DAHLGREN DIVISION
NAVAL SURFACE WARFARE CENTER
Panama City, Florida 32407-7001

NSWC PC/MP-04/07

EXPENDABLE ELECTRIC LANDROVER (EEL)

MITCH GAVRILASH

LITTORAL WARFARE TECHNOLOGY AND SYSTEMS DEPARTMENT

NAVAL SURFACE WARFARE CENTER - PANAMA CITY

JULY 2004

REPORT DOCUMENTATION PAGE

Form Approved
OBM No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	JULY 2004	

4. TITLE AND SUBTITLE
EXPENDABLE ELECTRIC LANDROVER (EEL)

5. FUNDING NUMBERS

6. AUTHOR(S)
MITCH GAVRILASH

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Attn:
Commanding Officer
NSWC-PC
110 Vernon Ave
Panama City, FL 32407-7001

8. PERFORMING ORGANIZATION REPORT NUMBER
NSWC PC/MP-04/07

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

The Defense Threat Reduction Agency
8725 John J. Kingman Rd.
Ft Belvoir VA 22060-6201

Office of Naval Research
800 North Quincy Street
Arlington VA 22217

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED

12b. DISTRIBUTION CODE
A

13. ABSTRACT (*Maximum 200 words*)
Three generations of Expendable Electric Landrover (EEL) robots were developed during fiscal year 2003 at the Naval Surface Warfare Center, Panama City (NSWC-PC). The vehicles were designed to meet stringent shock requirements in order to survive air deployment via parachute drop. On 20 March 2003, as part of the "STORK" Project demonstration, an EEL became the first unmanned ground vehicle (UGV) successfully emplaced by a fixed wing unmanned aerial vehicle (UAV). The STORK Initiative was part of the USSOCOM Pathfinder Advanced Concept Technology Demonstration (ACTD). The Defense Threat Reduction Agency (DTRA) provided funding for the EEL development.

14. SUBJECT TERMS
EXPENDABLE ELECTRIC LANDROVER, ROBOTS, UNMANNED GROUND VEHICLES,

15. NUMBER OF PAGES
104

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

NSN 7540-01-280-5500

Standard Form 298 (Rev 2-89)
Prescribed by ANSI std. Z39-18
298-102

FOREWORD

Three generations of Expendable Electric Landrover (EEL) robots were developed during fiscal year 2003 at the Naval Surface Warfare Center, Panama City (NSWC-PC). The vehicles were designed to meet stringent shock requirements in order to survive air deployment via parachute drop. On 20 March 2003, as part of the "STORK" Project demonstration, an EEL became the first unmanned ground vehicle (UGV) successfully emplaced by a fixed wing unmanned aerial vehicle (UAV). The STORK Initiative was part of the USSOCOM Pathfinder Advanced Concept Technology Demonstration (ACTD). The Defense Threat Reduction Agency (DTRA) provided funding for the EEL development.

Although the vehicle prototyping cost was kept below $20 K per robot by utilizing commercial-off-the-shelf (COTS) parts, it was rugged enough to remain fully functional after an accidental 25-foot free fall. Software development costs were minimized by porting (with only minor modifications) the amphibious crawlers control system previously implemented for the Office of Naval Research (ONR) funded Surf Zone Reconnaissance Project. The capability of remotely programming and re-tasking the robots is a unique feature of the EELs and Surf Zone Crawlers, which is not found in commercially available robots.

The robots provide live video feedback and can be joystick controlled, but when outfitted with the appropriate sensors they are also capable of fully autonomous operation. The asynchronous communication protocol is suited for time-multiplexed, narrow band channels and has been successfully exercised using acoustic (underwater), magneto-inductive (air to underwater) and secure SATCOM links. Remote command and control of groups of (semi) autonomous robots is possible across large distances.

This report has been reviewed and approved by the- Littoral Warfare Technology and Systems Department.

Approved by:

DELBERT C. SUMMEY, Head
Littoral Warfare Technology and
Systems Department

CONTENTS

ILLUSTRATIONS

I. EXPENDABLE ELECTRIC LANDROVER (EEL)

A. OVERVIEW

The Naval Surface Warfare Center Panama City (NSWC-PC) Surf Zone Robotics Team, under the sponsorship of the Defense Threat Reduction Agency (DTRA), conceptualized, designed and implemented an inexpensive, impact resistant, four-wheel-drive land robot. The vehicle leverages the control system previously developed for the amphibious crawlers as part of the Office of Naval Research (ONR) 321OE's Very Shallow Water (VSW)/ Surf Zone (SZ) Program.

The concept emerged in June 2002 as representatives from Air Force Research Laboratory (AFRL) Tyndall Robotics and NSWC-PC met to select potential unmanned systems to participate in the DTRA Comprehensive Hazmat Emergency Response Capability Assessment Program (CHERCAP) exercise scheduled for early February 2003. At that time, Eglin Air Force Base (AFB) Unmanned Aerial Vehicle (UAV) Battlelab and AFRL Tyndall Robotics were already working on a separate effort known as the "STORK" Project, which was part of the Pathfinder Advanced Concept Technology Demonstration (ACTD). The STORK demonstration required a UAV to deliver an unmanned ground vehicle (UGV) via parachute drop. After the UAV successfully deployed the UGV, the UGV was to be controlled from over the horizon (OTH) through a wide bandwidth communication relay onboard the loitering UAV.

B. DESIGN REQUIREMENTS

The STORK UGV design constraints were vehicle survivability after a 15 feet per second collision with a concrete or asphalt surface, a maximum size of 1 foot by 1.5 feet by 2.5 feet and a weight limit of 65 pounds. Initial laboratory shock tests at NSWC-PC showed that pneumatic tires can successfully mitigate the initial impact g forces seen by a robot's internal electronics. Additional requirements included bi-directional radio link for command and control and unidirectional video link to an operator control unit (OCU).

C. DESIGN METHODOLOGY

The EEL was designed as a low cost UGV that could be air-deployed and provide OTH ground surveillance via live video. EEL development costs were minimized by leveraging control systems previously designed and implemented by the Navy for amphibious surf zone crawlers.

In pursuit of modest fabrication costs, EEL was constructed using commercial-off-the-shelf (COTS) components*. Ten-inch pneumatic tires and inexpensive, yet very quiet, 12 VDC automotive gear motors were selected for locomotion. NSWC-PC engineers completed initial computer aided design (CAD) drawings of the fully symmetric, four-wheel drive, slip steered Expendable Electric Landrover (EEL) robot by September 2002. Figure 1 depicts an isometric CAD rendering of the vehicle with top cover plate removed.

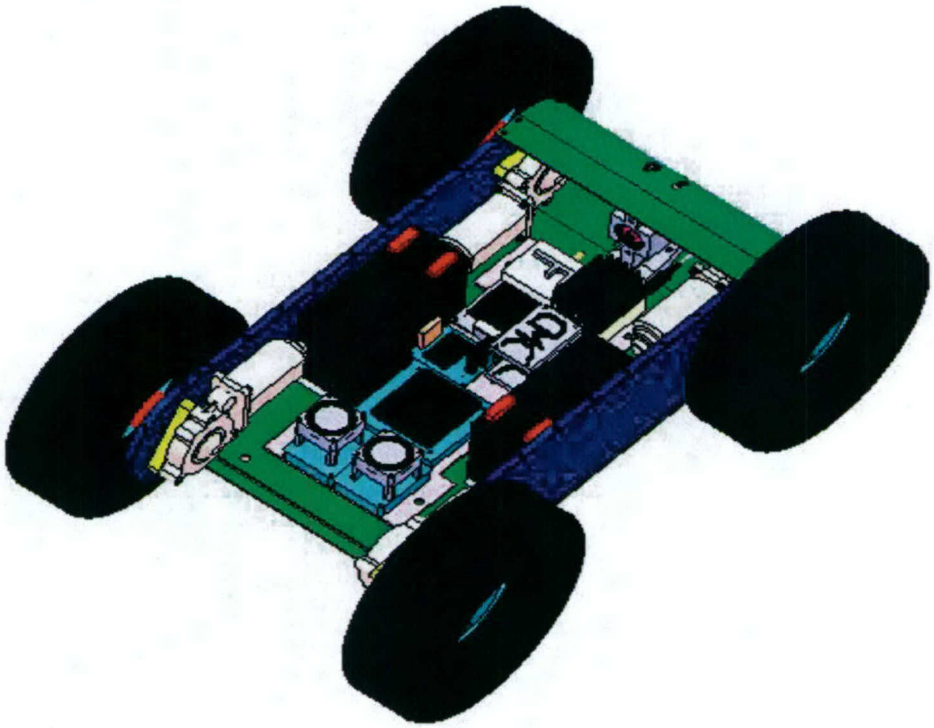

FIGURE 1. CAD ISOMETRIC VIEW OF THE NSWC-PC EXPENDABLE ELECTRIC LANDROVER (EEL)

D. MECHANICAL DESIGN AND SPECIFICATIONS

Mechanical design highlights include pneumatic tires on 5-inch steel hubs, passive axles inside externally mounted bearings to protect the power train from axial and radial shock and a four point resilient mounted electronics plate. Vehicle dimensions as shown in Figures 2 and 3 are: 10 inches tall (tire diameter), 18 inches wide (measured laterally from outside the steel hubs) and 28 inches long (18 inches longitudinally between motor axles).

---

* The appearance of trade names in this document does not constitute endorsement by the Department of Defense, the Navy, or the Dahlgren Division Naval Surface Warfare Center – Panama City (NSWC - PC)

FIGURE 2. FIRST EEL PROTOTYPE OPERATIONAL OCTOBER 2002 (OPENING IN THE FRONT IS FOR VIDEO CAMERA)

FIGURE 3. SECOND EEL ROBOT COMPLETED NOVEMBER 2002 (C3 AND VIDEO DIPOLE ANTENNAS VISIBLE IN THE BACK)

The vehicle's nominal weight is 54 pounds, which includes the chassis, batteries, motors, wheels, on-board computer, and navigation sensors. Early tests confirmed a payload capability of 15 additional pounds while maintaining adequate mobility on hard surfaces, sand and gravel. Initial experimental results show that capability for larger payloads (up to 30 pounds) could be achieved by replacing the narrow pneumatic tires with wider, slick surface wheels, such as depicted in Figure 4.

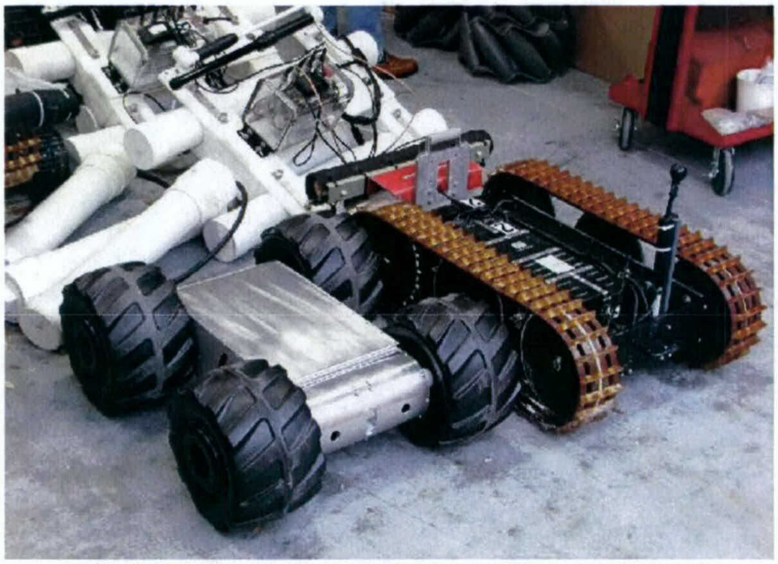

FIGURE 4. EEL WITH PLASTIC WHEELS NEXT TO SURF ZONE AMPHIBIOUS CRAWLER

E. ELECTRONICS DESIGN

Electrical power is provided by a pair of 12 V, 7.2 Ampere-hour (Ah) lead acid gel cell batteries connected in parallel. Nominal vehicle speed is 1 meter per second with an estimated run time of two hours. An AMD 188ES-based 16-bit microcontroller, a flux gate compass and two Wide Area Augmentation System (WAAS) enabled Global Positioning System (GPS) receivers (one topside, second bottom-side of vehicle) support vehicle geo-location and operation either in joystick or supervised autonomy (waypoint navigation) modes. Industrial, scientific, and medical (ISM) band spread spectrum modems provide the communications channel for command and control. For remote teleoperation of the robot, a wide-angle analog color camera and an L-band video link provide visual feedback.

