Search and Rescue Robots for Integrated Research and Education in Cyber-Physical Systems

Orion Lawlor*, Mike Moss*, Steven Kibler**, Charles Carson*, Shaun Bond*
Seta Bogosyan**

*Computer Science Department, **Electrical and Computer Engineering Department ,
University of Alaska Fairbanks, USA

Abstract—**This paper presents a novel research-education integration approach that uses the interdisciplinary theme of Cyber-Physical Systems (CPS) to further advance teaching and learning in such systems and related areas, in electrical and computer engineering (ECE) and computer science (CS). The primary targeted learners in this emerging area are graduate and undergraduate students of ECE and CS, as well as teachers and students of STEM in high schools, who work together over a common CPS design project, with the graduate and undergraduate students teaching and mentoring the high school students. In the first year of this GK12 project, a networked ground and aerial vehicle based Search and Rescue(S&R) Robotics system is selected as the CPS design project with an associated multi-school competition organized via the use of teleoperation over the internet. The first year results demonstrated great promise not only in terms of improved learning among students and teachers of multiple disciplines and levels, but also provided valuable hardware and software platforms for on-line experimentation and education in CPS, robotics, graphics and animation, UGV and UAV control, communication and networking.**

Keywords—CPS, UAV/UGV control, search and rescue robots, quadricopter, remote experimentation, remote education, GK12.

I. INTRODUCTION

The amazing advancements in technology over the past decades have motivated studies and projects seeking novel and more effective ways of education in different areas of engineering [1][2][3][4][5][6] to prepare the future workforce. This paper presents the first year outcomes of one such project, namely NSF Cyberalaska GK12 grant, aiming for integrated research and education in Cyber-Physical Systems (CPS).

GK12 projects require graduate students (or, so-called Fellows) conducting research in an area related to the general theme of a given GK12 project, i.e. climate change, wireless sensor networks, renewable energy systems etc., to bring their research to K-12 classrooms as teachers and mentors. GK12 grants offer more than basic, randomly implemented outreach, as the program requires graduate research to be brought to classrooms via well-planned, systematic course modules, all prepared and presented by the graduate students in classrooms up to 10 hours a week. The ultimate aim is to increase graduate students' knowledge in their research area via teaching related material, thereby also improving their communication and overall professional skills. Through these activities, it is also aimed to create increased motivation and understanding of science, technology, engineering, and math (STEM) for the K-12 students and teachers. The program performance is externally evaluated every year for the fulfillment of its objectives.

Different from other GK12 activities, CyberAlaska also involves undergraduate students from multiple disciplines as Fellows in this integrated research and education process, only with less number of hours in STEM classrooms (3-4 hours weekly). Our outreach approach is also different from most other similar efforts, as we involve the outreached STEM students not only as passive learners and users of technology created by Fellow research, but as *apprentices* to Fellows, and as actual, active *contributors* in all feasible aspects of a theme- related (CPS, in this case) annual design project, in a level-appropriate and systematic manner.

Another novelty of the presented work is the use of Cyber-Physical Systems (CPS), as the general theme to promote integrated research and education, and to cover a wide range of disciplines and concepts relevant to electrical and computer engineering and computer science. While there are a few recent studies on the use of CPS technologies for education [7], CyberAlaska is the first known effort to provide integrated research and education in CPS to graduate and undergraduate students, and is certainly the first effort to infuse CPS concepts into high school STEM curriculum.

Cyber-Physical Systems (CPS), recognized by NSF as a priority area of the future, represent a new generation of systems that integrate computing, control, and communication capabilities with the dynamics of physical and engineered systems. CPS has a wide range of applications, which are all very active areas of research. Providing citations only for application fields relevant to this paper due to space limitation, one could mention intelligent transportation systems, smart bulidings, smart grids, collision avoidance systems, telesurgery/telerobotics [8], coordinated unmanned aerial and ground vehicles [UAVs and UGVs], search and rescue robotic systems [9],[10],[11],[12] etc. as examples of CPS.

