



AFRL-RX-TY-TR-2011-0049

## DEVELOPMENT OF INTELLIGENT UNMANNED SYSTEMS

---

Carl D. Crane and David G. Armstrong

Center for Intelligent Machines and Robotics  
University of Florida  
P.O. Box 116300  
Gainesville, FA 32611

Contract No. FA4819-05-D-0011-0001

May 2011

**DISTRIBUTION STATEMENT A:** Approved for public release; distribution unlimited.  
325FW-1106, 4 October 2011.

**AIR FORCE RESEARCH LABORATORY  
MATERIALS AND MANUFACTURING DIRECTORATE**

## **DISCLAIMER**

**Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not constitute or imply its endorsement, recommendation, or approval by the United States Air Force. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Air Force.**

**This report was prepared as an account of work sponsored by the United States Air Force. Neither the United States Air Force, nor any of its employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.**

## NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the 325th Fighter Wing Public Affairs Office at Tyndall Air Force Base, Florida and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (<http://www.dtic.mil>).

AFRL-RX-TY-TR-2011-0049 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

WALTZ.WALTER.M.1006453097  
M.1006453097

Digitally signed by WALTZ.WALTER.M.1006453097  
DN: c=US, o=U.S. Government, ou=DoD, ou=PKI,  
ou=USAF, cn=WALTZ.WALTER.M.1006453097  
Date: 2011.06.30 10:21:30 -0500

---

WALTER M. WALTZ  
Work Unit Manager

RICHLIN.DEBRA.L.1034494149  
.1034494149

Digitally signed by RICHLIN.DEBRA.L.1034494149  
DN: c=US, o=U.S. Government, ou=DoD, ou=PKI,  
ou=USAF, cn=RICHLIN.DEBRA.L.1034494149  
Date: 2011.07.11 15:53:29 -0500

---

DEBRA L. RICHLIN, DR-III  
Acting Chief, Airbase Engineering  
Development Branch

RHODES.ALBERT.N.III.1175488622  
.N.III.1175488622

Digitally signed by RHODES.ALBERT.N.III.1175488622  
DN: c=US, o=U.S. Government, ou=DoD, ou=PKI,  
ou=USAF, cn=RHODES.ALBERT.N.III.1175488622  
Date: 2011.09.07 15:25:32 -0500

---

ALBERT N. RHODES, PhD  
Chief, Airbase Technologies Division

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

**REPORT DOCUMENTATION PAGE**

*Form Approved  
OMB No. 0704-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

**PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

<b>1. REPORT DATE (DD-MM-YYYY)</b> 01-MAY-2011		<b>2. REPORT TYPE</b> Final Technical Report		<b>3. DATES COVERED (From - To)</b> 16-MAR-2005 -- 19-MAR-2011	
<b>4. TITLE AND SUBTITLE</b> Development of Intelligent Unmanned Systems				<b>5a. CONTRACT NUMBER</b> FA4819-05-D-0011-0001	
				<b>5b. GRANT NUMBER</b>	
				<b>5c. PROGRAM ELEMENT NUMBER</b> 0909999F	
<b>6. AUTHOR(S)</b> Crane, Carl D.; Armstrong, David G.				<b>5d. PROJECT NUMBER</b> GOVT	
				<b>5e. TASK NUMBER</b> F0	
				<b>5f. WORK UNIT NUMBER</b> QF503017	
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> Center for Intelligent Machines and Robotics University of Florida P.O. Box 116300 Gainesville, FL 32611				<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>	
<b>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> Air Force Research Laboratory Materials and Manufacturing Directorate Airbase Technologies Division 139 Barnes Drive, Suite 2 Tyndall Air Force Base, FL 32403-5323				<b>10. SPONSOR/MONITOR'S ACRONYM(S)</b> AFRL/RXQES	
				<b>11. SPONSOR/MONITOR'S REPORT NUMBER(S)</b> AFRL-RX-TY-TR-2011-0049	
<b>12. DISTRIBUTION/AVAILABILITY STATEMENT</b> Distribution Statement A: Approved for public release; distribution unlimited.					
<b>13. SUPPLEMENTARY NOTES</b> Ref Public Affairs Case # 88ABW-1106, 4 October 2011. Document contains color images.					
<b>14. ABSTRACT</b> This final report describes the technologies designed and developed by the University of Florida (UF) in support of the Air Force Research Laboratory (AFRL). An off road vehicle (Fig. 1) and a hybrid Toyota Highlander (Fig. 2) have been automated and instrumented with pose estimation (GPS and inertial) and object detection (ranging (LADAR), vision and light detection) sensors. The control architecture consists of four primary elements, i.e. Planning Element, Perception Element, Intelligence Element, and Control Element. The architecture is implemented on a system distributed over ten single-board computers that intercommunicate via the Joint Architecture for Unmanned Systems (JAUS) version 3.2 protocol.					
<b>15. SUBJECT TERMS</b> Robotics, Unmanned Ground Vehicle, JAUS (Joint Architecture for Unmanned Systems), Artificial Intelligence, SLAM (Simultaneous Localization and Mapping)					
<b>16. SECURITY CLASSIFICATION OF:</b>			<b>17. LIMITATION OF ABSTRACT</b> UU	<b>18. NUMBER OF PAGES</b> 34	<b>19a. NAME OF RESPONSIBLE PERSON</b> Walter M. Waltz
<b>a. REPORT</b> U	<b>b. ABSTRACT</b> U	<b>c. THIS PAGE</b> U			<b>19b. TELEPHONE NUMBER (Include area code)</b>

