Engineering, Intelligent Robots and Computer Vision: Algorithms, Techniques and Active Vision (SA116).