In the first year of the project, a networked UAV and UGV based Search & Rescue (S&R) system was selected as the annual CPS design project, around which Fellows' educational activities would be planned. This involved classroom teaching of CPS concepts and components, supported by hands-on educational modules, and conveyed in parallel to the project component being developed in class. In accordance with the theme, the first annual CyberAlaska high school challenge was also set to be an "S&R system searching for lost hikers in Alaska". The system should also be teleoperable over the web to allow for the participation of high school students from 1500 miles away, whom the Fellows had mentored throughout the year,

With the design of the S&R system, it was aimed to familiarize all the participating students (graduate, under-graduate, high school) and teachers with the highly interdisciplinary and inter-weaved concepts of CPS through a real-life example, which requires the use of electrical, computer, control, communication engineering concepts as well as hardware design and reliable software development for remote access. Due to the increasing popularity of remote labs [13], and their relevance for the needs of remote education in rural schools of Alaska, a web-accessible UAV-UGV robotic team would also be a valuable asset for the sustainability of our activities after the life of the grant.

This paper describes the in-class development process of the S&R system and the first-year outcomes of CyberAlaska activities, and is organized as follows: Section I gives a general introduction; Section II describes the development process of the UAV control system and its CPS related components; Section III presents the development and control of the UGV; Section IV describes the networking , and Sections V and VI discuss results and conclusions.

II. Development of networked air and ground search and rescue robot system

In the first year, CyberAlaska graduate and undergraduate Fellows from UAF ECE and CS departments initially collaborated in building a web-accessible S&R system themselves to gain experience with such systems. They also worked with the participating teachers and faculty advisors to prepare CPS relevant lesson plans and course modules that would accompany each step of system design and development during the academic year.

With the start of the academic year, they collectively guided several high school student teams in two classrooms (one in town, one remotely) in the development of S&R systems in class.

The developed S&R system, demonstrated in Fig.1, consists of a semi-autonomous UAV (quadricopter) and UGV pair performing a search and rescue mission indoors, with some level of coordination. Both vehicles are run autonomously with a teleoperator having the option to command position references to both vehicles using the video feed supplied by the UAV cameras, as well as a web interface developed by the team, both provided on a web page. The developed web interface has the capability to demonstrate current position and allows the operator to enter reference vehicle positions by clicking on the screen. The teleoperator monitors the operation, commands positions, and interrupts all operation in case of an emergency. The scenario for the first year was to locate and contact lost hikers (plastic human heads produced by the team on a 3D printer) without over-running or damaging the hikers. The UAV system uses a wall-mounted Kinect for its localization, and provides a live video feed of the field to be used mostly for the control of the UGV.

Below is a brief discussion of the developed S&R system components:

III. Unmanned aerial vehicle hardware

A. Aerial vehicle: AR Drone

The Parrot AR Drone 2.0 is an inexpensive, easy-to-control quadricopter, which combines an ARM A8 CPU running embedded Linux, an onboard IEEE 802.11 wireless network, and a sensor package including both pressure and ultrasonic altimeters, a 9-axis inertial measurement unit, and high definition video. This UAV is suitable for indoor use and has many built-in safety features such that it may be operated safely around people, including a soft foam hull that surrounds the propellers to prevent direct contact with objects such as people or walls, and a "kill switch" that immediately deactivates the motors if the propellers make contact with anything. The drone has two cameras, a low resolution 320x240 high frame rate 60fps camera looking down, and a high resolution 720p and moderate frame rate 30fps camera looking forward.