Reset

**TABLE OF CONTENTS**

LIST OF FIGURES ..... ii

1. SUMMARY ..... 1

2. INTRODUCTION ..... 2

2.1. Purpose..... 2

2.2. Background ..... 2

2.3. Scope..... 2

2.4. Goals & Objectives ..... 3

3. METHODS, ASSUMPTIONS AND PROCEDURES ..... 4

3.1. Adaptive Planning Framework (APF)<sup>[1]</sup> ..... 4

3.2. Cognitive Resource Management Framework (CRMF)<sup>[2]</sup> ..... 5

3.3. Terrain Smart Sensor (TSS)<sup>[3]</sup> ..... 6

3.4. World Model Knowledge Store (WMKS)<sup>[4]</sup> ..... 8

3.5. Simultaneous Localization, Mapping (SLAM) and Object Tracking<sup>[5]</sup> ..... 10

3.6. Vision Based Lane Finder and Path Finder<sup>[6]</sup> ..... 11

3.7. Reactive Driver (RD)<sup>[7, 8]</sup> ..... 11

4. RESULTS AND DISCUSSION ..... 14

4.1. APF<sup>[1]</sup> ..... 14

4.2. CRMF<sup>[2]</sup> ..... 14

4.3. TSS<sup>[3]</sup> ..... 15

4.4. WMKS<sup>[4]</sup> ..... 16

4.5. SLAM and Object Tracking<sup>[5]</sup> ..... 17

4.6. Vision Based Lane Finder and Path Finder<sup>[6]</sup> ..... 18

4.7. RD<sup>[7, 8]</sup> ..... 18

5. CONCLUSIONS..... 20

5.1. APF<sup>[1]</sup> ..... 20

5.2. CRMF<sup>[2]</sup> ..... 20

5.3. TSS<sup>[3]</sup> ..... 21

5.4. WMKS<sup>[4]</sup> ..... 21

5.5. SLAM and Object Tracking<sup>[5]</sup> ..... 22

5.6. Vision Based Lane Finder and Path Finder<sup>[6]</sup> ..... 22

5.7. RD<sup>[8]</sup> ..... 22

6. RECOMMENDATIONS ..... 24

7. REFERENCES ..... 27

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS ..... 28

**LIST OF FIGURES**

<b>Figure</b>	<b>Page</b>
1. Off-Road Autonomous Vehicle .....	1
2. Toyota Highlander Autonomous Vehicle .....	1
3. Architecture and Control Diagram.....	3
4. Grid Map.....	7
5. Sample Output from a Road with Barrel Obstacles.....	16
6. Overlay of Data Collected with the Passenger Side LADAR (red) and Driver Side LADAR (blue) .....	17
7. Lane Finder with the Detected Lane Lines Drawn over the Image .....	18
8. Lane Finder Operating at Night .....	18