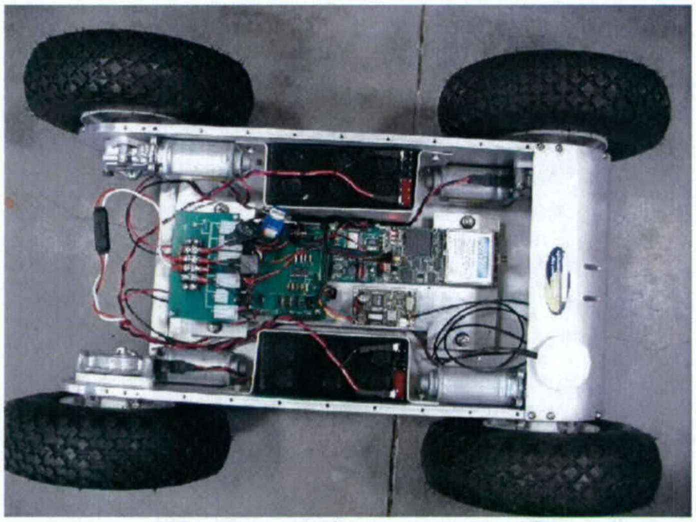

FIGURE 5. EEL ELECTRONICS

F. CONTROL SYSTEM

1. Low-Level Vehicle Control

To compensate for terrain biases during waypoint navigation, a compass-based proportional-integral-differential (PID) closed control loop, operating at a frequency of 10 Hertz, effectively tracks the robot's intended heading in natural environments. Controls are defined by the equations below:

$$Err(i) = Intended\_Heading - Compass \tag{1}$$

$$Err\_Integral = Err\_Integral + Err(i) \tag{2}$$

$$Delta\_Err = (\, Err(i) + 3 * Err(i\text{-}1) - 3 * Err(i\text{-}2) - Err(i\text{-}3)\,)\, /\, 6 \tag{3}$$

$$PID\_Output = Kpro * Err(i) + time\_intv * Kint * Err\_Integral + (Kdif/time\_intv) * Delta\_Err \tag{4}$$

$$Speed\_Left = Nominal\_Speed + PID\_Output \tag{5}$$

$$Speed\_Right = Nominal\_Speed - PID\_Output \tag{6}$$

Where *Kpro, Kint, Kdiff* are the loop gain coefficients for the proportional, integral and differential terms respectively, *Err(i)* denotes present error between the intended and actual heading of the robot, and *Err(i-N)* denotes past errors between the intended and actual heading of the robot.

The classic first order discrete time equivalent of the derivative term has been implemented by a higher order difference, Equation (3) to overcome susceptibility to noise.

2. ROBO Command System

The EEL uses the NSWC-PC-developed "ROBO" command system, which was originally developed to support the amphibious mine countermeasures (MCM) surf zone vehicles. ROBO is an ASCII-based command set and communications protocol, which is described in Table 1. ROBO is a fully functional application programming interface (API) intended for groups of remotely programmable autonomous robots.

TABLE 1. ROBO ASCII-BASED COMMAND SET

ROBO Elementary Commands	ROBO ROV* and Diagnostic Commands	
	COMMAND	REPLY
A(djust relative timer) # seconds	**a**(ccelerate)	**a** # (nominal speed)
B(ack-up) # seconds	**b**(ack-up)	**B**
C(orrection) Diff. GPS RTCM** Corr. String	**c**(oordinates)	**c** GPS String
D(estination) sets global X#,Y#, Z#	**d**(ecelerate)	**d** # (nominal speed)
E(xecute Convention) RESERVED	**e**(nergy)	**e** # (percent)
F(orward) # meters	**f**(orward)	**F**
G(roup) # (assign)	**g**(roup)	**g** # (currently assigned to)
H(eading) # set heading in compass degrees	**h**(eading)	**h** # (intended)
I(mage) # (bytes)	**i**(d)	**id** #
J(og) # selects diagnostic maneuver type	**j**(og)	**j**
K # sets heartbeat interval in seconds	**k**(eep heartbeat going)	**K**
L(eft) # (relative degrees)	**l**(eft)	**L**
M(ove) to X #, Z # (for obstacle avoidance)	**m**(achine) RESERVED	
N(avigation bounds) e.g., Z<aX+b	**n**(avigation boundary)	**n** # nav. bound or obstacle
O(rigin) lat #, lon # (sets origin coordinates)	**o**(rientation)	**yaw, roll, pitch**,
P(osition) X#, Y#, Z# (set relative position)	**p**(osition) rel. to origin	**x, y, z**
Q(uery) # report type	**q**(uiet) mode	**q** (toggles verbose flag)
R(ight) # (relative degrees)	**r**(ight)	**r**
S(top) all stop	**s**(top motors)	**s**
T(ransit) range # (meters), bearing # (deg.)	**t**(ime)	**t** # (seconds)
U(nknown object) sensor # detection code	**u**(nknown object type)	**u** # (object type)
V(elocity) left side, right side, nominal speed	**v**(elocity)	**V_left, V_right, V_nominal**
W(aypoint) X #, Z #	**w**(ell being)	**w** # # (status and health)
# **X** set x, positive in North latitude direction	**x**	**x** # (relative meters North)
# **Y** set y, depth/altitude, zero at sea level	**y**	**y** # (depth in meters)
# **Z** set z, positive in the East longitude dir.	**z**	**z** # (relative meters East)
ROV = remotely operated vehicle* * Radio Technical Commission for Maritime Services*		

3. Multi-Vehicle Control

Multiple vehicles can be controlled individually or assigned to and addressed in groups. Figure 6 depicts the ROBO 'data link/media access layer' which corresponds to Joint Architecture for Unmanned Systems (JAUS) message Subsystem ID header.

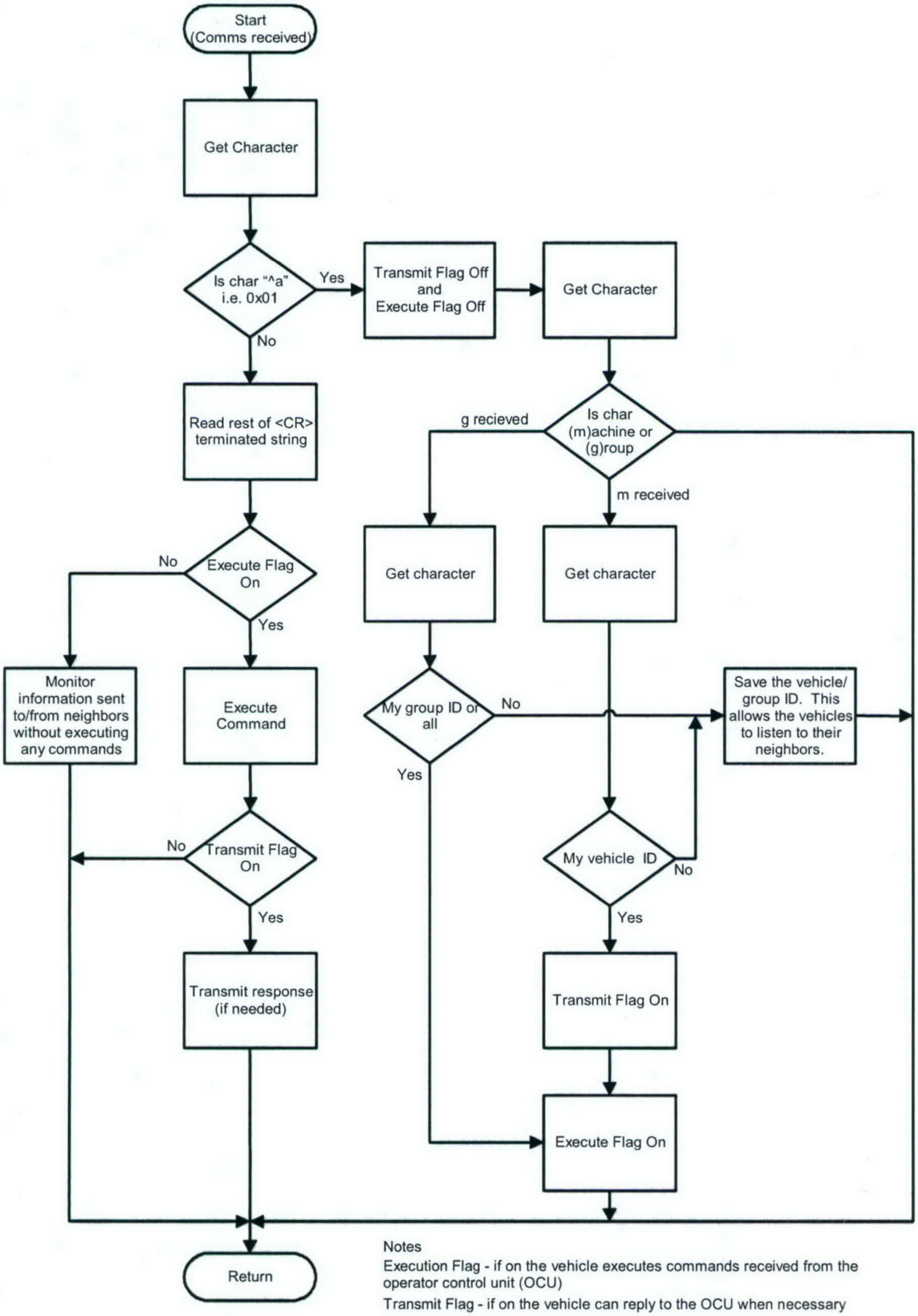

FIGURE 6. THE ROBO DATA LINK/MEDIA ACCESS LAYER

4. Remotely Programmable Robots

The NSWC-PC Surf Zone Robotics Team has put forth and implemented the concept of operationally programmable autonomous littoral systems (OPALS) as a means of remotely programming and re-tasking swarms of small autonomous vehicles over low bandwidth communication channels. The team has successfully demonstrated the concept using acoustic, magneto-inductive and satellite communication channels. Command and control messages as well as highly compressed images have been transferred at data rates ranging from 20 to 2400 bits per second (bps).

In addition to the ROBO commands listed in Table 1, a scripting mechanism based on FORTH, an ANSI standard programming language, provides conditional branching, loops, arithmetic and logic operations for robot behavior development. The mini-FORTH command interpreter supports mission scripting in terms of elementary control commands. FORTH was selected for its execution speed since new words are only interpreted once, then compiled and placed in an internal dictionary.

The ROBO commands and programming constructs, provided in Appendices A and D respectively, define the robot's vocabulary and API. An example defining a search path by entering at the command prompt a sequence of four relative destinations expressed in range and bearing format is listed below:

": path 20 90 T 4 180 T 20 270 T 4 180 T ;".

A lawnmower pattern could be expressed by repeating *path* N times:

": search N 0 DO path LOOP ;".

With the addition of obstacle detection electronics, a random walk behavior (obstacle avoidance is inherently built in) could be implemented using the ROBO API.

Example of a safe wandering macro (random walk behavior) follows:

```
: rndm_dist    maxdist GET RND mindist GET + F ;   ( move forward a random distance )
: rndm_turn    maxturn GET RND H ;                 ( execute a random angle turn )
: back         backdist GET B rndm_turn ;          ( back-up followed by random turn )
: back_flag    collision GET bounds GET OR ;       ( determine if need to back-up )
( execute random walk in a bounded area until relative timer exceeds 600 seconds )
: walk   BEGIN rndm_dist back_flag 1 = IF back ELSE rndm_turn ENDIF t GET 600 > UNTIL ;
```

NSWC-PC has implemented ROBO templates for various behaviors including obstacle avoidance algorithms and numerous search patterns supporting autonomous MCM operations in the SZ and VSW. Recent work has been focused on the development of collaborative and cooperative behaviors for multi-vehicle systems.

G. NAVIGATION

The following equations provide vehicle position estimates:

$$\mathit{North\_DR\_Meters} = \mathit{North\_DR\_Meters} + \mathit{Nominal\_Speed} * \mathit{time\_intv} * \cos(\mathit{Compass}) \tag{7}$$

$$\mathit{East\_DR\_Meters} = \mathit{East\_DR\_Meters} + \mathit{Nominal\_Speed} * \mathit{time\_intv} * \sin(\mathit{Compass}) \tag{8}$$

$$\mathit{North\_GPS\_Meters} = (\mathit{GPS\_Latitude} - \mathit{Origin\_Latitude}) * 1852 \quad \text{// 1852 meters per nautical mile} \tag{9}$$

$$\mathit{East\_GPS\_Meters} = (\mathit{GPS\_Longitude} - \mathit{Origin\_Longitude}) * \cos(\mathit{Latitude}) * 1852 \tag{10}$$

$$\mathit{North\_Position} = \mathit{North\_DR\_Meters} - (\mathit{North\_DR\_Meters} - \mathit{North\_GPS\_Meters}) * \mathit{Sigma\_DR}\text{^}2 / (\mathit{Sigma\_DR}\text{^}2 + \mathit{Sigma\_GPS}\text{^}2) \tag{11}$$

$$\mathit{East\_Position} = \mathit{East\_DR\_Meters} - (\mathit{East\_DR\_Meters} - \mathit{East\_GPS\_Meters}) * \mathit{Sigma\_DR}\text{^}2 / (\mathit{Sigma\_DR}\text{^}2 + \mathit{Sigma\_GPS}\text{^}2) \tag{12}$$

Where the vehicle position estimate (*North, East, Down* coordinate system) is comprised of deduced reckoning (*DR*) based on timer compass projections in Equations (7) and (8), fused with *GPS* input ('flat Earth approximation') in Equations (9) and (10). Furthermore, latitude may be defined by the following equations:

$$\mathit{Latitude\_in\_Minutes} = \mathit{Origin\_Latitude\_in\_Minutes} + \mathit{North\_Position} / 1852 \tag{13}$$

$$\mathit{Longitude\_in\_Minutes} = \mathit{Origin\_Longitude\_in\_Minutes} + \mathit{East\_Position} / (1852 * \cos(\mathit{Latitude})) \tag{14}$$

The robot can report its position (JAUS 'local/global pose' functional equivalents) in the form of North/East displacement from an arbitrarily selected origin or Latitude/Longitude format, conversion conveyed by Equations (13) and (14). Approximation is valid for distances up to twelve nautical miles for GPS localization accuracies.

