Autonomous Robotics for Installation and Base Operations for Industrial Hygiene (ARIBO-IH)

Christopher Korpela, Kenneth Chaney, and Pareshkumar Brahmbhatt

***Abstract*— A framework is proposed for industrial hygiene inspection using a remotely-operated ground vehicle with multiple sensor payloads attached to it for detecting various hazardous gases and chemicals. A control scheme and a graphical user interface between the vehicle and operator is strictly mandated for tasks requiring remote inspection. Through leveraging existing navigation and path planning algorithms, the system can autonomously patrol hazardous areas and report dangerous levels back to the user. This paper presents recent results validating the industrial hygiene framework using the proposed system during sensor tests.**

Fig. 1. ARIBO-IH prototype vehicle inspecting a gas leak in a notional hazardous site. The inset in the lower left shows the camera view from the perspective of the robot.

I. Introduction

Throughout the Army, Industrial Hygiene (IH) teams are responsible for inspection, environmental reconnaissance, and emergency response. Industrial hygiene is an integral part of installation force protection and is an important component of an installations toxic industrial chemical spill planned response. Current best practices for conducting the IH mission requires direct human exposure to these hazardous environments. Robotic systems offer the potential to remove humans from these dangerous situations while maintaining the reliability and accuracy of the response team. Applying robotic solutions to this domain also contribute to the Department of Defense (DoD) unmanned systems goals outlined in the Unmanned Systems Integrated Roadmap FY2011-2036 [1]. Furthermore, robotic IH solutions are a force multiplier because these systems can be sent into a hazardous environment, parked, and allowed to collect data autonomously. Additionally, using robotic platforms for the IH environment is faster and safer than equipping and decontaminating a human. Finally, if successful, this project has potential DoD-wide applications.

A. Trends

Aging stocks of munitions and newly developed systems are creating larger quantities of dangerous materials that require monitoring and potentially, emergency response. In the current austere fiscal environment, enlarging a trained, professional IH team is a significant challenge. It is desirable to reuse/re-purpose existing inventory and to improve efficiency where possible. One way to accomplish this goal is to automate tasks using technology such as robots and networked systems. This automation can allow a smaller team of trained personnel to effectively manage a large group of tasks.

B. Problems

An Industrial Hygiene mission is to reduce soldier and employee exposure to environmental factors and stresses including: chemical (e.g., liquid, particulate dust, fumes, mist, vapor and gas), physical (e.g., electromagnetic radiation, temperature, ambient pressure, noise, vibration and ionizing radiation), and biological (e.g., agents of infectious diseases, insects, mites, molds, yeasts, fungi, bacteria and viruses) elements. The majority of hazards come from industrial processes on Army installations. Army industrial hygiene personnel are at risk from exposure to these hazardous environments in the conduct of their duties. Additionally, rapidly equipping human teams for response to incidents and post-action decontamination pose difficult challenges.

C. Benefits

ARIBO (Autonomous Robotics for Installation and Base Operations) is a TARDEC[1]-sponsored program to create “living labs” and leverage robotics technologies at military installations and civilian campuses. Through the use of robotic-enabled ground vehicles in a structured, controlled environment, the ARIBO-IH pilot will increase researchers, manufacturers, and users understanding and familiarity of these

Manuscript received December 1, 2014. This work was supported in part by the U.S. Army Tank Automotive Research, Development and Engineering Center (TARDEC). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government or TARDEC.

C. Korpela is with the United States Military Academy at West Point, NY 10996 USA christopher.korpela@usma.edu

K. Chaney is with Drexel University in the Department of Computer Engineering kpc49@drexel.edu

P. Brahmbhatt is with the University of Nevada, Las Vegas in the Department of Mechanical Engineering brahmbha@unlv.nevada.edu

[1]Tank Automotive Research, Development and Engineering Center

systems in real-world operational scenarios. The ARIBO-IH pilot safely provides the service of IH inspections, removing the human IH professional from a potentially hazardous situation while reducing cost. Additionally, the project will facilitate the design, standardization, deployment, and supervision of the resulting ARIBO-IH inspection robots. Pilot locations include the Stanford Linear Accelerator Center (SLAC) National Accelerator Laboratory, Fort Bragg, North Carolina, and the the United States Military Academy (USMA), at West Point. By using cadets as researchers, they are exposed to Army technologies and systems at the beginning of their careers. The benefits of generating officers with technological backgrounds in robotics systems is paramount to achieving the DoDs long term unmanned systems goals.