## 1. SUMMARY

This final report describes the technologies designed and developed by the University of Florida (UF) in support of the Air Force Research Laboratory (AFRL). An off road vehicle (Fig. 1) and a hybrid Toyota Highlander (Fig. 2) have been automated and instrumented with pose estimation (GPS and inertial) and object detection (ranging (LADAR), vision and light detection) sensors. The control architecture consists of four primary elements, i.e. Planning Element, Perception Element, Intelligence Element, and Control Element. The architecture is implemented on a system distributed over ten single-board computers that communicate via the Joint Architecture for Unmanned Systems (JAUS) version 3.2 protocol.

The primary contributions of this work are in the following areas:

1. Adaptive Planning Framework (APF)
2. Cognitive Resource Management Framework (CRMF)
3. Terrain Smart Sensor (TSS)
4. World Model Knowledge Store (WMKS)
5. Vision based lane finder and path finder
6. Simultaneous Localization, And Mapping (SLAM) and object tracking
7. Local Reactive Driver (RD)

The methods, results, and conclusions for each technology listed above are discussed in this report. For more details, PhD dissertations for each topic have been published and are referenced.



**Figure 1. Off-Road Autonomous Vehicle**



**Figure 2. Toyota Highlander Autonomous Vehicle**

## 2. INTRODUCTION

### 2.1. Purpose

The purpose of this work is to develop advanced robotic technologies to support unmanned ground vehicles in the following specific areas: APF; CRMF; TSS; WMKS; Vision based lane finder and path finder; SLAM and object tracking; and Local RD.

### 2.2. Background

The system architecture is a natural extension of the JAUS Reference Architecture, Version 3.2, which defines a set of reusable components and their interfaces. The system architecture is formulated using existing JAUS-specified components wherever possible along with a JAUS-compliant inter-component messaging infrastructure. Tasks for which there are no components specified in JAUS required the creation of “Experimental” components using “User-defined” messages. The actual core software to support the JAUS messaging system was developed and tested extensively.

At the highest level, the architecture consisted of four basic elements, which are depicted in Figure 3. The Planning Element contains the components that act as a repository for a priori data such as the road network. This element will also perform the high level route planning and re-planning based on that data plus real-time information provided by the rest of the system. The Control Element contains the Primitive Driver that performs closed-loop control on vehicle actuators to keep the vehicle on a specified path. The Perception Element contains the components that perform the sensing tasks required to determine the vehicle’s position:, to find a dirt road, to find the lanes on a paved road, to locate both static and dynamic obstacles and to evaluate the smoothness of terrain. Finally, the Intelligence Element contains the components that work together to determine the best course of action to navigate the vehicle in a complex environment based on the current mission and situation.

### 2.3. Scope

The scope of the work completed under this contract can perhaps best be described with an overview of how the system actually works. First, the road network and goal point are used by the High Level Planning system to plan an initial route and pass it to the Sub-System Commander via a mission spooler. Meanwhile, the Perception Element and its various Smart Sensors begin their search for roads, lanes, paths, static and dynamic obstacles and then feed their findings in the form of traversability grids to the various Smart Arbiters. The Smart Arbiters fuse their assigned Smart Sensor grid inputs into a single traversability grid for each Smart Arbiter. Since the output of any Smart Sensor or Smart Arbiter is a traversability grid, the design allows for rolling up Smart Sensors or Smart Arbiters in any combination that makes sense to support any new behavior simply by establishing the appropriate JAUS service connections. The Smart Arbiters have been tailored specifically for the various behaviors that they directly support. In addition to the grid outputs, the Perception Element also contains an assortment of embedded Situation Assessment Specialists that continuously provide metadata findings on the conditions, states, and events related to their specific sensor. Other Situation Assessment

Specialists roll up the perception metadata along with all the other findings provided by the other Specialists as required.