H. OPERATOR CONTROL UNIT (OCU)

The EEL OCU is comprised of a notebook computer, a 900 MHz ISM band spread spectrum modem, a 2.4 GHz video receiver and a National Television System Committee (NTSC) video - to - Universal Serial Bus (USB) converter. The EEL can be controlled using a variety of methods, such as a joystick or by using ASCII ROBO waypoint commands entered at the OCU terminal. The advantage of waypoint operation is that in the event of a command and control (C2) radio frequency (RF) link failure, the vehicle will complete its transit to the current waypoint and then simply stop.

Optionally, the EEL can be controlled using the OCU GUI developed by NSWC-PC for the amphibious crawlers as part of ONR's VSW/SZ Reconnaissance Program. The GUI provides a convenient user interface that allows the operator to command the robot while monitoring the vehicle's health and status. Figure 7 shows a screen display from the OCU.

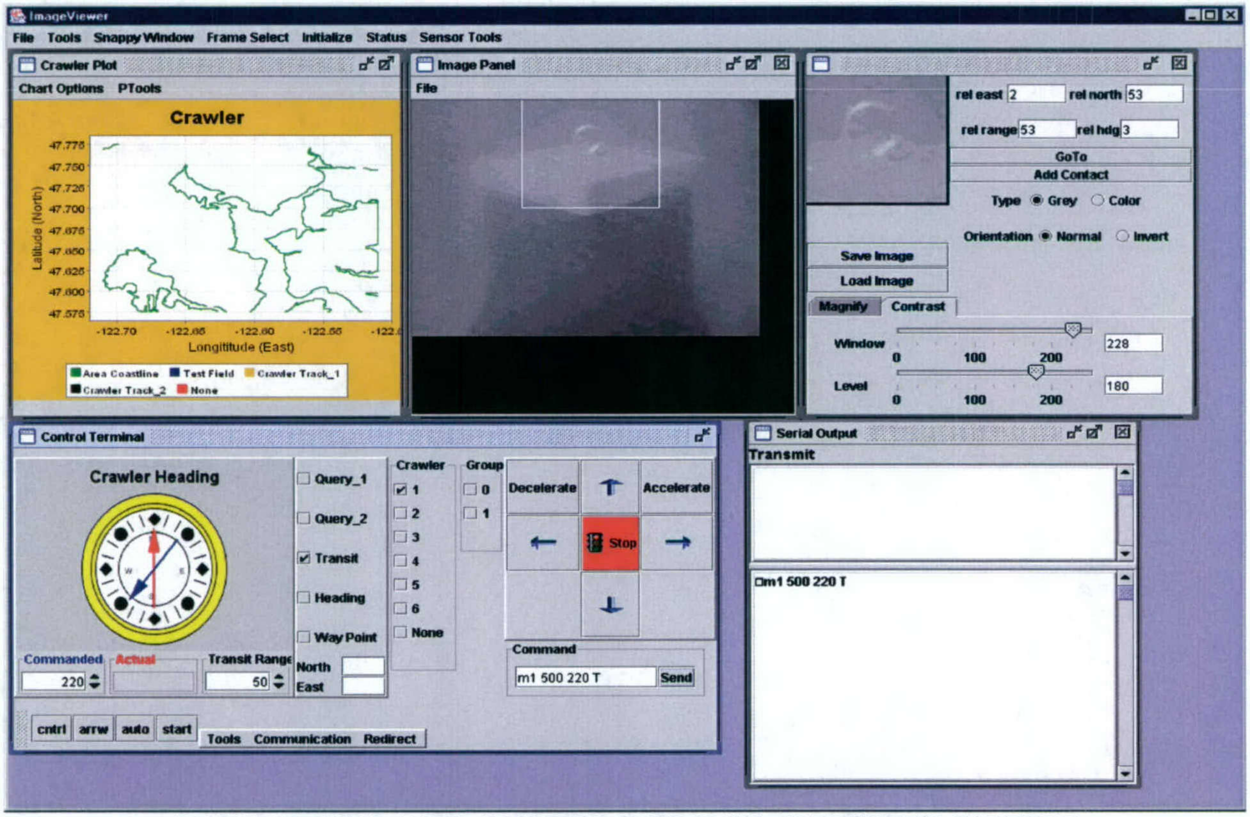

FIGURE 7. OPERATOR CONTROL UNIT (OCU) GUI SCREEN CAPTURE

II. DTRA COMPREHENSIVE HAZMAT EMERGENCY RESPONSE CAPABILITY ASSESSMENT PROGRAM (CHERCAP) INTEGRATED TECHNOLOGY DEMONSTRATION (ITD)

A. OVERVIEW

For the CHERCAP ITD, EEL was tasked to remotely inspect the top deck of an ammonia barge. In this staged scenario the vehicle could have been delivered to the barge by an unmanned aerial vehicle. However, during this exercise, EEL was manually placed on the barge because the feasibility of deploying a UGV via a UAV was not demonstrated until several weeks later during the STORK demonstration at Eglin Air Force Base, B6 Test Area.

B. EEL CHERCAP CONFIGURATION

During the ITD, EEL was controlled from a JAUS compliant operator control unit (OCU). Figure 8 provides the vehicle system configuration for the DTRA CHERCAP Exercise, showing the AFRL Tyndall Robotics/Wintec provided JAUS/ROBO translator and video server.

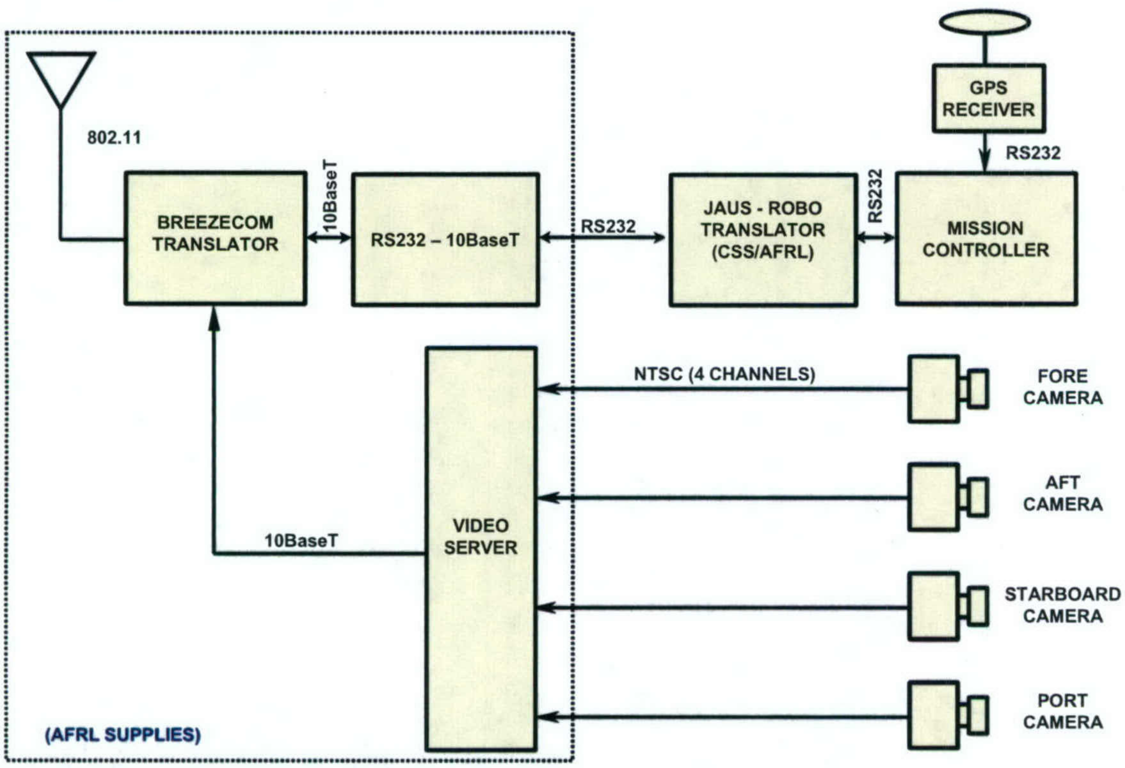

FIGURE 8. VEHICLE SYSTEM CONFIGURATION FOR DTRA CHERCAP EXERCISE

Although ROBO is a fully functional API intended for groups of remotely programmable autonomous robots, only a small subset of commands map onto the current (Version 3.0) JAUS protocol/interface standard. Specifically, commands relevant to simple teleoperation and 'supervised autonomy' are the JAUS 'wrench' primitive, 'global waypoint' command as well as 'global pose' request corresponding to the ROBO (Table 1) 'Velocity', 'Destination' and 'coordinates?' commands respectively.

C. CHERCAP DEMONSTRATION

The CHERCAP ITD successfully demonstrated the use of the JAUS protocol for controlling/tele-operating a wide variety of vehicles, including an unmanned surface vehicle (USV), UGVs, and unmanned underwater vehicles (UUVs). Figure 9 shows the NSWC-PC Autonomous Search and Hydrographics Remotely Operated Vehicle (ASH/ROV) System at CHERCAP Exercise. Figures 10 through 14 show additional JAUS compliant-robots participating in CHERCAP.