D. *Example Uses*

The robotic systems developed under this project could be employed in a number of situations to include:

- Environmental reconnaissance in routine industrial hygiene tasks and emergency response
- Weather station at the emission source
- Ventilation duct inspection
- Investigate suspected terrorist devices
- Site abatement or mitigation projects

This paper presents a solution to the industrial hygiene problem using a small ground vehicle equipped with a sensor package. The mobility platform for the system (Fig. 1) is detailed in Sec. II. To allow for unattended operation in a known environment, Sec. III describes the user interface to remotely monitor sensor readings and toggle between teleoperation and autonomous modes. Sec. IV describes the sensor package used to perform the IH mission with sensor testing and results found in Sec. VI. Navigation and path planning are described in Sec. V.

II. Mobility Platform

The iRobot PackBot is a fielded small robot used primarily for bomb disposal [2], [3]. The design is "semi-modular" where payloads can be installed to the base chassis, but they can only be installed in specific configurations, and when the software is properly configured. The integration of new payloads is often difficult and expensive. The vehicle uses military BB-2590 Li-Ion rechargeable batteries. While they have been widely deployed to both Iraq and Afghanistan, the Packbot is very expensive to purchase and maintain and overall reliability has been a challenge. There are two different generations: the obsolete 500 version and the current production 510 version. Many parts are interchangeable between the two models. The U.S. Army has discontinued the Packbot and no longer supports it as a program of record.

With a large inventory of unused and unsupported robots, the RS-JPO (Robotic Systems Joint Program Office) funded a 12 month effort to implement IOP on the existing Packbot platforms (Interoperability Profile V2, using JAUS which is the Joint Architecture for Unmanned Systems). Creating a kit design for retrofitting all fielded PackBots would reduce costs of maintaining the aged fleet. The standardized version

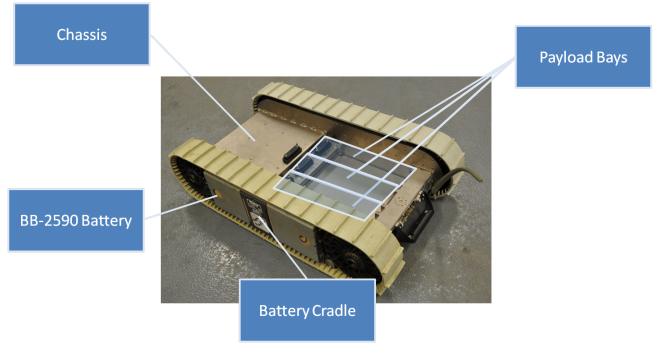

Fig. 2. Anatomy of the GVR-Bot.

of the PackBot makes a good research platform for a number of reasons:

- Open architecture
- Design is completely government-owned
- Designed for IOP V1 compliance
- Cost relatively low

The research platform became known as the GVR-Bot (Ground Vehicle Robotics is a branch of TARDEC) as shown in Fig. 2. The radio frequency was changed to 2.4 GHz so it could easily connect over standard existing wireless Internet protocols. All of the internal electronics of the robot were replaced with new motherboards, interface boards, and motor controllers. Bootloaders were added to the internal control boards (allowing flashing of all the software without disassembling the robot). See Fig. 3 for an example printed circuit board. The internal computer of the GVR-Bot manages linear and rotational accelerations/velocities of the vehicle, odometry, orientation, battery levels, GPS, and communications among others. The GVR-Bot used for IH missions contains an external embedded computer (Intel Core i7) to perform navigation and obstacle avoidance tasks, video monitoring, sensor integration through a microcontroller (Arduino Uno), and collects LIDAR data using a tilting Hokuyo UTM-30LX.