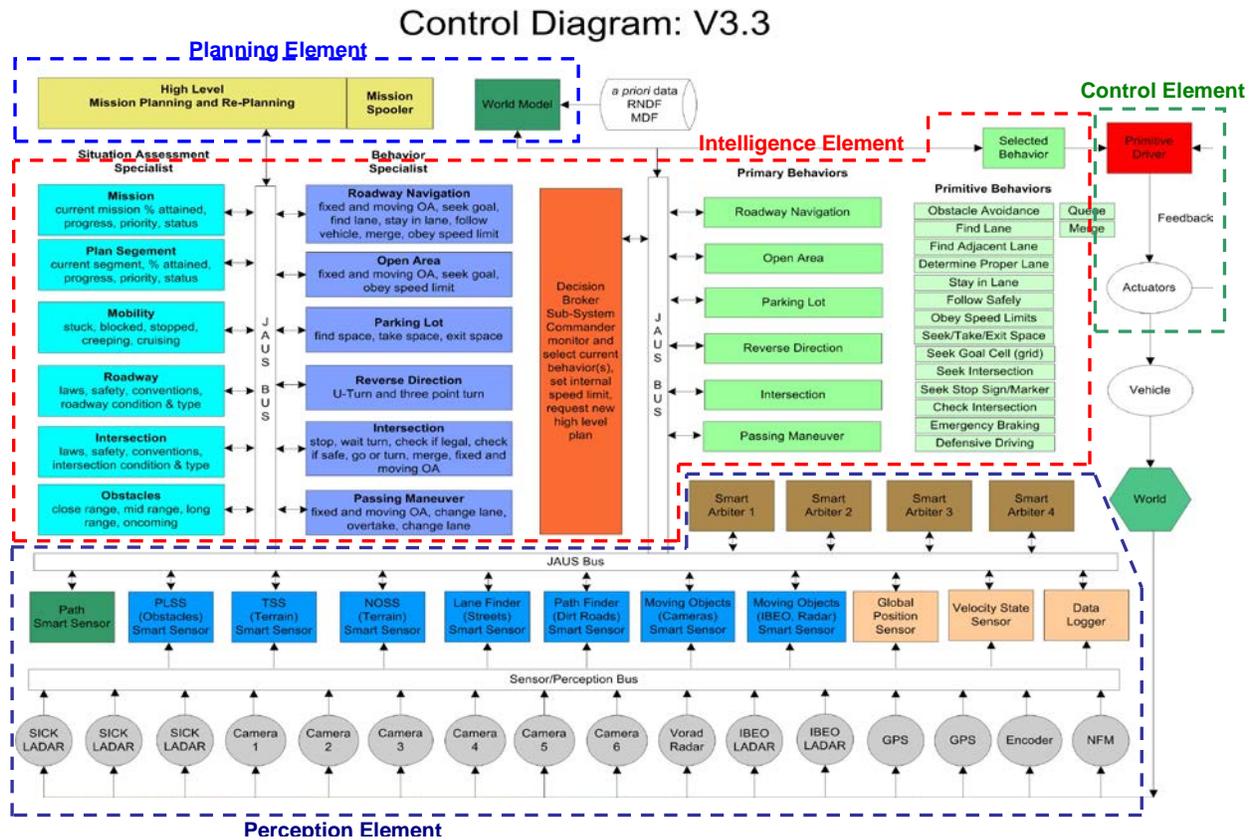


Figure 3. Architecture and Control Diagram

The architecture contains an assortment of both Behaviors and Behavior Specialists. There is a one to one correspondence between the two. The Behaviors contain the actual methods for solving specific driving problems, whereas the Behavior Specialist renders its findings to the Decision Broker about the performance and situational suitability of the behavior that it monitors.

Finally, the Decision Broker is the sole decision maker as to what behavior will control the vehicle. The Decision Broker continuously evaluates the suitability of all of the available behaviors and monitors the Situation Assessment findings required to render the best decision. The appropriate behavior is selected and that behavior determines the control commands to send to the Control Element. The vehicle moves in the real world and the APF process continues.