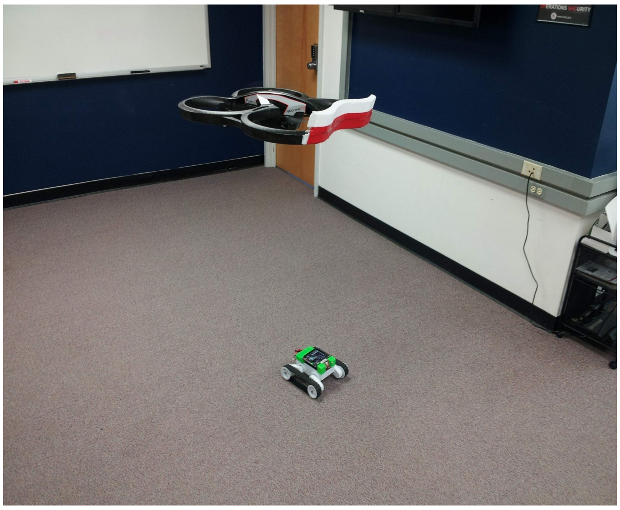

Figure *UAF Search&Rescue System with the UAV flying over the UGV*

B. Aerial vehicle control software

The UAV accepts commands through a wireless internet link via UDP command packets. Because the drone has no GPS, a Kinect sensor was used to track the quadricopter's location. This required a computer to read and interpret the Kinect sensor and assign commands to the drone through the wireless link.

Originally a software program named HeliSimple was used to interface the PC to the AR Drone [14], which was useful for getting started and for the initial manual control of the drone. As automation and control feedback was added, there were increasing difficulties due to the mismatch between our design stack and HeliSimple, so it became necessary to create a new control software library.

The new system, named Falconer [http://github.com/mrmoss/falconer], uses standard network sockets for communication. There are several ports used for control and communication with the AR Drone. Falconer uses three of these ports; one for control, one for navigation data, and one for video. The control socket is used for sending commands to the AR Drone, which is relatively straight-forward using the provided Software Developers Kit (SDK) [15]. The navigation data socket multicasts health and telemetry data. The protocol is provided in the SDK. The video stream from the AR Drone is a custom protocol based on the popular H.263 codec with which many HD videos are encoded. To convert the video a library called libav[16] was used, which is a standard library and are cross platform (they can be easily installed and used on Linux, Mac, and Windows).

C. Aerial vehicle localization: Kinect

The Microsoft Kinect Sensor, often used for mapping in previous studies, was used to localize and track the UAV. The Kinect is an inexpensive, parallax-based 3D image sensor, where an infrared laser emitter shines through a holographic diffraction grating to project a fixed pattern of dots into a room. An infrared camera then senses these dots from a vantage point shifted 10cm to the right, which allows onboard hardware to reconstruct depth from the observed parallax shift at each dot. The dots are interpolated to 640x480 resolution at a 30fps rate, and shipped across a USB 2.0 link to a control computer.

Because the sensor operates by reflected infrared light, objects that are not infrared reflective are difficult to observe except at close range. The AR Drone's exterior hull is constructed from extruded polystyrene (EPS) foam colored with carbon black, which reflects infrared light poorly, resulting in difficulties observing the drone in the Kinect depth image at ranges exceeding one meter. Applying red reflective tape, which is also reflective in infrared, to the exterior of the hull improved detection up to a few meters, but the low 3.5cm height of the indoor hull caused unreliable detection beyond that range. This was solved by cutting a foam ribbon 1.5cm wide by 5cm high by 40cm from infrared-reflective white EPS to match the curve of the hull, and glued this to the top of the hull with foaming urethane adhesive. This increases the weight of the hull by 5 grams, but does not appear to noticeably degrade the aerodynamics.

The larger overall target size provided by the foam ribbon allows the Kinect to reliably detect the UAV to a distance exceeding 6.5 meters, at the far edge of our field. Beyond this distance, the Kinect's depth quantization error becomes large enough to make flight control difficult.

D. Kinect depth image processing

The UAV's inertial measurement unit provides no sensor to determine absolute location, in relation to the rover or the field, which is why the Kinect was used to provide position feedback in this project. To interface with the Kinect, a software library called libfreenect [17] exposes a simple depth map interface, providing a 2D image giving the depth at each pixel.