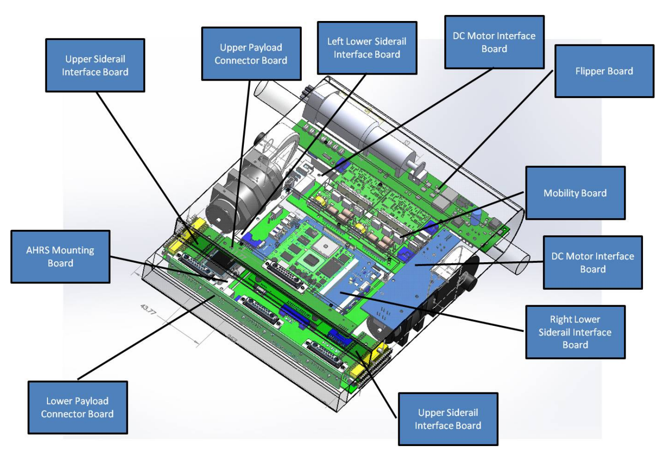

Fig. 3. Front assembly interface board. This printed circuit board and subassemblies are part of the complete electrical component redesign of the Packbot.

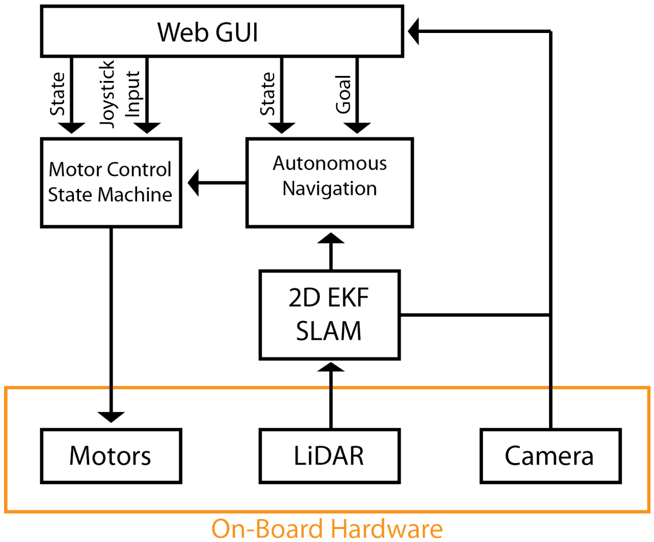

Fig. 4. System flow of individual robot interface.

III. User Interface Software Suite

A standardized graphical user interface (GUI)[2] was developed to be robust, intuitively usable by anyone, and serve as an all-in-one software suite (Fig. 4) for servicing inputs and outputs via Robot Operating System (ROS) which acts as the backend of the GUI. The GUI frontend seen by the user is responsible for relaying environmental sensor and navigation data. The GUI software backend is standardized for use on any ground robot that is able to traverse through a 3 degree of freedom (DOF) work space (x,y,θ) using an under-actuated controller. The user is able to control the robot's movement with a single virtual joystick controlling forward motion and turning. A wrapper is written to translate the joystick commands to actual motor commands. A second virtual joystick is available and dedicated to manipulating the camera viewpoint towards regions of interest in real time. This joystick allows the user to control a pan-tilt unit attached to the camera base. The camera feed and joystick bandwidth is automatically adjusted and delegated based on the bandwidth that is available at a particular instance between the robotic platform and the control unit. The GUI will alert the user if any bandwidth limits have been reached and will attempt to provide the user with the most up-to-date sensor and video data to allow the user to make the most informed decision possible in a given situation. Bandwidth concerns have a real impact upon mobile safety applications and must be taken into account; [4] goes into further detail about how mobile constraints can alter the user interface and backend.