## 2.4. Goals & Objectives

The objective of this work was to develop and improve capabilities as well as advance the state of the art in unmanned ground vehicle technologies for various uses including base perimeter security operations.

### 3. METHODS, ASSUMPTIONS AND PROCEDURES

#### 3.1. Adaptive Planning Framework (APF)<sup>[1]</sup>

One of the most daunting issues facing autonomous vehicle researchers is how to best exploit sensor and other information discovered during the execution of a plan. If the system takes too long to deliberate on the possible meanings and implications of this newfound data and knowledge, the vehicle may well have progressed beyond the point where it can benefit from the data. Indeed, it may now be sitting atop the unforeseen obstacle that spawned the influx of new information.

An APF for incorporating and managing a collection of virtual Situation Assessment Specialists, Behavior Specialists, and a Decision Broker to support an autonomous ground vehicle (AGV) was devised to address these competing needs. The goal for the Situation Assessment Specialists is to continually render/update their findings regarding a predetermined set of Conditions, States, Events, and Recommendations that are of importance to the vehicle's Behavior and Decision Specialists, and to manage the execution and modification of the vehicle's high-level behavior.

To facilitate implementation of this framework on a fielded AGV, a Knowledge Representation Scheme has been devised that models a team of cooperating "Specialists" divided into three sub-domains:

- Situation Assessment Specialists—each devoted to rendering their findings regarding a set of Conditions, States, and Events that are likely to be of importance to other Specialists.
- Behavior Specialists—each devoted to rendering their Recommendations on the suitability of their associated Behavior Module for controlling the autonomous vehicle, as well as reporting on what behaviors, plans, and sub-goals and other capabilities their Behavior Module might possess.
- Decision Specialist—a collection of one or more Decision Brokers charged with considering the recommendations and findings from the Situation Assessment and Behavior Specialists and making the final determination of how to proceed.

The framework also establishes a reasoning mechanism and control strategy for propagating facts into findings into recommendations and then into executed actions. This strategy must address conflict resolution, truth maintenance, and response to missing information and must support asynchronous operation of the entities. The framework may use either a centralized repository (e.g., a blackboard or knowledge store) as the source and sink of all information produced/consumed by the Specialists, or a decentralized messaging scheme (e.g., a publish/subscribe model) where each Specialist maintains its own copy of what is relevant. Further, the framework places no constraints on the method to be used by a given Specialist, thus supporting a hybrid architecture of various artificial intelligence (AI) and conventional techniques (i.e., a given Specialist could be an Expert System, a Neural Network, a Bayesian Network, a linear program, or a purely algorithmic program).

The operational goal of the APF is to use the elements of the Knowledge Representation Scheme derived during the design phase to produce actionable, high-level decisions at run-time. These decisions, in turn, lead to vehicle behaviors that achieve a mission or a set of goals in light of the current situation. This is accomplished by allowing each Specialist to repetitively apply its rules

and algorithms to produce its Findings. The concept of operation at the lowest level requires each Specialist to gather and analyze inputs and produce results as quickly as possible (nominally targeted as 20 Hz). These “local” Findings are immediately made available to the entities that need them, possibly for further refinement or in support of a behavioral decision. Thus, the concept of operation at the vehicle level is that data, information, and earlier Findings are transformed into new Findings, which are in turn used to produce even newer Findings, to enable Behavior Specialists to provide Recommendations, and/or affect decision-making.

The Center for Intelligent Machines and Robotics’ (CIMAR) NAVIGATOR AGV emerged as the primary field-testing platform for the current work. It is a highly mobile all-terrain vehicle capable of autonomously traversing severe off-road terrain and achieving speeds approaching 30 mph on smooth terrain. It provides multiple LADAR sensors, a highly precise localization capability (twin Global Positioning Systems (GPS’s), a Smiths Aerospace Inertial Measurement Unit, and a drive shaft encoder), as well as actuation of its throttle, brake, steering, and gear shift. The NAVIGATOR’s software runs on a bank of eight networked Linux-based computers (including one dedicated for the APF) and follows a JAUS-compliant component architecture that includes components for sensor arbitration, path planning, and motion planning.