Reliably segmenting this depth image to detect the UAV is a difficult task, because the background beyond the field may contain an arbitrary combination of non-reflective, reflective, and moving objects. Thus motion segmentation proved quite problematic at this point, despite the appeal that it would work in arbitrary, uncertain environments. As a solution, the team created a scheme to detect the UAV based on volume segmentation. The scheme starts by defining a 3D rectangular airspace over the play field; any depth pixels that project to 3D points inside this airspace are treated as part of the UAV. A rectangular airspace worked well for our purposes because no background objects protruded over the field, and no field objects extended more than a meter above the field, but in general one could define an arbitrarily complex airspace volume where UAV operations are allowed, and filter 3D point returns against this airspace. This volume segmentation scheme worked reliably even with moving backgrounds, such as spectators, as long as they did not enter the airspace or depth-shadow the UAV.

Because the UAV covers many depth pixels, and in general many projected 3D points will lie inside the airspace, we take the center of mass of all points inside the volume. This inherently filters the pixel quantization artifacts that are inevitable in a sensor of this type, and operates rapidly enough to provide 3D position fixes at the 30fps natural frame rate of the Kinect.

E. UAV flight control

Given a 3D position report from the Kinect, a proportional-derivative (PD) autopilot controller was developed to command the UAV's target pitch and roll values. Because the Kinect's minimum latency is 30ms, and the image processing and network add additional random communication delays, the AR Drone's onboard controller [18] was used to track deviations from those reference values at its natural rate of 200Hz. If the number of 3D points seen by the Kinect that project inside the airspace is too low, this indicates the UAV has left the airspace and the autopilot should be disabled. The final product of the Kinect, PD controller, and AR Drone interface was named Parrot Kinect [http://github.com/mrmoss/parrot_kinect]. The UAV's position control system is illustrated in Fig.2.

F. *Development of Web Interface*

The video stream from the UAV to the webpage only displays a small fraction of the actual field. The UAV must move for the teleoperator to be able to explore the entire field, navigate obstacles and locate hikers. To this aim, an interface was developed by the Fellows that provides both the UAV's video stream and a virtual version of the field displaying the UAV's current position and its desired position. The teleoperator uses this interface to command the UAV to different positions on the field based a 12 x 12 grid painted on top of the virtual field.

The user interface for control of the UAV is browser-based, and was created with a combination of HTML and JavaScript. The developed interface was based on a mixture GameMaker [19] and Processing [20]. Processing is a popular Java based programming language and GameMaker is a Pearl based game creation software toolkit. Both Processing and GameMaker are designed for programmers that are just starting out, which made it easy for the beginner students at all levels to use and understand.

The motivating idea behind a browser-based interface is that it allows for remote experimentation or remote access (as required by most S&R missions in some manner) from anywhere in the world, and without requiring a specific computer platform or software. Using JavaScript, mouse clicks are captured from a specific region of the browser window. These clicks are converted to coordinates which are then sent to the server for use in directing the UAV. An image of the field layout is drawn in this region, which is then used to tell the UAV what field position to travel to via mouse click. Visual feedback from the UAV is displayed through the use of two image objects to simulate double buffering. The images are displayed one on top of the other. The bottom image object updates itself with a new frame from the server and the positions are swapped continuously.

IV. UNMANNED GROUND VEHICLE HARDWARE

The project's hardware design choices were driven primarily by the mission requirements and educational activities planned for high school students. The S&R mission required a safe, reliable indoor operation, flexible networking and control capability. Considering limitations experienced with every new project, it soon became clear that high school students' primary hardware and software contributions would realistically need to be on the ground vehicle at least in the first year of the project, hence a sufficiently economical solution should be sought to allow us to equip each team with their own hardware. After some debate, the Atmel AVR-based Arduino [21] microcontroller was selected, since this platform has a portable and extremely friendly standard library and programming interface.