A. Navigation Modes

As previously mentioned, the GUI allows the user to utilize two joysticks with one being for robot motion, and the other for camera pan-tilt motion; this two joystick scheme constitutes the first control mode. The GUI is designed to feature additional control modes. The control modes change the level of autonomy the robot exhibits during its mission. This feature, of course, requires for the different control modes to be hard-coded into the robot to ensure full functionality, both in terms of autonomous motion and safety of equipment in the facility being traversed. The change in control modes helps reduce the probability of mission failure or loss of robot if the control unit loses connection to the robot. The modes also change the level of involvement of the user in navigation duties, thus allowing the user to focus on higher level duties such as finding sources of contamination. This reduction of mental fatigue on the user helps increase missions success. All navigation modes utilize the GUI built-in features of map updating when differentials are found, obstacle proximity alerts, and environmental sensor warnings.

The second navigation mode allows for full autonomous exploration of the environment regardless of having any *apriori* knowledge of the environment or not having a map. The robot uses LIDAR-based SLAM-EKF [5], [6] to generate a map if none exists already. The map data is held in the on-board PC of the robot and can be transmitted on demand or streamed live. All navigation and obstacle avoidance actions are recorded on the full 3-D map, as metadata, that is generated as the robot progresses through the environment. Video data can also be toggled on if the user desires to visually inspect mission progress. By default, in order to reduce bandwidth saturation, map data is not transmitted to let mission critical sensor data take transmission priority. A fail-safe feature sends a warning message to the user to indicate that the robot has reverted back to Mode 1 for manual navigation. This only occurs if the robot is unable to continue through the environment or has encountered a navigational error in its on-board programming. If the user takes no action or the connection has been lost, the robot again elevates autonomy to its original setting to reverse course and return to the mission start position.

B. HTML Interface

The desire to create a web-based control application has been explored since the inception of the Internet [7]. The GUI is a web application that has all the mentioned features to be toggled on or off through a settings window which is separate from the main robot interaction window in order to reduce the clutter the user sees while performing a mission (Fig. 5). This will reduce the likelihood that the user will mistakenly click or press the wrong button during mission critical moments. The aforementioned navigation modes can be toggled on-the-fly by the user and are accessible directly from the main robot interaction window. The GUI frontend itself is based on Robot Web Tools [8], [9] which is a set of modules that help create web-based robot control applications. It allows ROS software package messages and topics to be accessible through a web interface that is constructed using HTML5 and Javascript via a wrapper called *rosjs* [10]. The GUI is essentially a web accessible overlay that allows the user to send and receive data across a connection between

[2] GUI is synonymous with Operator Control Unit (OCU).

Fig. 5. Screenshot of the developed web-based GUI allowing for view of the camera and 2D map.

Type	Range	Accuracy	Response Time	Typical Consumption
Temperature	-40°C to 125°C	± 2 °C between 0 and 70°C ± 4°C between -40 and 100°C	1.65 seconds	6 uA
Humidity	0 – 100 % Relative Humidity	± 4 % between 30 and 80 ± 6% between 0 and 100	< 15 seconds	0.38 mA
Carbon Monoxide	30 – 10000 PPM	0.13 – 0.31	1 second	3 mA
Methane	500 – 10000 PPM	0.6 ± 0.06	30 seconds	61 mA
LPG	500 – 10000 PPM	0.56 ± 0.06	30 seconds	61 mA

Fig. 6. Types of sensors showing accuracy, response time, and power consumption rates.

a ROS server, the control unit, and its corresponding ROS client, the robot. The advantage of using HTML5 is its cross-platform compatibility allowing the web application to be used by PCs, tablets, and smart phones [11]. On the ROS client, the robot computer runs a *roscore* node that uses a package called *rosbridge* which allows socket-based access to ROS through Javascript [12].