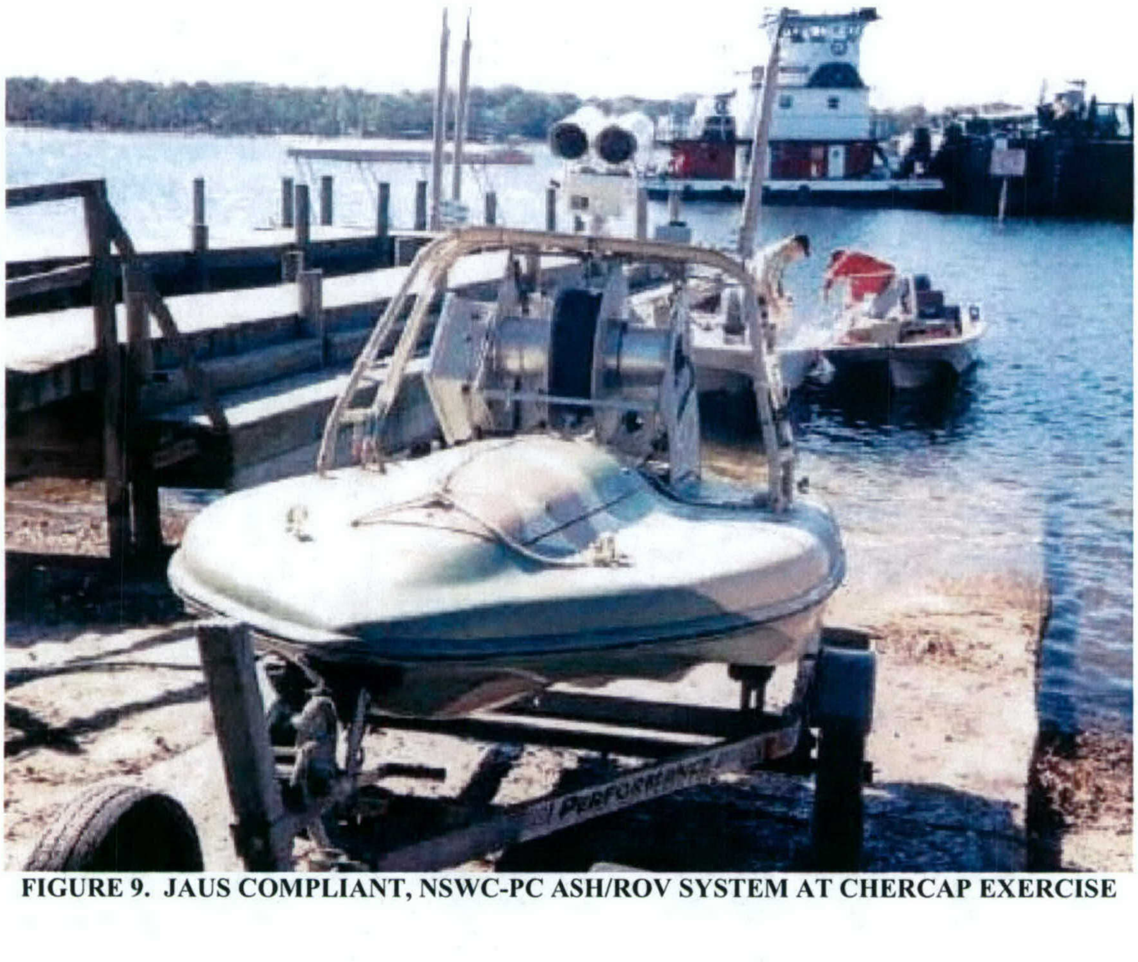

FIGURE 9. JAUS COMPLIANT, NSWC-PC ASH/ROV SYSTEM AT CHERCAP EXERCISE

FIGURE 10. ROBO COMPLIANT, NSWC-PC AMPHIBIOUS CRAWLERS AT CHERCAP EXERCISE

FIGURE 11. ROBO OR JAUS COMPLIANT, NSWC-PC EEL ROBOT AT CHERCAP EXERCISE

FIGURE 12. NSWC-PC EEL (LEFT) AND AFRL "TIGER" (RIGHT) NEXT TO JAUS OCU COMMAND VAN

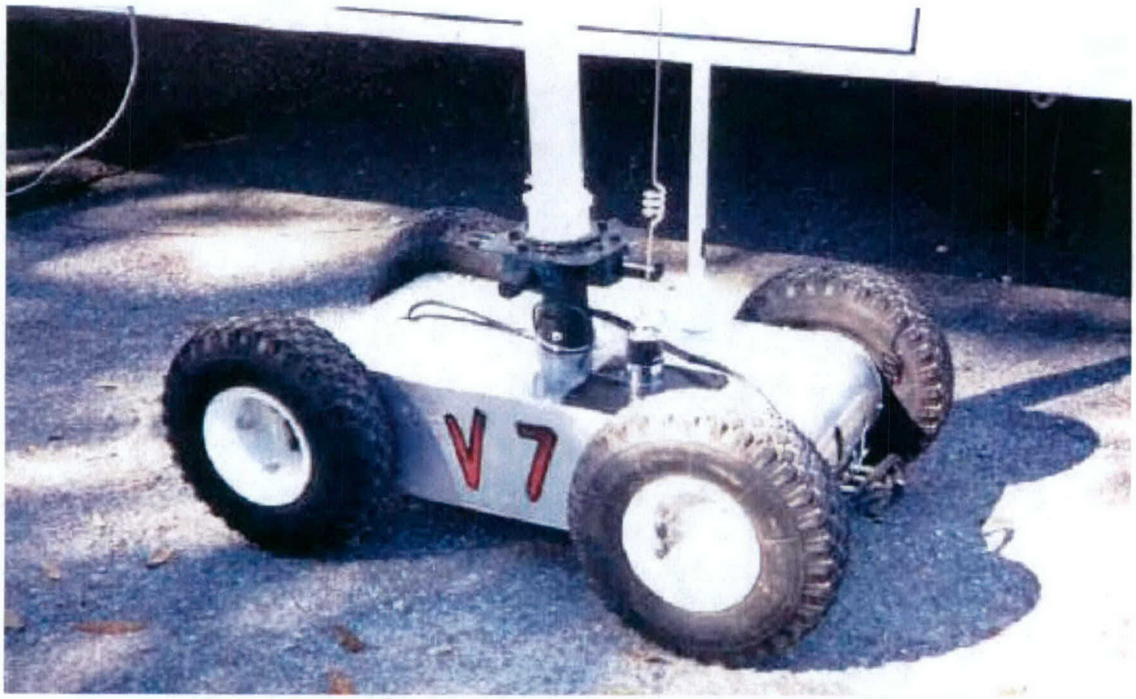

FIGURE 13. THE FOUR VIDEO CAMERAS AND C3 ANTENNAS VISIBLE IN THIS EEL CLOSE-UP

III. STORK INTEGRATION AND DEMONSTRATION

A. OVERVIEW

The STORK Project was a collaborative effort between NSWC-PC, AFRL Tyndall, Eglin UAV Battlelab, DRS Unmanned Technologies, and Wintec, Inc. A key objective of STORK was to demonstrate the deployment of a UGV by an UAV via parachute drop. Figure 14 shows two of the UGVs used in the demonstration. This section briefly summarizes the results of the integration, demonstration, and the successful deployment of an EEL UGV by a SENTRY HP UAV (Figures 15 through 17).

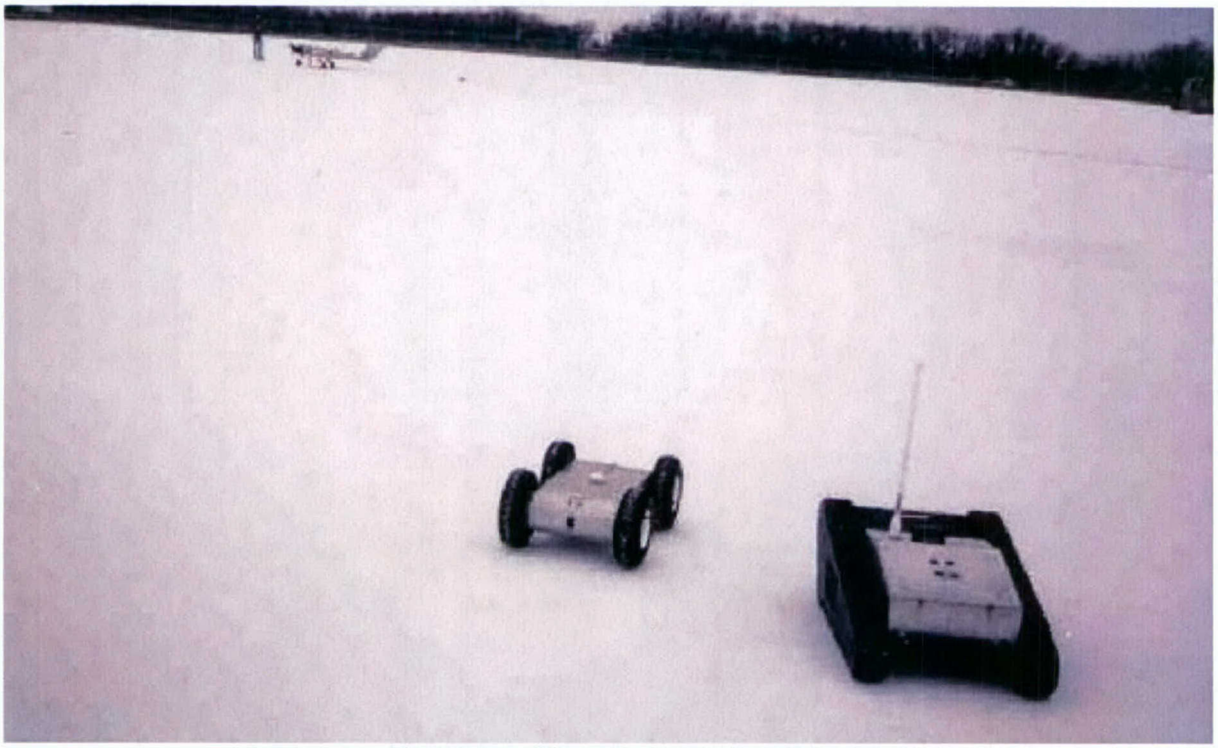

FIGURE 14. NSWC-PC EEL AND AFRL "MATILDA" AT DRS DURING STORK PROJECT INTEGRATION

FIGURE 15. NSWC-PC EEL NEXT TO DRS UNMANNED TECHNOLOGIES SENTRY HP AT EGLIN AFB

B. INTEGRATION TESTING

Preliminary integration testing held in Mineral Wells, TX determined parachute placement on the EEL. A single point harness drop resulted in right front gearbox damage on impact due to large tangential shock, shown in Figure 16. It was determined that a four point parachute harness (Figures 17 and 18) works better than one point for distributing the impulse forces to all wheels. Axial and radial axle shock was successfully mitigated at an impact velocity estimated upwards of 50 feet per second.

FIGURE 16. EEL AFTER FIRST PARACHUTE DROP (SINGLE POINT HARNESS), MINERAL WELLS, TX

FIGURE 17. EEL WITH PARACHUTE AND FOUR POINT HARNESS ATTACHED (FRONT VIEW)

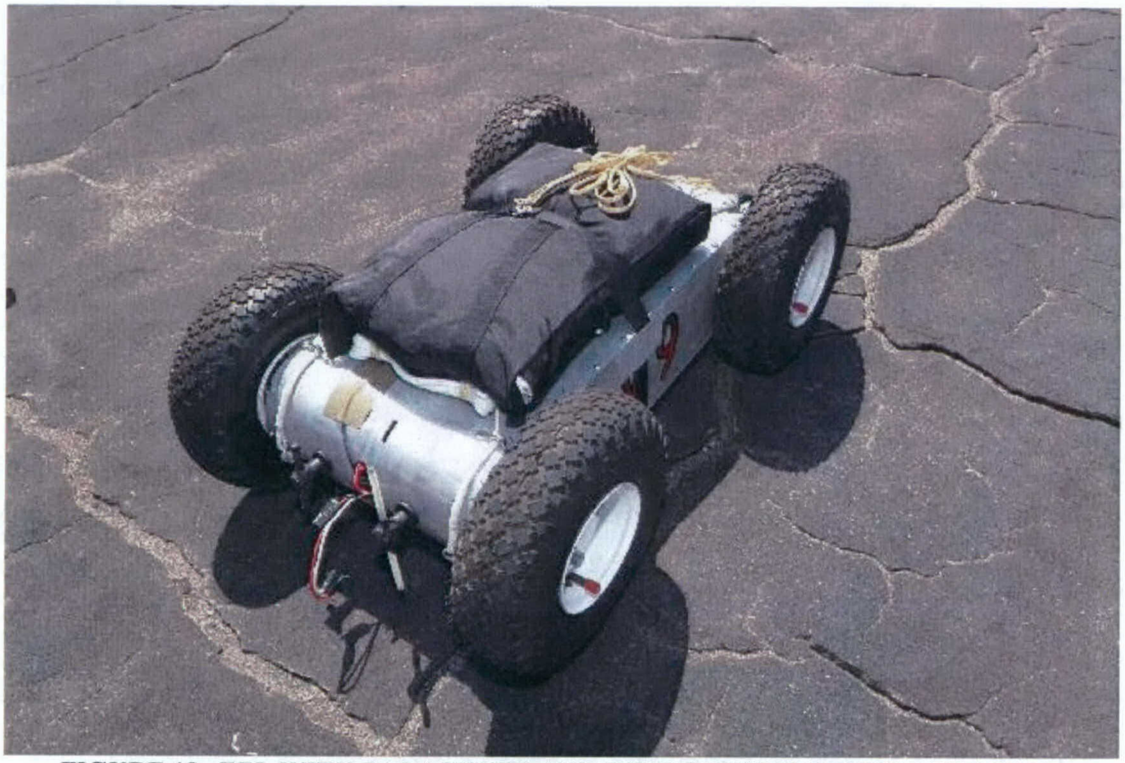

FIGURE 18. EEL WITH PARACHUTE AND FOUR POINT HARNESS ATTACHED (REAR VIEW)

C. DEMONSTRATION RESULTS

EEL survived three consecutive parachute drops, each resulting in an average impact speed of 20 feet per second on hard surfaces. Figures 19 and 20 show the attachment of the EEL to the Sentry HP for the STORK demonstration.

FIGURE 19. EEL SECURED UNDER SENTRY HP, READY FOR TAKE-OFF

FIGURE 20. SENTRY HP UAV AND EEL UGV AIRBORNE, 20 MARCH 2003

The demonstration consisted of three parachute drops from the Sentry HP. The first drop (Figures 21 and 22) resulted in an accidental free fall of 25 feet when the parachute separated from the harness. Even after the free fall the EEL operated properly.

FIGURE 21. EEL AT LANDING SITE AFTER AN ACCIDENTAL FREE FALL OF 25 FEET

FIGURE 22. EEL CLOSE-UP AT LANDING SITE DOES NOT SHOW ANY VISIBLE DAMAGE

Following further modification to the UAV/UGV interface the second and third drops were successful. The first successful delivery of a fully functional UGV by a UAV was accomplished (Figure 23).

FIGURE 23. FULLY FUNCTIONAL NSWC-PC EEL EXECUTES WAYPOINT COMMANDS.

The first generation EEL has operated in temperatures varying from 15-95 degrees Fahrenheit. Furthermore, during cold weather operations it was observed that EEL maintained good traction on icy surfaces.

IV. SECOND GENERATION EEL: TACTICAL EXPENDABLE REMOTELY PROGRAMMABLE ROBOT

A. OVERVIEW

The Tactical Expendable Remotely Programmable (TERP) Robot is the second generation EEL vehicle. Similar to EEL, TERP was designed using COTS components to minimize cost. The TERP was designed to provide greater payload carrying capability, and therefore is larger than its predecessor.

B. DESIGN

The EEL's initial design goals were to maximize the volume against set size limits as well as the vehicle collision strength inside weight limit restrictions. Design and prototyping of a second generation EEL (Figures 24-26) were completed in early July 2003. The solid connection between the wheel axles and motor gearboxes was replaced by adjustable clutches (visible in Figure 25) that slip when subject to large tangential shock. With the footprint and weight limits removed, notable improvements include a slightly wider, shorter wheelbase, larger diameter wheels, and additional batteries for larger payload capability and improved mobility on uneven terrain.