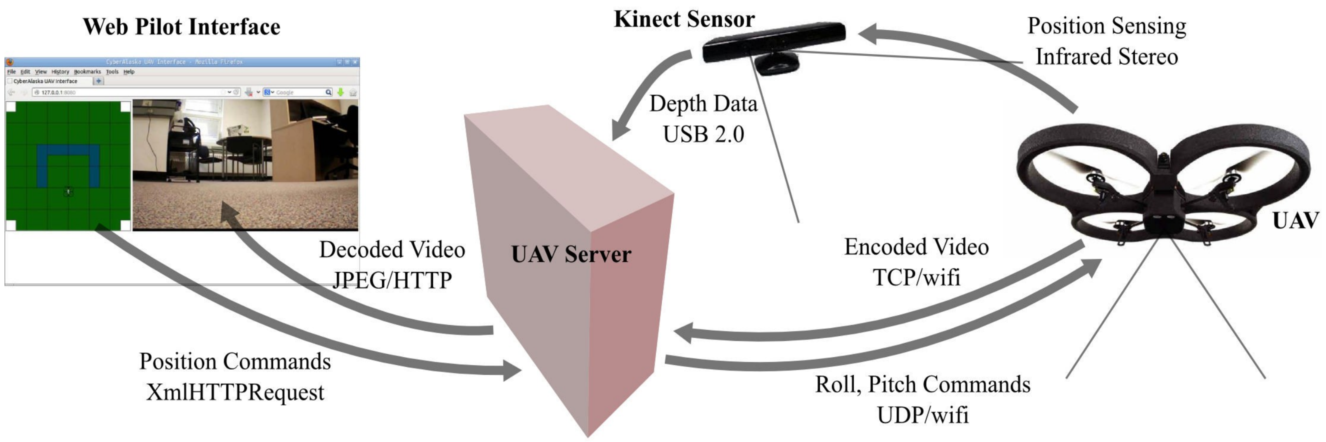

Figure *UAV position control system*

A. Networked tracked ground vehicle: RovoDuino

To have a reliable integrated platform that required little troubleshooting, a custom Arduino board was built, rather than an off-the-shelf Arduino with a shield stack. The board would contain an Atmega2560 (the same microcontroller found on the Arduino Mega board), two 2-channel H-bridge motor drivers for controlling 4 motors, a breakout for an XBee radio, and breakouts for all the remaining available pins for easy connection of sensors or other mechanisms. The control board was named RovoDuino, given in Fig.3, a contraction of Rover 5 and Arduino. The board design has been released as open source hardware.

The RovoDuino, given in Fig.4, was built on a Dagu Rover 5 platform [22], which comes pre-built with two tracks driven by four DC gearmotors and four quadrature encoders. Using an existing off-the-shelf platform minimized the time and effort of constructing the robot. The only extra requirement was some custom made brackets for holding the control board onto the chassis reliably. Board mounting brackets were fabricated on a local 3D printer, which was found to be a fast and inexpensive process for building plastic robot parts.