IV. Sensors

The sensor sub-system links directly to the user interface by providing plug-and-play functionality depending on mission requirements. Using modular sensors with standard interfaces and protocols, a wide variety of sensors can be fitted on the GVR-Bot. Power is provided through the vehicle's two BA-2590 batteries for up to 2 hours as the batteries each have 14 Ah, and with the full system operating, the current drain does not exceed 3A. The power system used for the prototype can support more powerful sensors if required. The initial prototype can detect gases, liquids, and atmospheric conditions as shown in Fig. 6.

For data acquisition, a microcontroller is used to process the sensor data and generate the data packets that are sent through the robot's embedded computer and wirelessly transmitted to the GUI to be stored in a database and displayed for the user. The software makes it possible for a user to modify, add, or subtract the current sensors used in order to fit the desired application. Furthermore, the multiple I/O channels on the microcontroller allow for easy access to integrate additional sensors. The structure of the sensor packet is scalable to add additional information as shown in Table I. The received data is saved in a database to provide a history of the event readings. As explained in the user interface section, the familiarity of the GUI ensures that the user does not have to learn a completely new control interface. A sensor toolbar was developed to mesh with the robot's current user interface to provide live data readings. When a sensor reading passes a dangerous threshold, the toolbar will change colors to alert the operator. Under each sensor, a button displays a graph of the sensor's history.

Sensor Packet		
Sensor ID	Binary	1 byte
Data Type	Binary	1 byte
Data Length	Binary	1 byte
Data	(data type)	(data length) bytes

TABLE I

Sensor packet structure used to communicate sensor data back to the user interface.

V. Navigation

The internal computer of the GVR-Bot passes sensor data to the on-board, external embedded computer for processing, map generation, and development of a navigation scheme. The external embedded computer is directly connected and receives updates from the LIDAR in real-time. The tilting capability of the LIDAR allows 3-D point cloud data to be collected for use in developing a 3-D navigation solution. For ARIBO-IH missions, accurate inertial measurements are not required because navigation occurs in highly structured indoor environments, and odometry data helps mitigate errors in position. The computer then uses the laser data to generate a 2-D planar occupancy grid map. The technique of using occupancy grid maps based on LIDAR scans is being implemented because it can be done at a low cost computationally, and it is a proven approach for localizing a ground robot in a real-world setting [13], [14]. The 2-D SLAM information is coupled with a 3-D pose estimate that is generated using an Extended Kalman Filter (EKF). This coupling allows the 2-D planar map to be overlayed with robot pose data estimated via EKF.

This information is locally stored in memory for purposes of future unplanned events that may occur during the mission, such as connection loss. The highly detailed 3-D map and a 2-D planar map are generated, while a coarser version of the 2-D map is created for the purposes of wireless transmission to the control unit to be displayed on the GUI for the user; while netowrk communications aren't normally a problem in modern buildings, it is hard to ensure wireless communications everywhere, approaches from [15] have been taken into account for multi-vehicle cooperation. This multi-resolution map representation does not rely on any type of filtering or downsampling, but it simultaneously

Fig. 7. Interior corridor of SLAC tunnel.

generates multiple maps of different resolutions that have scalability and low cost computationally. This approach is further described in [14], [16]. The high computation capability of the computer allows for both the SLAM and 3-D EKF components to be run at soft real-time with a low enough latency in calculation that it is negligible compared to actual or hard real-time 3-D navigation schemes.

A. *Practical Implementation*

The Stanford Linear Accelerator Center (SLAC) National Accelerator Laboratory has partnered with TARDEC and West Point to field a GVR-Bot for use in the two-mile long tunnel. Currently, maintenance crews must traverse the tunnel and visually inspect for gases, liquids, radiation, vibrations, ultrasonic frequencies, and atmospheric effects that could be harmful to humans. As previously described, the GVR-Bot navigates the tunnel as shown in Fig. 7 which is generally clear of debris and has known features throughout the corridor. Sensor packages are installed as described in Sec. IV and the graphical interface provides robot pose and environmental information as discussed in Sec. III.