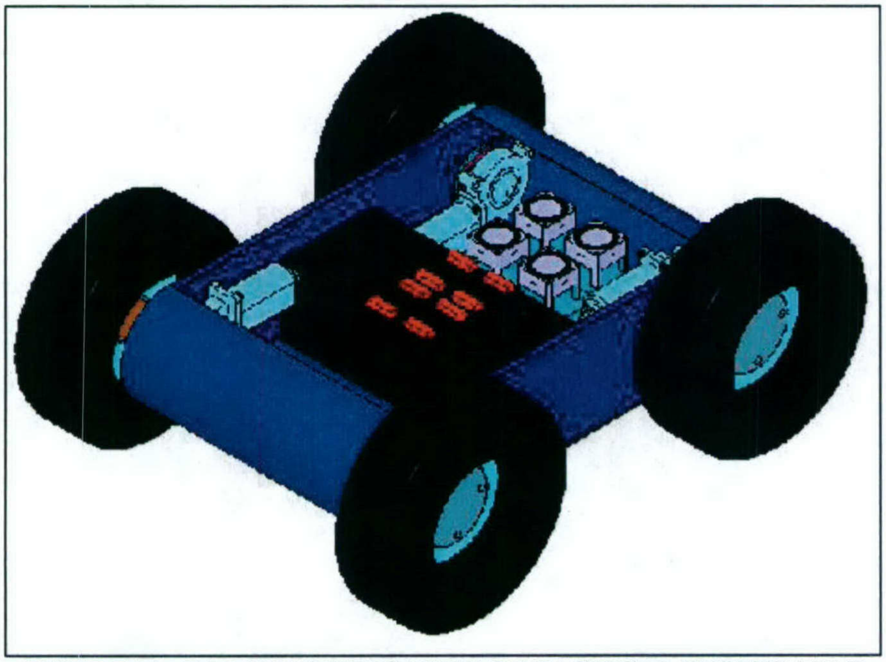

FIGURE 24. CAD ISOMETRIC VIEW OF TACTICAL EXPENDABLE REMOTELY PROGRAMMABLE (TERP) ROBOT

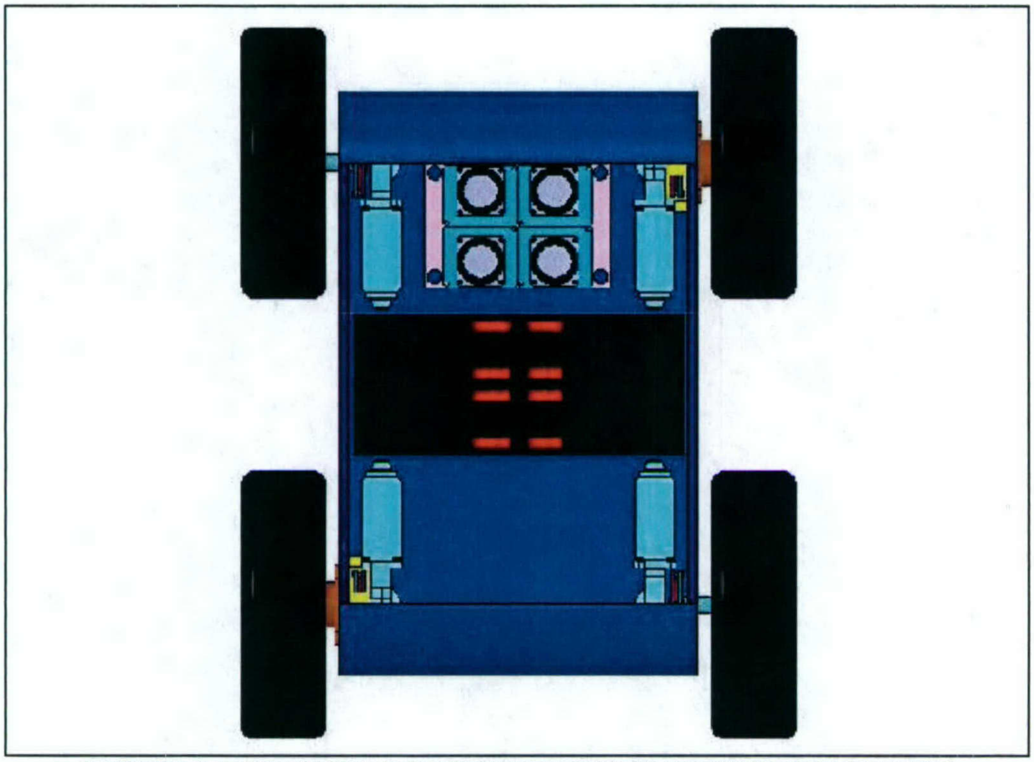

FIGURE 25. TOP VIEW OF TERP DEPICTS CLUTCHED GEARBOXES AND ADDITIONAL BATTERIES

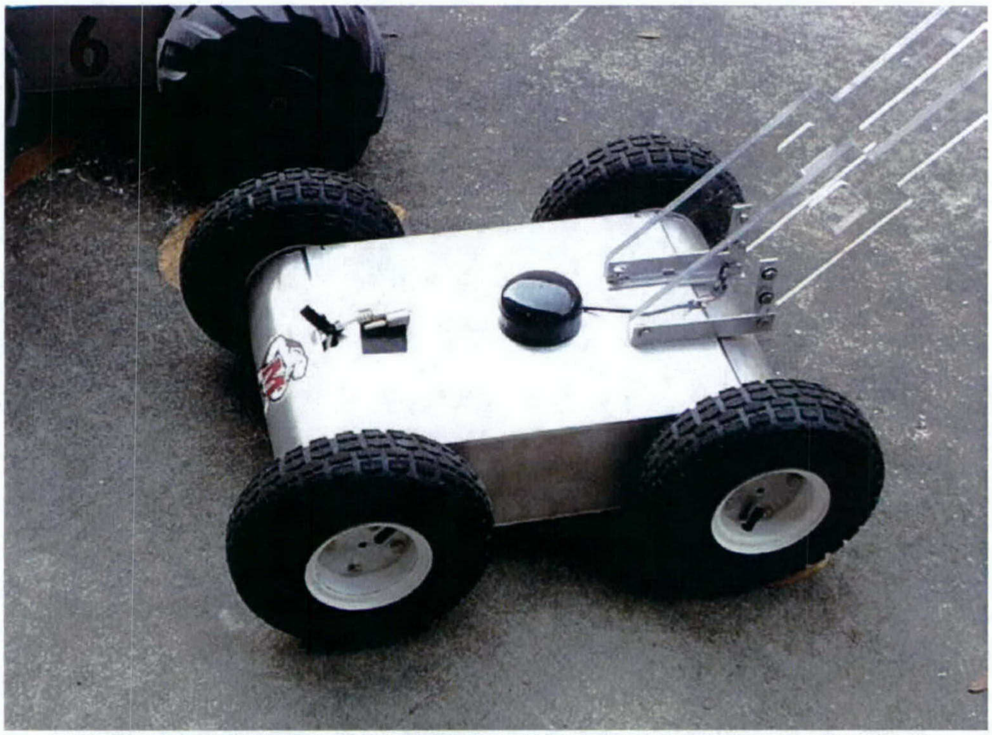

FIGURE 26. FIRST PROTOTYPE OF THE SECOND GENERATION EEL: TERP ROBOT

Overall, the configuration of the TERP Robot remained very similar to the first generation EEL:

- Aluminum chassis with passive axels inside externally mounted bearings
- Four GM* power window motors and 10.5-inch pneumatic tires for locomotion
- 12-Volt, 7.2 Ah lead acid rechargeable batteries for power source
- 16-bit AMD 188ES-based control computer and four reversible motor speed controllers
- Tri-axial flux gate compass and WAAS enabled GPS original equipment manufacturer (OEM) unit for positioning
- 900 MHz ISM spread spectrum RF modem for command and control
- Inexpensive pan and tilt color video camera with 2.4 GHz RF video transmitter

* The appearance of trade names in this document does not constitute endorsement by the Department of Defense, the Navy, or the Dahlgren Division Naval Surface Warfare Center – Panama City (NSWC - PC)

V. DTRA CRITICAL INFRASTRUCTURE PROTECTION COMMON OPERATING PICTURE (CIPCOP) DEMONSTRATION AND THE THIRD GENERATION EEL: RYOT

A. OVERVIEW

The third generation EEL vehicle: "RYOT" participated in a second 2003 DTRA sponsored ITD at NSWC, Dahlgren, VA site on 24 July 2003. The role of the robot in this demonstration was to carry an automatic chemical agent detector alarm (ACADA) chemical sensor to the site of a possible chemical spill while providing standoff distance for the human first responders.

B. BACKGROUND

NSWC-PC personnel arrived at Dahlgren in mid July with the second generation EEL (TERP) robot. The EEL requirement for the demonstration was to transport a nominal 27-pound payload (the ACADA sensor) to the suspected chemically contaminated area. Since sensor and robot integration was beyond the scope of the exercise, the ACADA also necessitated its own control computer, a wireless Institute of Electrical and Electronics Engineers (IEEE) standard 802.11-compliant local area network (LAN) connection and a GPS enabled cell phone. Due to the nature of the four-wheel drive, slip steered locomotion on rubber tires and higher than anticipated total payload weight, TERP exhibited marginal mobility performance particularly when executing pivot turns.

C. THIRD GENERATION EEL: RYOT

Previous experiments showed that wider plastic wheels could greatly improve mobility on grass and sand. Figures 27 and 28 depict three different EEL wheel arrangements.

FIGURE 27. FIRST GENERATION EELS WITH PNEUMATIC RUBBER TIRES AND PLASTIC WHEELS

FIGURE 28. EEL WITH PLASTIC WHEELS (LEFT) AND WIDER TERP WITH RUBBER TIRES (RIGHT)

To overcome mobility limitations, a decision was made to build a new chassis that would accommodate larger plastic wheels. The new chassis consists of a simple box with two passive axles near the longitudinal ends of the enclosure. A suitable size plastic box could not be located in time, so a RYOBI* cordless drill carrying case was used instead (Figure 29).

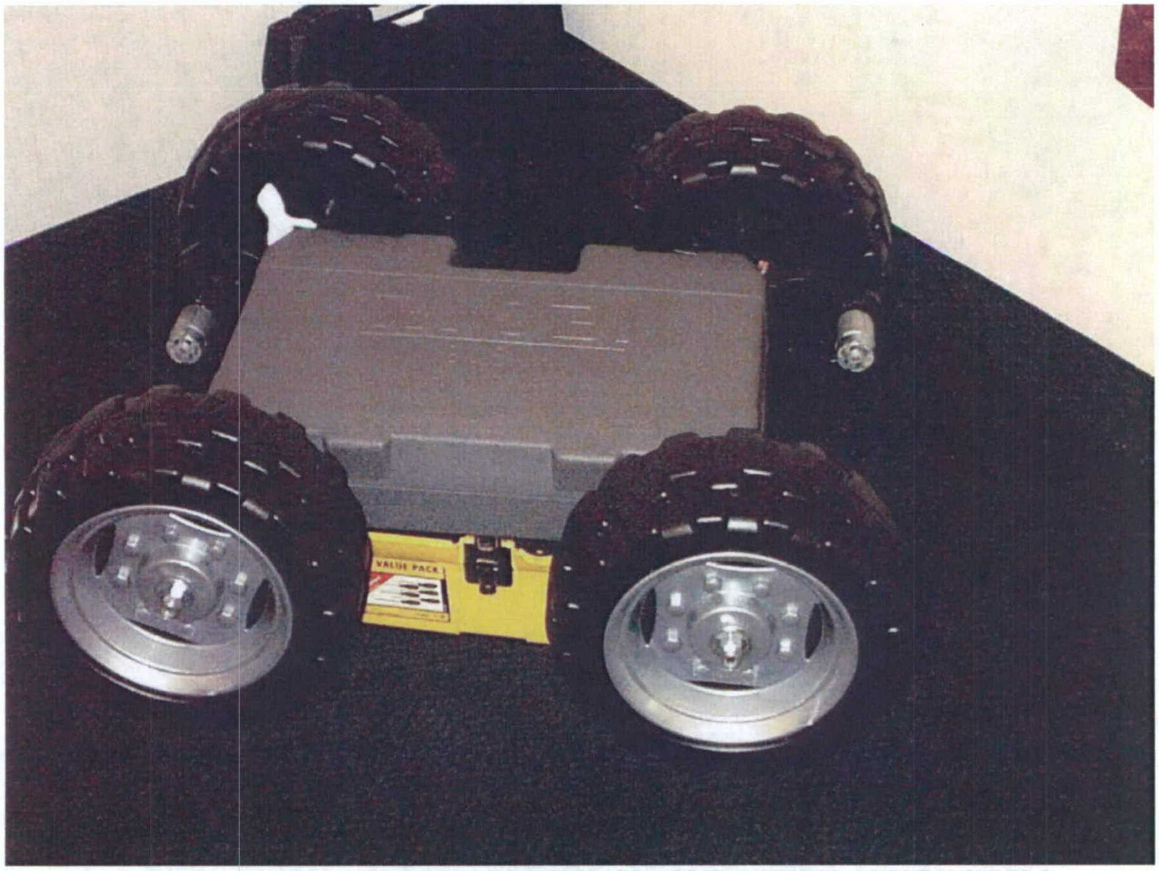

FIGURE 29. THIRD GENERATION EEL CHASSIS WITH PLASTIC WHEELS

The TERP robot electronics were repackaged in a weatherproof electrical enclosure and attached on top of the plastic chassis. The RF C2 modem, video camera/transmitter and GPS unit were secured on top of the electronics enclosure. The ACADA sensor and support equipment were mounted close to the vehicle geometric center. On 19 July 2003, the new vehicle was tested and confirmed excellent mobility regardless of surface type.

The stylized RYOBI logo on the vehicle chassis is partially covered by the electronics enclosure. The remaining letters read RYOT. The third generation EEL will be referred to as "RYOT". Although capable of a maximum speed of two meters per second, RYOT's nominal running speed was set to approximately one meter per second.

* The appearance of trade names in this document does not constitute endorsement by the Department of Defense, the Navy, or the Dahlgren Division Naval Surface Warfare Center – Panama City (NSWC - PC)

D. CIPCOP DEMONSTRATION

Figures 30-33 depict the RYOT loaded with the chemical sensor payload during the CIPCOP demonstration. RYOT met all requirements and successfully completed the CIPCOP demonstration.