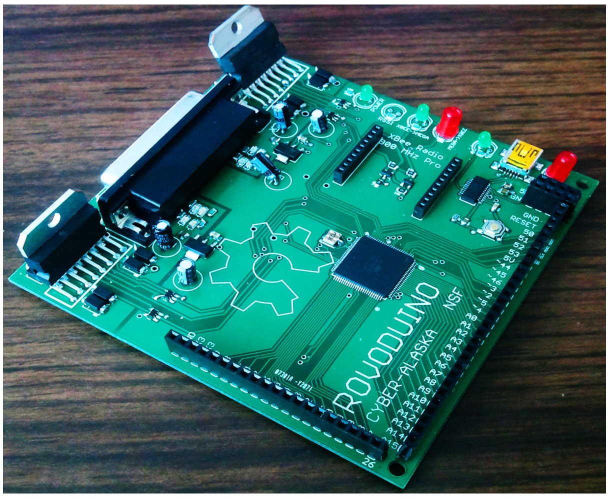

Figure *RovoDuino control board*

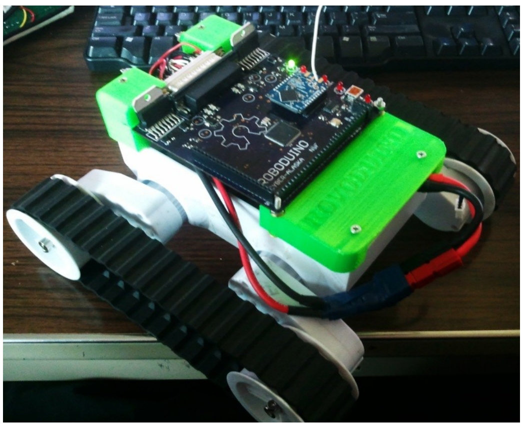

Figure RovoDuino

B. RovoDuino Control

Actuation of the motors of the differential-drive, RovoDuino is accomplished through a Pulse Width Modulated (PWM) capable pin on the Arduino. The H-bridge drivers amplify the PWM signal to 7.2 volt and 2 amp current levels useful to the motors. Because the tracks tie the motors into pairs, each pair of motors was assigned the same PWM signal. Also because of the pairing, feedback was taken from only one motor, with the assumption that there was no significant slip between the wheel and the track. PID controllers were designed for the velocity control of the wheel motors.

IV. NETWORKING SOFTWARE

A. Serial communication

The original communication setup for the challenge had another for the rover-side. The Arduino [23] serial interface on the rover-side is a worthy serial interface, but students had difficulty synchronizing higher-level messages over the byte-oriented asynchronous communication channel. The PC side interface was originally a serial library that was redesigned to work in the Arduino style, but it was not exactly the same.

To address these difficulties, it was decided that a new serial interface would be created, especially for ease of use in the classroom. This new interface was called SerialSync, and allows users to communicate over serial using a set of 16 variables that synchronize automatically with a single call. If a variable is changed on the PC, the same variable on the Arduino is updated. The new system is easier to use, and students were more productive than with raw serial communication.

One of our participating schools was from a city about 1500 miles away from UAF. All mentorship and education activities to this school were conveyed on regular basis, about 3-4 hours every week using Skype during classroom lectures and course videos for extra support. This school was also going to participate in the Annual Search & Rescue CPS Challenge, which necessitated a special interface to be created so the students could control the S&R system for the competition, with time-to-completion being the most important criteria. This led to the SocketSync interface. The SocketSync interface is identical to the SerialSync interface, except the implementation uses network sockets instead of serial ports. The advantage of using the same interface was the students only had to change one line in their code to communicate over the network. At the challenge site, a server was created called a SerialSocketRelay which allows a SerialSync and a SocketSync to talk to each other. The SerialSocketRelay interface was necessary because, even though the students were communicating over the internet, the Arduino-based challenge rover could only communicate through serial.

The challenge was held in a building on campus, with a restrictive university firewall blocking incoming connections from outside of campus. To allow students in another city to connect to the challenge rover, all traffic had to be forwarded through an SSH tunnel to a server in another building that had ports accessible to the outside world. Surprisingly the SSH tunnel only added 50ms of latency, so students were able to teleoperate their rovers reliably, with some remote teams placing better times than local teams.

B. PC compiler toolchain

Different computer security policies at each school and associated difficulties that were experienced called for a completely portable system that did not have to be installed and required nothing to be installed. Most of the computers in the classroom were Windows. The solution was a system comprised of compiler, a portable programming environment, and all the libraries needed[24]. The program uses the open source MinGW compiler [25], which is free to distribute. The portable programming environment used was AkelPad Portable [26]. AkelPad has syntax highlighting, code completion, and assignable hotkeys; which was everything needed. A single .zip container was prepared for the compiler, environment, and all the libraries used.