VI. Sensor Testing and Results

Since gas sensors are inherently inaccurate due to their manufacturing processes and biases, each sensor had to be calibrated by graphing the output of each sensor compared to the known gas in the testing environment. The testing environments consists of 1.0 liter sealed containers with the sensor inserted into one end and a septum for needles inserted in the other as illustrated in Fig. 8. The 1.0 liter volume simplified the calculations in parts-per-million (ppm) and each gas was introduced to the environment via a syringe and needle.

By comparing the sensor output (x-axis) to the known concentration of gas in the testing environment (y-axis), a curve is formed, which can be best fit to an exponential curve by focusing the comparison on moderately to severely dangerous gas concentrations, which is approximately 2,000 to 10,000 ppm. The best fit equation for the most accurate

Fig. 8. Testing environments (from left to right): MQ-4, MQ-6, MQ-7, and MQ-8 sensors.

TABLE II
R^2 values associated with fit curves in Figure ??

Test	R^2
Methane Test 1	0.994
Methane Toxic	0.996
CO Test 1	0.998
Hydrogen Test	0.969
LPG Test 1	0.999

test is used in software to correct the sensor output to the most accurate output for transmission to the operator through the OCU. Figs. 9-13 displays the graph for the MQ-4 (Methane) sensor along with the graphs for MQ-6 (Liquefied Petroleum Gas), MQ-7 (Carbon Monoxide), and MQ-8 (Hydrogen) sensors.

Using the correction equations in software with the given sensor outputs, the final results closely match the expected concentrations. The corrected outputs will be able to discern the difference between a safe and a dangerous environment.

VII. Conclusions

Industrial Hygiene monitoring and response functions possess the potential for improving efficiency and safety through the use of robots. Robotic systems such as the GVR-bot can remove humans from dangerous environments while accurately and reliably conducting tasks that are critical to

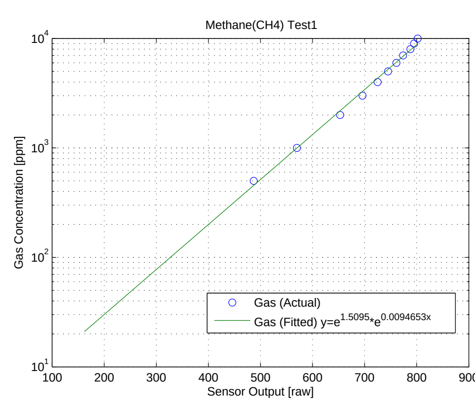

Fig. 9. Methane gas test.

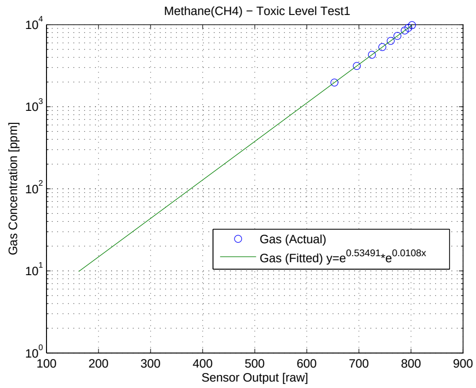

Fig. 10. Methane toxic levels.

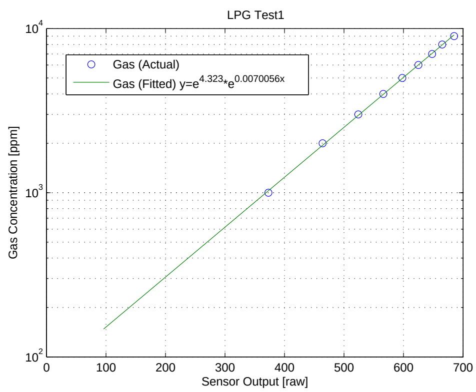

Fig. 11. Liquefied Petroleum Gas test.