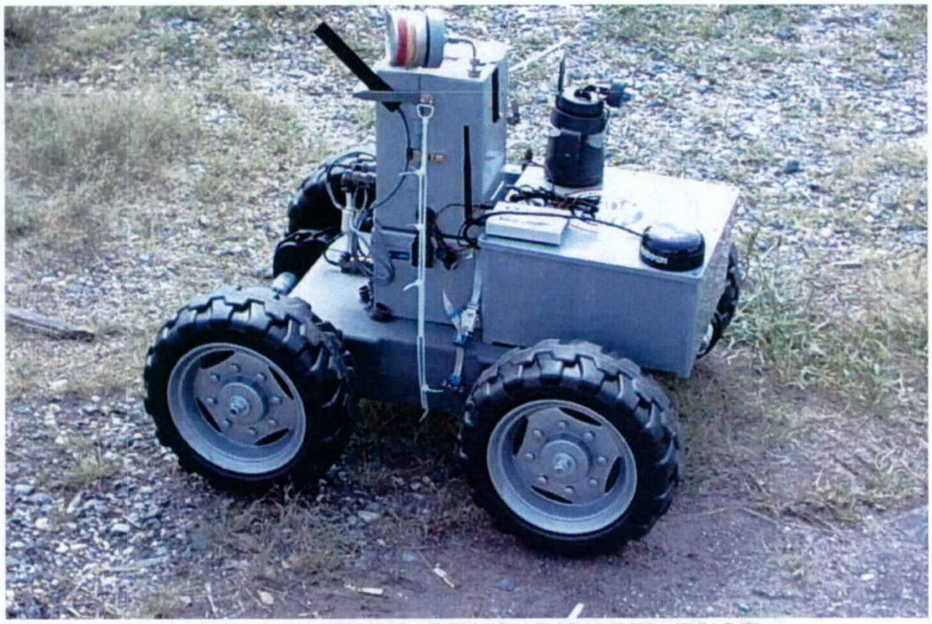

FIGURE 30. THIRD GENERATION EEL (RYOT) WITH CHEMICAL SENSOR PAYLOAD

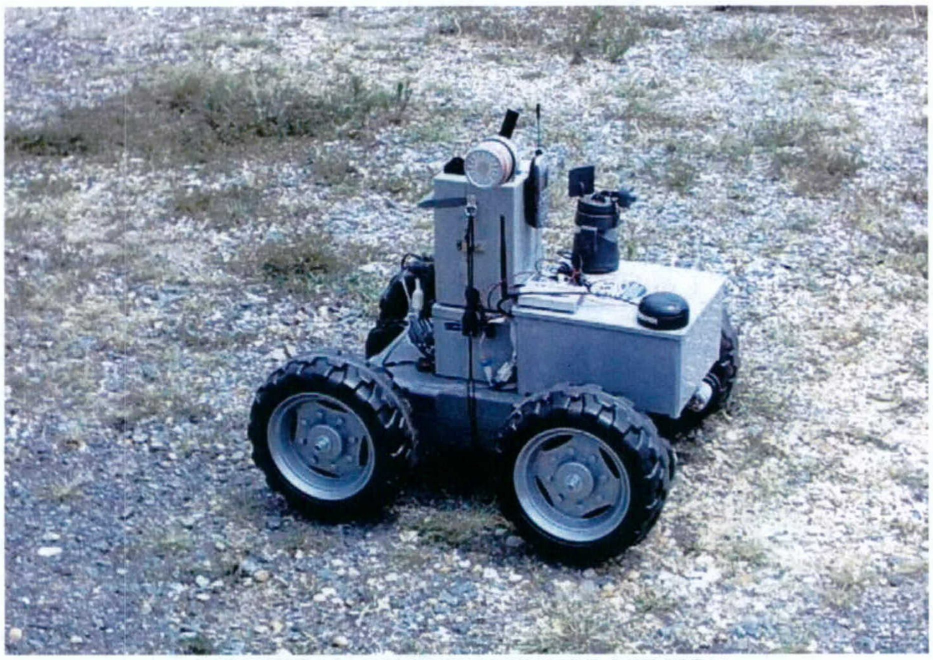

FIGURE 31. RYOT READY FOR MISSION

FIGURE 32. RYOT MOBILITY TESTS 19 JULY 2003

FIGURE 33. RYOT AFTER COMPLETING ITS MISSION ON 24 JULY 2003

E. RYOT 2

Upon return to NSWC-PC, the Surf Zone Robotics team constructed one additional third generation EEL. Figure 34 compares the RYOT with the TERP. The RYOT 2 chassis with EEL generation one electronics plate is shown in Figure 35.

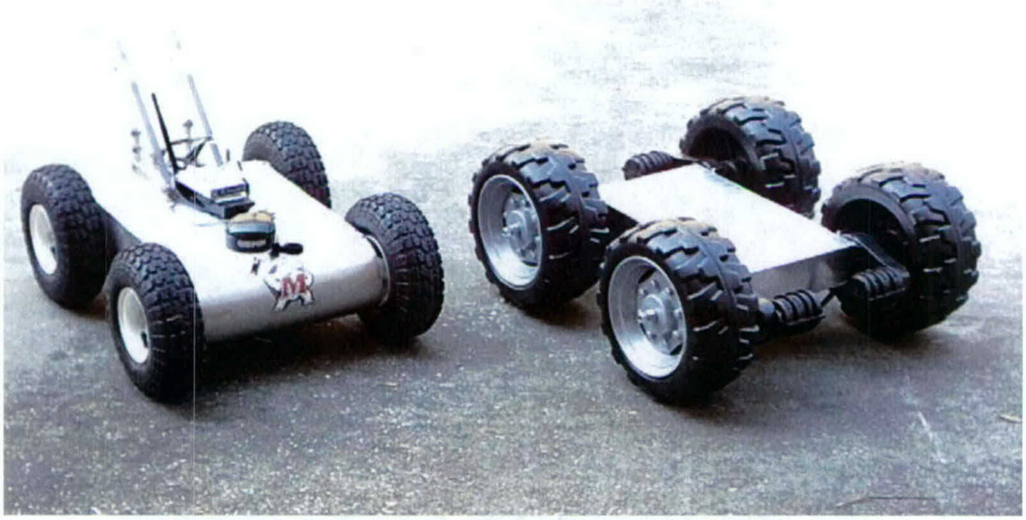

FIGURE 34. EEL GENERATION TWO: TERP (LEFT), AND EEL GENERATION THREE: RYOT (RIGHT)

FIGURE 35. RYOT 2: LIGHTWEIGHT ALUMINUM CHASSIS

VI. CONCLUSION

A. SUMMARY

The Surf Zone Robotics Team conceptualized and built three generations of inexpensive four-wheel drive, slip steered vehicles. The EEL generation one demonstrated survivability after collisions at speeds in excess of 20 feet per second and became the first UGV successfully delivered by a UAV. EEL generation two has improved mobility and clutched transmission for protection against large tangential shocks. Although not as rugged, EEL generation three has larger payload capability and exhibits excellent all terrain mobility. All EELs leverage the control system and ROBO API previously developed for the Surf Zone Crawling Robots.

The ROBO API provides the means to remotely program, command, control and communicate with a group of robots over various narrow band channels. It also allows dynamic development of vehicle behavior libraries suitable for surf zone MCM operations. Regardless of syntax specifics, this concept of operationally programmable autonomous littoral systems (i.e. OPALS) can be extended to other unmanned vehicles.

NSWC-PC plans to enhance the EEL communication protocol; in addition to ROBO, operators will transparently decode pertinent JAUS messages without the use of intermediate translators. This enhancement will satisfy the Office of the Secretary of Defense (OSD) mandate for ground vehicles and could promote interoperability with other unmanned systems and OCUs. One notable consequence of the integration work for the DTRA CHERCAP exercise is that NSWC-PC was invited to participate in the JAUS Reference Architecture Working Group. Scientists at NSWC-PC hope to contribute to JAUS interface future development to support full autonomy and remote programming capability (the OPALS concept) similar to the NSWC-PC control system and ROBO API developed for the Surf Zone Amphibious Crawlers.

B. RECOMMENDATIONS AND FUTURE WORK

The design of the EEL robots described herein has been driven by size/weight limits and stringent shock survivability requirements. Emphasis was placed on simplicity and use of COTS parts to minimize prototyping and fabrication cost. The EELs perform well on flat or inclined asphalt, concrete, gravel and clay surfaces. The current vehicles can be used as test beds for future work to develop control strategies, navigation algorithms, and cooperative behaviors for swarms of autonomous robots.

Essential near term improvements include ultrasound sonar for autonomous obstacle detection/avoidance and possibly an omni directional antenna for the video link. Higher power density batteries are also being considered for extending the vehicle run /standby time: specifically, rechargeable lithium ion batteries for normal use and non-rechargeable lithium/manganese dioxide batteries for the missions where the robot would be expended. Furthermore, additional weatherproofing for EEL's internal electronics is necessary to ensure reliable operation in all weather conditions.

For adequate mobility in natural environment terrains such as sand, grass, soft silt and mud the EEL's creators recommend investigating the possible benefits of larger ridged wheels that employ the Archimedes screw effect during turns, and for improved efficiency consider elliptical roller omni-directional wheels. Additionally, segmented, articulated chassis with at least one and a half degrees of freedom between sections could prevent high centering on small proud objects such as curbs and pipes.

VII. BIBLIOGRAPHY

1. Gavrilash, M., Kucik D., Threatt S., "Surf Zone Robotics/ Control Systems Development," Office of Naval Research, Ocean Engineering and Marine Systems, Fiscal Year 1999, 2000, 2001 and 2002 Annual Progress Reports.

2. "The Joint Architecture for Unmanned Ground System (JAUGS) Working Group, Volume 2, Reference Architecture Specification," Version 3.0 Draft, June 24 2002.

3. Bernstein, C., Connolly, M., Gavrilash, M., Kucik, D., Threatt, S., "Demonstration of Surf Zone Crawlers: Results from AUV Fest," Fifth Symposium on Technology and the Mine Problem hosted at Naval Postgraduate School, Monterey, CA, April 2002.

4. Common Control Language Group, "High-Level Group Control for Interoperability and Group Cooperative Missions in Very Shallow Water and Surfzone," Working White Paper, June 2001.

5. Gage, D.W., "Development and Command-Control Tools for Many-Robot Systems," Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE) Microrobotics and Micromechanical Systems, Philadelphia PA, Volume 2593, pp 121-129, 23-24 October 1995.

6. Duarte, C., "Port Arbitrated Behavior Common Control Language Outline," May 2000.

7. Komerska, R., Chappell, S., Peng, L., Blidberg, R., "Generic Behaviors as Interface for a Standard AUV Command & Monitoring Language," The Autonomous Undersea Systems Institute (**AUSI**), Working Draft Version A3, September 1999.

8. Phoha, S., "A Proposal for a Generic Behavior Message-Passing Language For Collaborative Missions of Heterogeneous Autonomous Underwater Vehicles," Penn State University/ Applied Research Laboratory (PSU/ARL), Memo to The Autonomous Ocean Sampling Network (**AOSN**) Community, 1999.

9. Gavrilash, M., "Distributed Underwater Network Evaluator (DUNE), An Object Oriented Framework for Modeling Swarms of Autonomous Vehicles," Modeling and Simulation Workshop, Naval Postgraduate School, Monterey, CA, 1998.

10. "Tactical Control System Segment to Air Vehicle Standard Segment Interface Design Description," NSWCDD TCS-229, August 1997.

11. "Joint Architecture for Unmanned Ground Systems (JAUGS)," Version 1.0 Draft, Unmanned Ground Vehicles/Systems Joint Project Office, AMSAM-DSA-UG; 1999

12 Gage, D.W., "Cost-Optimization of Many-Robot Systems," Proceedings of SPIE Mobile Robots IX, Boston MA, Volume 2352, pp 260-266, 2-4 November 1994.

ACKNOWLEDGEMENTS

Defense Threat Reduction Agency
Matt Holm

Office of Naval Research
Dr. Tom Swean (321 OE Ocean Engineering and Marine Systems)

Coastal Systems Station, NAVSEA, Panama City
Mitch Gavrilash, Code R23 (EEL Concept, Powertrain and Control System Design)
Dave Wilbanks, Code A62 (Mechanical Design)
Jeff Blankenship, Code E31 (Shock Testing and Analysis)
Lee Cofer, Code R11 (Electronic Design and Implementation)
Dan Kucik, Code R11 (System Design and Layout)
Don Hutchison, Code A92 (Electronics Prototyping)
Ron Reck, Code E51 (Mechanical Prototyping)
Stan Threatt, Code R11 (Operator Control Station Design and Implementation)
Eric Tuovilla, Code R12 (RF Antennas Design and Implementation)
Chuck Bernstein, Code R23 (Project Guidance and Management Support)
Phil Bernstein, Code R11 (Management Support)
Rob Cole, Code XPA0V (Professional Video and Photography)

AFRL Tyndall Robotics
Major Tim Schulteis

Eglin UAV Battlelab
Captain John Price

DRS Unmanned Technologies
Gary Skaggs

Wintec, Inc.
Woody English

Applied Research Associates, Inc
Randy Plank

3e Technologies International, Inc.
Benga Erinle

USSOCOM

720th Special Tactics Group

6th Ranger Training Battalion

APPENDIX A

ROBO COMMAND SETS

TABLE A-1. ROBO COMMAND SET: COMMANDS WITH ARGUMENTS (27 TOTAL)

Command	Description	Units	Example/Comments
A	**Adjust** relative seconds counter	Seconds	Start_count A<CR> (see "seconds" command as wel
			200 A<CR> (sets current relative timer to 200 second
B*	Move **Back** (open loop direction control)	Seconds	Time_to_travel B<CR>
			3 B<CR> (back up for 3 seconds)
C	RTCM differential **Correction**	Integer	Number_of _bytes C<CR>
			45 C<CR> (transfer 45 bytes to the GPS receiver)
D*	**Destination** (waypoint in absolute coords)	DegreesMinutes. Fract_Min*100000	Latitude Longitude D<CR>
			300935623 –854623174 D<CR> (Positive LAT. implie
F*	Move **Forward** with PID control	Meters	Distance_to_travel F<CR> (see "done" command as
			5 F<CR> (move forward 5 meters on heading set by
G	Assign vehicle to **Group**	Integer	Group_number G<CR>
			3 G<CR> (assign vehicle to group number 3)
H*	Acquire desired **Heading**	Degrees	Heading H<CR> (H calls L or R commands; see "turn
			270 H<CR> (vehicle acquires West heading after piv
I*	**Image** transmission (binary data)	Integer	Number_of_bytes I<CR>
			1500 I<CR> (transfer 1500 bytes of binary data from
			Initiated by sp_image command to SP
K	Set heartbeat interval to **Keep** alive	Seconds	Number_of_seconds K<CR> ("k" must be received o
	(Software watchdog timer)		120 K<CR> (the lower "k" command must be receivec

RTCM = *Radio Technical Commission for Maritime Services*

** The highlighted commands represent the minimum command set required for basic vehicle operation*

TABLE A-1. ROBO COMMAND SET: COMMANDS WITH ARGUMENTS (27 TOTAL), CONT.