V. Results

The primary products of our first year activities were several web-accessible search and rescue systems, with all of its components built under the leadership of 2 graduate and 3 undergraduate CS and ECE Fellows. The participating undergraduate students contributed to all design aspects of the system either as Fellow assistants or as direct contributors, wherever possible. All Fellows were also quite successful in infusing CPS concepts into the curriculum of introductory engineering courses of two high schools (one in town, one 1500 miles away) using the design project and related course modules. Both classrooms were introduced to all aspects of the design process in preparation for the end-of-year CPS challenge. Under the Fellows' mentorship, all students were able to write the onboard firmware for their ground vehicle, interface it to their own PC control program, and field an operational ground vehicle by the day of the Challenge. The students also gained some experience in 2-D graphics, and GUI development. For a fair contest between teams participating locally and remotely, the challenge was run under a strict teleoperation protocol, with student teleoperators sequestered in a different room from the field and spectators, and "lost hikers" deployed to random locations and variable environmental obstacles created for each timed run. All 6 teams were able to control the UAV to coordinate search with their ground vehicle to find and gently touch each of the two "lost hikers" within the three minute time limit, on each of the two field designs. Rules were established regarding "hit and run" where the hikers are pushed or knocked over, and unfortunately one team experienced this incident due to the team's last-minute control program user interface change. Rules had also been established against the UGV crossing a "water" obstacle, but every team was able to drive around these obstacles reliably.

As a requirement of the program, Fellows, students, and teachers were given pre-and post-surveys by the CyberAlaska external evaluator. The Fellows and teachers were also interviewed by the evaluator. While the official report of the survey results are due in July 2013, the interviews and unofficial feedback from students and teachers indicate great satisfaction with our program.

VI. Conclusions

The Search& Rescue system proved to be a beneficial CPS project as it provided the Fellows with considerable in-depth and broad knowledge in their fields and across fields even by the end of the first year. The Fellows were also successful in providing the high school students with an understanding and some capability to develop a coordinated, multi-vehicle search and rescue mission via networked teleoperation, as their first experience in CPS. While all students were involved in system development at different levels, only a few high school students with prior background in programming and robotics were able to actively participate in all design aspects. The repertoire and experience gained in the first year is expected to improve classroom performance significantly in upcoming years. CyberAlaska's educational implementation details are subject to another paper in the near future.

The team also has plans in place to improve the control and increase the coordination level between the vehicles, also enhancing the functionalities of the web interface. We aim to provide solid experimentation platforms and education modules for the sustainability of CyberAlaska, and believe the existing web accessible hardware and software S&R platforms and associated course modules could already serve as useful assets for remote education in CPS and mechatronics.

Acknowledgment

This work was supported by NSF CPS/OCI/GK12 CyberAlaska grant (#OCI-1045601) and the Alaska Center for Unmanned Aircraft Systems Integration (ACUASI).

References

[1]. Grepl, R. ; Vejlupek, J., Lambersky, V. ; Jasansky, M. ; Vadlejch, F. ; Coupek, P. "Development of 4WS/4WD Experimental Vehicle: platform for research and education in mechatronics", IEEE International Conference on Mechatronics (ICM), 2011 pp: 893 – 898

[2]. F. Javier Maseda, Itziar Martija, and Irene Martija, "A Training Tool and Methodology to Allow Concurrent Multidisciplinary Experimental Projects in Engineering Education", IEEE TRANSACTIONS ON EDUCATION, VOL. 55, NO. 3, AUGUST 2012, pp. 357-364.

[3]. Michael J. Callaghan, Kerri McCusker, Julio Lopez Losada, Jim Harkin, and Shane Wilson, "Using Game-Based Learning in Virtual Worlds to Teach Electronic and Electrical Engineering", IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, VOL. 9, NO. 1, FEBRUARY 2013, pp. 575-584.

[4]. James McLurkin, Joshua Rykowski, Meagan John, Quillan Kaseman, and Andrew J. Lynch, "Using Multi-Robot Systems for Engineering Education: Teaching and Outreach With Large Numbers of an Advanced, Low-Cost Robot", IEEE TRANSACTIONS ON EDUCATION, VOL. 56, NO. 1, FEBRUARY 2013, pp.24-33.