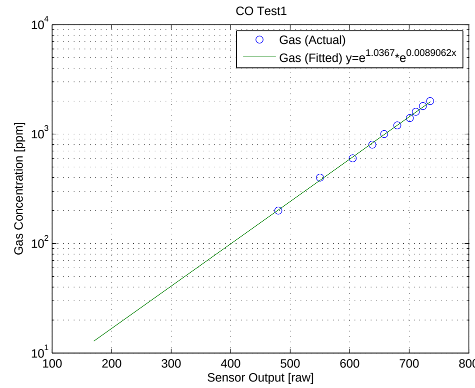

Fig. 12. Carbon Monoxide test.

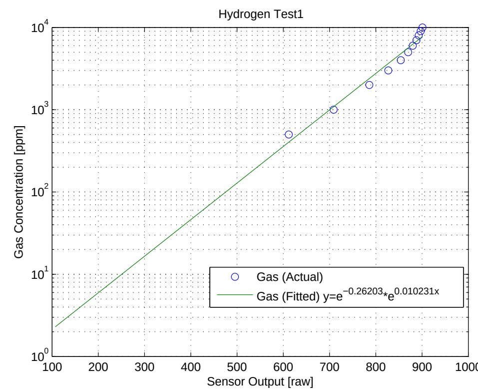

Fig. 13. Hydrogen test.

operations on all Army installations. These improvements in efficiency and safety can be obtained in a fiscally responsible manner through the use of existing Army systems. Additionally, research conducted at the Academy benefits the Army long-term by exposing future officers to Army programs and technology at the beginning of their career.

References

[1] D. of Defense, "The unmanned systems integrated roadmap fy2011-2036," 2011.

[2] "iRobot packbot," http://www.irobot.com, accessed: 2014-09-4.

[3] R. Hogg, A. Rankin, S. Roumeliotis, M. McHenry, D. Helmick, C. Bergh, and L. Matthies, "Algorithms and sensors for small robot path following," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 4, 2002, pp. 3850–3857 vol.4.

[4] P. B. M. S. J. H. C. P. Peter Erickson, Andrew Weinert and S. Miller, "Designing public safety mobile applications for disconnected, interrupted, and low bandwidth communication environments," in *IEEE International Conference on Technologies for Homeland Security*. IEEE, 2013.

[5] J. Weingarten and R. Siegwart, "Ekf-based 3d slam for structured environment reconstruction," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 2005, pp. 3834–3839.

[6] J. A. Castellanos, R. Martinez-Cantin, J. D. Tardós, and J. Neira, "Robocentric map joining: Improving the consistency of ekf-slam," *Robotics and Autonomous Systems*, vol. 55, no. 1, pp. 21–29, 2007.

[7] K. Goldberg and R. Siegwart, *Beyond Webcams: an introduction to online robots*. MIT Press, 2002.

[8] http://robotwebtools.org/.

[9] J. Lee, "Web applications for robots using rosbridge," *Brown University*, 2012.

[10] S. Osentoski, G. Jay, C. Crick, B. Pitzer, C. DuHadway, and O. C. Jenkins, "Robots as web services: Reproducible experimentation and application development using rosjs," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 6078–6083.

[11] W. Hilton, D. M. Lofaro, and Y. Kim, "A lightweight, cross-platform, multiuser robot visualization using the cloud," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 1570–1575.

[12] C. Crick, G. Jay, S. Osentoski, B. Pitzer, and O. C. Jenkins, "Rosbridge: Ros for non-ros users," in *Proceedings of the 15th international symposium on robotics research (ISRR)*, 2011.

[13] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[14] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," in *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*. IEEE, 2011, pp. 155–160.

[15] P.Abichandani, H. Benson, and M. Kam, "Multi-vehicle path coordination in support of communication," in *Proc. International Conference on Robotics and Automation (ICRA'09)*, Kobe, Japan, May 2009.

[16] M. Habbecke and L. Kobbelt, "Iterative multi-view plane fitting," in *Int. Fall Workshop of Vision, Modeling, and Visualization*, 2006, pp. 73–80.