Command	Description	Units	Example/Comments
L	Turn **Left**	Degrees	Number_of_degrees L<CR>
			90 L <CR> (turn left 90 degrees)
***LBL_MASTER**	**LBL Master** Transmitter Location	DegreesMinutes. Fract_Min*100000	Latitude Longitude LBL_MASTER<CR>
*Used only underwater			301034653 –854523124 LBL_MASTER<CR>
***LBL_SLAVE(1,2)**	**LBL Slave(s)** Transmitter Location	DegreesMinutes. Fract_Min*100000	Latitude Longitude LBL_SLAVE<CR>
*Used only underwater			300935623 –854623174 LBL_SLAVE<CR>
M	**Move** to waypoint (range/bearing format)	Meters*10 & Degrees	Range Bearing M<CR>
			200 180 M<CR> (transit 20 meters due South)
Implemented as*MV 12.29.03			Unlike the T or W commands, this command does no
			It is intended for obstacle avoidance
N	Define boundary past where **Not** allowed	Integer	A*100 B*100 C*100 INDEX
			If the inequality a*X+b+cZ>0 is not true the vehicle cr
			0 0 0 index N<CR> clears the boundary
			Command clear_Bounds<CR> is equivalent to 0 0 0 i
O*	Sets **Origin**. This is the (0,0) point for relative X/Z(North/East) coordinate system	DEGREESMINUTES. FRACT_MIN*100000	Latitude Longitude O<CR>
			300935623 –854623174 O<CR>
P*	Set vehicle's current relative **Position**	Meters*10	North Down East P<CR> (with respect to origin set b
			343 12 123 P<CR>
Q	**Query** type		* Currently (12.29.03) not implemented (SEE Q2 inst

** The highlighted commands represent the minimum command set required for basic vehicle operation*

TABLE A-1. ROBO COMMAND SET: COMMANDS WITH ARGUMENTS (27 TOTAL), CONT

Command	Description	Units	Example/Comments
R*	Turn **Right**	Degrees	Number_of _degrees R<CR>
			60 R <CR> (turn right 60 degrees)
S*	**Stops** vehicle/Abandons current waypoint	N/A	S<CR> (Note if only the two characters S<CR> are is STOP command also)
T*	**Transit** to waypoint (range/bearing fmt.)	Meters*10 & Degrees	Range Bearing T<CR>
			200 180 T<CR> (transit 20 meters in the South directi
U*	**Unknown**/Object type	Integer	Sensor_number U<CR> (usually unsolicited message
			1 U<CR> (collision with proud object signaled by bum
V*	Set **Velocity**	Integer	Left_motor_speed Right_motor_speed Nominal_Spee
W*	Transit to **Waypoint** (relative X/Z format)	Meters*10	North East W<CR>
			345 124 W<CR> (waypoint 34.5 meters North, 12.4 m
X	Set vehicle's relative **North (X)** position	Meters*10	North X<CR>
			234 X<CR> (set vehicle's current position 23.4 meter
Y	Set vehicle's depth/altitude	Meters*10	Down Y<CR>
Z	Set vehicle's relative **East (Z)** position	Meters*10	East Z<CR>
			567 Z<CR> (set vehicle's current position 56.7 meter

** The highlighted commands represent the minimum command set required for basic vehicle operation*

TABLE A-2. ROBO COMMAND SET: COMMANDS WITHOUT ARGUMENTS (23 TOTA

Command	Description	Units	Example/Comments
a	**accelerate**	N/A	Increment speed by 10. Nominal running speed is 40 for TAR2, 150 for E
b	**backup**	N/A	Unconditional ROV command
C*	Request vehicle's **coordinates**	N/A	c<CR>
	Reply	Degrees-Minutes. Fract_Min*100000 & Integers	Latitude Longitude Fix_Quality Number_of_Satellites Age<CR><LF>
	Example		300935623 –854623174 1 5 0<CR><LF>
d	**decelerate**	N/A	Decrement speed by 10. Nominal running speed is 40 for TAR, 150 for El
e	Request **energy** level	Percent	Only in ROV mode (11.25.02); * not implemented for EEL
f	**forward**	N/A	Unconditional ROV command
g	Request **group** currently assigned to	Integer	Only in ROV mode (11.25.02)
h	Request intended **heading**	N/A	h<CR>
	Reply	Memory Location	Intended_Heading_Memory_Address<CR><LF>
j	**jog** (travel on current compass heading)	N/A	Unconditional ROV command
k	**keep** heartbeat going (software WD)	N/A	Once initiated, the command has to be repeated to prevent mission ABO
l	Pivot turn to the **left**	N/A	Unconditional ROV command
n	**navigation** boundary and obstacle flags	Integers	REPLY Nav flags: collision 0, boundary 0<CR><LF> see collision&bound

** The highlighted commands represent the minimum command set required for basic vehicle operation*

TABLE A-2. ROBO COMMAND SET: COMMANDS WITHOUT ARGUMENTS (23 TOTAL), CON

Command	Description	Units	Example/Comments
O*	**orientation**	N/A	o<CR>
	Reply	Degrees	Compass_Heading Pitch Roll<CR> (for TAR2 and EEL)
	Example		230 22 10<CR><LF>
P*	Current vehicle's relative **position**	N/A	p<CR>
	Reply	Meters*10	North Down East<CR><LF>
q(uiet)	**quiet** mode on/off	N/A	Suppress any replies from vehicle
r	Pivot turn to the **right**	N/A	Unconditional ROV command
s	**Stop** motors	N/A	Note if only the two characters s<CR> are issued, mission is ABORTED
t	Request for current relative **time**	N/A	t<CR>
	Reply	Seconds	second_timer_value<CR><LF>
	EXAMPLE		3600<CR><LF> (relative timer reads one hour)
V(EL)	Current **velocity**	Integers	vel<CR>
	Reply		Speed settings 150 150<CR><LF>
w	**Well** being, currently (12.29.03) **wheel** counts	N/A	w<CR>
	Reply	Integers	Status_byte Health_byte<CR> (see Q2)
x	Request **North (x)** position value	Meters*10	No actual reply, but the value is pushed on the interpreter stack
y	Request **Yaw** (Compass) value	Degrees	No actual reply, but the value is pushed on the interpreter stack
z	Request **East (z)** position value	Meters*10	No actual reply, but the value is pushed on the interpreter stack

** The highlighted commands represent the minimum command set required for basic vehicle operation*

APPENDIX B

JOYSTICK TELE-OPERATION

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED

APPENDIX B. JOYSTICK TELE-OPERATION

Vehicle velocity can be set by issuing the "V" (Velocity) command.

Example: "20 40 0 V". The three arguments are Left Motor Speed, Right Motor Speed, and Nominal Vehicle Speed respectively. The "V" assigns the values to corresponding variables; it does not initiate any vehicle action. For forward movement, the velocity setting command should be followed by the "f" (forward) command: "20 40 0 V f".

The speed range (for RYOT) is –100 (full speed reverse), zero (stop), +100 (full speed forward). For joystick control the Nominal Vehicle Speed does not need to be set to a particular value; it can and should be left set to zero.

"0 0 0 V f" would essentially stop the vehicle, but in order to avoid creep, an explicit "S" command should be issued, when the joystick is at rest.

50 50 0 V f
Half speed straight ahead

S
Stop

-25 -25 0 V f
Quarter speed back

-33 +33 0 V f
Left pivot turn at one third motor speed

40 20 0 V f
An arc to the right

S
Stop

Note: All strings are terminated by a carriage return (x0D) with no line feed.

APPENDIX C

DEFINING SEARCH ZONES AND FORBIDDEN AREAS BY LINE INEQUALITIES

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED

APPENDIX C. DEFINING SEARCH ZONES AND FORBIDDEN AREAS BY LINE INEQUALITIES

Assuming a relative coordinate system has been established and that X is positive in the North direction and Z is positive in the East direction.

Operator Control Station Setup

User selects two points (X1, Z1) and (X2, Z2) to define a line, and a third point (Xp, Zp) on either side of the line to indicate the allowed half plane.

For the general case, we first determine the slope and Z intercept for the line.

$$dX = X2 - X1$$

$$dZ = Z2 - Z1$$

The slope is $a = dZ/dX$

The Z intercept is $b = Z1 - a{*}X1$

Next we look at the inequality $Zp>a{*}Xp+b$

If true we set $a = -a$, $b = -b$ and $c=1$

$c{*}Z - a{*}X - b > 0$ is true in that half of the plane

i.e. $\mathbf{Z>a{*}X+b}$

If false then $c=-1$

$-c{*}Z + a{*}X + b>0$ is true

or $\mathbf{Z<a{*}X+b}$

The North-South lines are handled separately since the slope is infinite.

If $dX = 0$

set $b = X1$

and $c = 0$

Next check Xp<X1

if true set a = -1

else a = 1 and b = -b

Evaluating c*Z+a*X+b>0 is equivalent to **X<b** and **X>b** respectively for the special case

Crawler Setup

The vehicle is given the **a,b and c** coefficients to be placed in an indexed array. Command is: **# # # # N<cr>**, where the four numbers are **a, b, c and index.** At every position update the vehicle verifies that **c*Z + a*X +b >0,** for all valid entries in the array. Any non-true inequality will signal the crossing of a boundary.

APPENDIX D

"CALC"*/ROBO PROGRAMMING CONSTRUCTS (32 TOTAL)

FROM "WRITE YOUR OWN PROGRAMMING LANGUAGE IN C++", BY NORMAN E. SMITH

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED

APPENDIX D. "CALC"/ROBO PROGRAMMING CONSTRUCTS (32 TOTAL)

For additional information on the "CALC" programming constructs refer to "Write Your Own Programming Language in C++" by Norman E. Smith.

+ (n1 n2 --- n1+n2)
"plus". Add n1 to n2. The sum is placed on the top of the stack.
1 2 +
The result, 3, is left on the top of the stack.

- (n1 n2 --- n1-n2)
"minus". Subtract n2 from n1 and place the result on the top of the stack.
2 1 -
The result, 1, is left on the stack.

* (n1 n2 --- n1*n2)
"star". Multiply n1 times n2.
2 2 *
The result, 4, is placed on the top of the stack.

/ (n1 n2 --- n1/n2)
"slash". Divide n1 by n2.
4 2 /
Leaves 2 on the top of the stack.

mod (n1 n2 --- remainder [n1/n2])
"mod". Calculate the remainder of n1 divided by n2:
4 2 mod
Leaves 0 on the stack.

dup (n1 --- n1 n1)
"dup". Duplicate the top stack entry:
1 dup
Results in two 1s on the stack.

swap (n1 n2 --- n2 n1)
"swap". Swap the top two entries on the top of the stack.
1 2 swap
Top of the stack is 1.

drop (n1 ---)
"drop". Remove the top number on the stack.

variable (---) Compile time
(--- addr.of.var) Run time
"variable". At compile time, creates a variable:
variable trash
At run time, push the address of the
variable onto the stack:
trash (--- addr.of.trash)

? (address ---)
"query". Type the contents of the address that is on the
top of the stack.
trash ?
types the value stored in the variable trash. This is
logically equivalent to "<- =".

@ (address --- value)
"fetch". Fetch the value from address and place on the top
of the stack.
trash @
retrieves the value stored at trash

! (value address ---)
"store". Store the value into address.
5 trash !
moves 5 into the variable trash.

< n1 n2 --- truth)
"less than". Compare the two numbers on the top of the stack and
return TRUE or FALSE.
5 6 < (returns true)
5 5 < (returns false)
6 5 < (returns false)

= (n1 n2 --- truth)
"equals".
5 6 = (returns false)
5 5 = (returns true)
6 5 = (returns false)

> (n1 n2 --- truth)
"greater than".

```
5 6 >      (returns false)
5 5 >      (returns false)
6 5 >      (returns true)
```

if else then (truth ---)
"if".

```
5 5 =
if ." True comparison"
else ." False comparison"
then
```

else (---)
"else". See 'if'.

then (---)
"then". See 'if'.

. (value ---)
"dot". Display the number at the top of stack.

words (---)
"words". Display all of the currently defined macros

.s (---)
"dot-s". Non-destructive Data Stack display

((---)
"left paren". Text between the braces is treated as a comment. Left brace starts a comment.

```
( This is a comment )
```

See ')'.