[5]. J.J.Andina, L.Gomes, S.Bogosyan, "Current Trends in Industrial Electronics", IEEE Transactions on Industrial Electronics Education, Vol.57, No.10, pp: 3245-52, Oct.2010.

[6]. Wenshan Hu, Guo-Ping Liu, and Hong Zhou, "Web-Based 3-D Control Laboratory for Remote Real-Time Experimentation", IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 60, NO. 10, OCTOBER 2013, pp. 4673-4682

[7]. Jeff C. Jensen, Edward A. Lee, and Sanjit A. Seshia. Virtualizing Cyber-Physical Systems: Bringing CPS to Online Education. InProc. First Workshop on CPS Education (CPS-Ed), April 2013.

[8]. S. Bogosyan, B. Gadamsetty, M. Gokasan, A. Sabanovic, M. Unel, "Experimental Evaluation of Sliding Mode and EKF Observers for Network Delay Compensation in Bilateral Control", International Review of Electrical Engineering, Volume: 5, Issue: 5, Pages: 2484-2493, October 2010.

[9]. R. Voyles and H. Choset, "Editorial: Search and Rescue Robots," guest editors for the Special Issue on Search and Rescue Robots, Journal of Field Robotics, v. 25, n. 1/2, Jan./Feb. 2008.

[10]. Raymond K. Sheh, et al "Advancing the State of Urban Search and Rescue Robotics through the RoboCupRescue Robot League Competition" Field and Service Robotics Conference July 16, 2012,

[11]. Steven Kibler and Dejan Raskovic, "Coordinated Multi-Robot Exploration of a Building for Search and Rescue Situations "44th IEEE Southeastern Symposium on System Theory, University of North Florida, Jacksonville, FL, March 11-13, 2012, pp. 159-163.

[12]. Cai Luo, Andre Possani Espinosa, Alessandro De Gloria, and Roberto Sgherri, "Air-ground multi-agent robot team coordination", Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 6588-6591.

[13]. Gomes, L, Bogosyan S., "Current Trends in Remote Laboratories", IEEE Transactions on Industrial Electronics, 56 (12), pp. 4744-4756, 2009.

[14]. T. Krajník, et al. "AR-drone as a platform for robotic research and education." Research and Education in Robotics-EUROBOT 2011. Springer, 2011. 172-

[15]. P. Eline et al, "ARDRONE open API platform", Available: http://projects.ardrone.org

[16]. F.Bellardal, "libavcodec": http://libav.org

[17]. OpenKinect. libfreenect. Available: http://openkinect.org

[18].] P.J. Bristeau, et al. "The Navigation and Control technology inside the AR.Drone micro UAV", *18th IFAC World Congress*, Milano August 2011.

[19]. M.Overmars, GameMaker. http://www.yoyogames.com, 1997

[20]. B. Fry, 2001. Processing. Available: http://www.processing.org

[21]. ArduinoSoftware: http://www.arduino.cc

[22]. C. Dagu. "Rover5 " http://www.dagurobot.com/products.php?27

[23]. Arduino Software. Arduino. Available: http://www.arduino.cc

[24]. O. Lawlor and M. Moss. OrionDev-G++, Available: http://cyberalaska.cs.uaf.edu/blogs, 2013

[25]. C. Peters. MinGW (Minimalist GNU for Windows). Available: http://www.mingw.org

[26]. A. Shengalts, "AkelPad - an open source editor for plain text." Available: http://akelpad.sourceforge.net, 2006

[27]. Mechatronics Course", IEEE Transactions on Industrial Electronics,Volume: 57 , Issue: 10, 2009, pp. 3343 - 3354

[28]. Maga, D. ; Dudak, J. ; Pavlikova, S. ; Hajek, J. ; Simak, B, "Support of technical education at primary and secondary level ",15th International Symposium MECHATRONIKA, 2012, pp. 1-4.