) (---)
"right paren". Close a comment. See '('.

: (---)
"colon". Start a new macro definition. The macro name is the next word in the input stream:

```
: new macro ;
```

See ';'.

; (---)
"semi-colon". Finishes a macro definition and exits compile mode. Must be paired with :. See ':'.

." text" (---)
"dot double quote". Type a string. Use only in a macro. Used in the form:

    ." This is a string."

cr (---)
"c-r". Type a carriage return.

0do n ---)
"zero-do". Begin a do loop. Can be used only in a macro definition.

    : ten-dots 10 0do " ." loop ;

See 'loop'.

loop (---)
"loop". The end of a do loop. Can be used only inside a macro definition. See 'do'.

i (--- n)
"i". Push the current do loop index onto the Data Stack.

begin (---)
"begin". Begin a do until loop. Can be used only in a macro definition.

    : wait_a_minute begin ." waiting" seconds @ 60 > until ;

See 'loop'.

until (truth ---)
"until". The end of a do until loop. Can be used only inside a macro definition.
See 'begin'.

APPENDIX E.

ROBO ADVANCED/DIAGNOSTIC COMMANDS (48 TOTAL)

APPENDIX E. ROBO ADVANCED/DIAGNOSTIC COMMANDS (48 TOTAL)

Note: Commands are listed in chronological rather than alphabetical order

rnd (n1 --- n2)
"random". Return a random number between 0 and n1. The result n2 is stored on top of the data stack.

scramble (---)
"scramble". Seeds the random number generator with the relative timer value.

RB (n1 n2 --- n3 n4)
"Range and Bearing". Takes two numbers n1, n2 representing the coordinates of an X(meters North)/Z(meters East) point and returns range(meters) and bearing(degrees) from current position.

turning (--- addr)
Pushes the address of the pivot turn flag onto the stack.

done (--- addr)
Pushes the address of the waypoint flag onto the stack.

collision (--- addr)
Pushes the address of the obstacle flag onto the stack.

bounds (--- addr)
Pushes the address of the boundary flag onto the stack.

clear_B (---)
Clears all line boundary entries.

seconds (--- addr)
Pushes the address of the relative timer onto the stack.

CRC (--- addr)
Pushes the address of the CRC flag onto the stack.

MI (--- addr)
Pushes the address of the 3rd serial port flag onto the stack.

Echo (--- addr)
Pushes the address of the echo flag onto the stack.

Bumper (---addr)
Pushes the address of the bumper status flag onto the stack.

Fe (--- addr)
Pushes the address of the ferromagnetic bit on the stack.

Al (--- addr)
Pushes the address of the bumper metal detect bit on the stack.

Leader (--- addr)
Pushes the address of group leader status onto the stack.

flip (--- addr)
Pushes the address of the pose flag onto the stack.

ac (--- addr)
Pushes the address of the analog compass value onto the stack.

object (--- addr)
Pushes the address of the encounter type onto the stack.

ra (--- addr)
Pushes the address of specular reflection angle onto the stack. (Computed at the crossing of a line boundary).

rd (--- addr)
Pushes the address of remaining distance to waypoint onto the stack. (Computed at the crossing of a line boundary).

na (--- addr)
Pushes the address of the heading perpendicular to intended track onto the stack. (when track following).

nd (--- addr)
Pushes the address of the distance to intended track onto the stack. (relevant when track following).

ver (---)
Displays control system software version data and time.

STOP (---)
Stops vehicle and aborts mission.

RESET (---)
Causes a cold reboot of the system (uses HW watchdog).

PIDRAIL (n1 ---)
Limits the range of PID speed corrections from -n1 to +n1.

Kc (--- addr)
Pushes the address of the direction PID proportional term gain onto the stack.

Ki (--- addr)
Pushes the address of the direction PID integral term gain onto the data stack.

Kd (--- addr)
Pushes the address of the direction PID differential term gain onto the stack. Used when moving forward.

Kdt (--- addr)
Pushes the address of the direction PID differential term gain onto the stack. Used only during pivot turns.

auto (--- addr)
Pushes the address of the navigation aid onto the stack.

process Used to recursively call the main processing loop.

ATTN (---)
Causes an "attention" character (0x01) to be issued.

debug_on (---)
Enables diagnostic messages and parameter stack messages.

debug_off (---)
Disables diagnostic messages and parameter stack messages. The only unsolicited messages are:

OK (system prompt)
Error (for invalid commands)
Underflow (not enough parameters on the stack)
Overflow (parameter stack overflow)

Completed (turn)
Reached (waypoint)
Creeping (if nominal speed is set to zero)
Encounter (Bumper/Detection events)
Rollover (inclinometer reading at limit)
Detection (object detected by sensors)
Flip-up/Flip-down (vehicle upside down)

id (--- id)
Pushes the vehicle currently assigned id number onto the stack.

RENAME (id ---)
Assigns a new vehicle id number.

Sonar (--- addr)
Pushes the address of the collision range onto the stack.

Bias (--- addr)
Pushes the address of the compass global offset onto the stack.

AUTO_BIAS (--- addr)
Pushes the address of bias calibration status onto the stack.

KVH (--- addr)
Pushes the address of the analog compass type onto the stack.

AD (channel_number --- value)
Returns the raw 12 bit AD reading from the specified channel.

DIN (bit_number --- value)
Returns the value the specified digital input (PIO 0-31) bit.

WZONE (--- addr)
Pushes the address of waypoint non-hunting zone onto the stack.

MPS (--- addr)
Pushes the address of 1 meter per second speed value the stack.

Q6 (---)
Report vehicle latitude/longitude position and heading.
The reply is: Q6 Vehicle_ID Latitude Longitude Heading
Format is: Q6 ID +/-DDMM.MMMMM +/-DDDMM.MMMMM DDD<CR>
Example: Q6 7 3008.43700 –8545.52900 273<CR>

APPENDIX F

Q2* DIAGNOSTIC REPORTS

** "Q2" refers to a Query form format*

TABLE F-1. Q2* DIAGNOSTIC REPORT

Description	Units	Data Type Sent	Worst case data (length)	Worst case length
Message Type	N/A	ASCII	$Q2	3
Machine ID	N/A	ASCII	m 1	3
North	meters*10	ASCII	-1234	5
East	meters*10	ASCII	-1234	5
Down	meters*10	ASCII	-1234	5
Compass Heading	degrees*10	ASCII	3599	4
Roll Inclinometer	degrees*10	ASCII	499	3
Pitch Inclinometer	degrees*10	ASCII	499	3
Left Motor Current	amps*100	ASCII	1234	4
Right Motor Current	amps*100	ASCII	1234	4
Static Battery Voltage	volts*100	ASCII	1234	4
Moving Battery Voltage	volts*100	ASCII	1234	4
Direction Byte	Binary	ASCII	FF	2
Health	Binary	ASCII	FFFF	4
Status Byte	Binary	ASCII	FF	2
Object Type Byte	Binary	ASCII	FF	2
Checksum	Binary	ASCII	*FF	3
Delimiters	SPACE			15
Terminator	<CR>			1
TOTAL				76
Notes:				
1) Binary Data: Binary values are converted to their ASCII equivalent, i.e. binary 11111111 = ASCII "FF"				
2) All floats and doubles are converted to ints/longs by multiplying by the value indicated under units				
3) Worst Case Data does not contain valid data but instead reflects the worst case length of the field				
Binary Data Breakdown				
Field	bit assignment	Description		
Direction	0	Right side forward	*currently Direction*10+flip	
Direction	1	Right side reverse		
Direction	2	Left side forward		
Direction	3	Left side reverse		
Direction	4-6	Future use		
Direction	7	Flip switch status	0=top plate up, 1=flipped	
Health	0	PE Comp Status	0=okay, 1=failure	*currently SP_flood
Health	1	Compass Status	0=okay, 1=failure	
Health	2	Gyro Status	0=okay, 1=failure	
Health	3	Encoder Status	0=okay, 1=failure	
Health	4	PE LBL status	0=okay, 1=failure	
Health	5	Bumper Status	0=okay, 1=failure	
Health	6	Bumper M/D Status	0=okay, 1=failure	

** "Q2" refers to a Query form format*

TABLE F-1. Q2 DIAGNOSTIC REPORT, CONTINUED

Description	Units	Data Type Sent	Worst case data (length	Worst case length
Health	7	Tracer Status	0=okay, 1=failure	
Health	8	PIC status	0=okay, 1=failure	
Health	9-15	Future use		
Status	0	Collision Flag	0=collision flag not set, 1=in collision mode	
Status	1	Barrier Flag	0=not in barrier mode, 1=in barrier mode	
Status	2	Bounds Flag	0=inside bounds, 1=outside bounds	
Status	3-7	Future Use		*currently SP_temp
Object Type	0	Proud Indicator	0=not proud, 1=proud	
Object Type	1	Aluminum Detected		
Object Type	2	Steel Detected		
Object Type	3-7	Future Use		
Notes: Direction byte conversions (just looking at bits 0-3):				
	0000	stop		
	0101	forward		
	1010	reverse		
	1001	left		
	0110	right		
Sample Report				
$Q2 m machine_num north east down heading roll pitch left_current right_current				
*Vstatic Vdynamic direction health status object_type*hh*				
Example				
*$Q2 m 2 1234 4321 120 355 -12 30 553 585 2378 2032 5 0 0 0*hh*				
Receiving Q2 report from machine 2 with the following update information				
north position	*123.4 meters*			
east position	*432.1 meters*			
down position	*12.0 meters*			
heading	*35.5 degrees*			
roll inclinometer	*-1.2 degrees*			
pitch inclinometer	*3.0 degrees*			
left motor current	*5.53 amps*			
right motor current	*5.85 amps*			
static battery voltage	*23.78 volts*			
moving battery voltage	*20.32 volts*			
direction	*we are moving forward and the vehicle has not flipped*			
health	*all systems/components are working properly*			
status	*no flags set*			
object type	*no objects detected*			
checksum	*hh -- exclusive or of bytes between $ and *, but not including $ or **			

APPENDIX G

EEL ("RYOT" VERSION) PARTS LIST

DISCLAIMER
The appearance of trade names in this document does not constitute endorsement by the Department of Defense, the Navy, or the Dahlgren Division Naval Surface Warfare Center – Panama City (NSWC - PC)

DISCLAIMER

The appearance of trade names in this document does not constitute endorsement by the Department of Defense, the Navy, or the Dahlgren Division Naval Surface Warfare Center – Panama City (NSWC - PC)

TABLE G-1. EEL (RYOT VERSION) PARTS LIST

Description	Manufacturer, Supplier	Model	Function	Qty	Unit Price
Motor/Transmission Assembly	Fisher Price	Power Wheels	Locomotion	4	$ 30
Wheel Assemblies	Fisher Price	Power Wheels	Locomotion	4	$ 20
Enclosure	BUD	10"x17"	Chassis	1	$ 30
Assorted Hardware	ACE	Threaded Rod 7/16"	Axles	2	$ 15
Battery	Panasonic	7.2 Ah	Power Source	2 - 4	$ 20
RC Reversible Speed Controllers	Tekin	Titan	DC Motor Speed Control	2 - 4	$ 100
Microcontroller	Tern	A104s	Guidance and Control	1	$ 400
Flux Gate Compass	Precision Navigation	TCM2	Direction and Orientation	1	$ 750
WAAS Enabled GPS	Garmin	GPS16-HVS	Global Positioning	1	$ 250
Connectors , Fuses, Wire, Velcro Tape	Radioshack	Assorted	Electronics Assembly	1	$ 100
Ultrasonic Transducer(s)	Robostore	Servo	Obstacle Detection	1	$ 200
Digital Radio Link 900 MHz (ISM)	World Wireless Communication	Microhopper	Communication	2	$ 300
Digital Radio Link 900 MHz (ISM)	Freewave	DGR-115R	Communication	2	$1300
Wireless Video Link 2.4 GHz	X10	Xcam2 and Receiver	Live Remote Video <30m	1	$ 200
Camera, Antennas, L band RF Video	Semco	Selectable Frequency	Extended Distance Video	1	$4500

DISTRIBUTION

	COPIES
DOD ACTIVITIES (CONUS)	
ATTN: MATT HOLM DEFENSE THREAT REDUCTION AGENCY/TDSF 8725 JOHN J. KINGMAN RD MSC 6201 FT BELVOIR, VA 22060-6201	1
ATTN: CODE 32OE (TOM SWEAN) OFFICE OF NAVAL RESEARCH 800 NORTH QUINCY STREET ARLINGTON VA 22217	1
DOD ACTIVITIES (NON-CONUS)	
DEFENSE TECHNICAL INFORMATION CENTER 8725 JOHN J KINGMAN RD SUITE 0944 FORT BELVOIR VA 22060-6218	2

		COPIES
INTERNAL		
XPAOY	(TECHNICAL LIBRARY)	10
XPAOV	(COLE)	2
R11	(BERNSTEIN, P)	2
	(KUCIK)	10
	CDs	20
	(COFER)	2
	(THREAT)	2
R12	(TUOVILLA)	2
R23	(GAVRILASH)	20
	Credit card size CDs	30
	(BERNSTEIN, C)	2
A62	(WILBANKS)	2
A92	(HUTCHENSON)	2
E	(DORAY)	2
E31	(BLANKENSHIP)	2
E51	(RECK)	2

NAVSEA

NAVAL SURFACE WARFARE CENTER

Panama City