### **3.2. Cognitive Resource Management Framework (CRMF)<sup>[2]</sup>**

AGVs are entrusted with performing highly complex tasks that necessitate the concurrent coordination of multiple system resources and increased capabilities management that are needed to meet these requirements in an effective and efficient manner. The CRMF provides a mechanism for the management and utilization of resources to sustain existing capabilities while facilitating the development and integration of new ones. This research addresses sensor resource management onboard AGVs but is directly applicable and extensible to other unmanned systems operating domains.

The framework conceptualization evolved through the careful analysis of resource management approaches across numerous academic disciplines. At its core, it embodies a Plant Engineering motif, evidenced by the Plant Engineer (PE), the centralized resource broker. It was designed with the following in mind: a well informed manager, using analytical information derived through artificial intelligence methods, will make optimal or near optimal decisions through the application of managerial and technical experience and judgment.

To facilitate implementation and integration with existing vehicle systems and standards, the framework defines a PE element along with a distributed collection of cooperative “analysts” which are utilized to construct a representative model of an autonomous vehicle’s resource capabilities:

- PE— a single centralized authority given sole discretion over assigning value to incoming jobs and to allocate and provision the resources necessary to fulfill the job based on all acquired analyst knowledge.
- Diagnostician/Systemizer Analyst (DSA)— charged with creating and maintaining a real-time representation of system state information, including resource capabilities and configuration.

- Communicator Analyst (CA)— plug-in interface acting as a gateway between the CRMF and vehicle control and planning architectures for importing vehicle mission state information.
- Resource Appraiser Analyst (RAA)— oversees performance management through the collection of framework data for real-time or offline processing.
- Resource Analysts (RA)— each devoted to a particular resource, relays abstract resource attribute representation to DSA for discovery and monitoring.
- Application Analysts (AA)— each devoted to a particular software application requiring the utilization of a resource and converts application specific resource need into generalized framework Job Request for submission to the Plant Engineer.

Together, these framework elements provide mechanisms for the discovery, monitoring, and modeling of resources; guidelines for the submission of job requests; and strategies for resource allocation and scheduling. In addition, the framework defines generalized abstract representations for application resource needs, termed Job Requests, and an extensible class structure of Resource Objects.

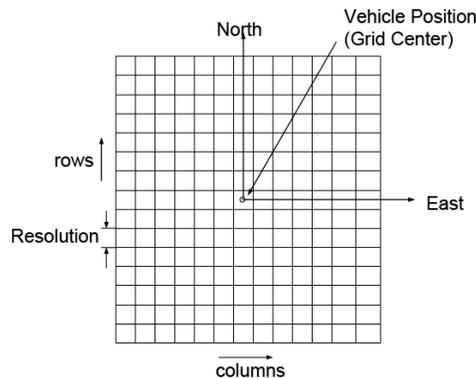
The goal of the reference implementation is to demonstrate how the CRMF enhances the autonomous navigation capabilities set developed for Urban Challenge to simultaneously support detailed environmental mapping and monitoring while exhibiting safe and effective navigation. The operating scenario is as follows: An AGV with a-priori road network knowledge and a predefined patrol region is deployed in an environment with no other information. Utilizing only onboard sensing, perception, and reasoning capabilities the robot will survey the environment and develop a world model representation of all objects, structures, and entities detected that will be maintained within a knowledge store. Upon successful characterization of the operating environment, the platform commences execution of the patrol mission in which it searches the region looking for anomalies. As implemented, the CRMF oversees an array of MATRIX Vision BlueFox cameras mounted on a rotary stepper actuator used to capture image information needed for object classification and characterization.

### **3.3. Terrain Smart Sensor (TSS)<sup>[3]</sup>**

The ability for an autonomous vehicle to operate effectively off-road or in an urban setting relies on the quality of the information it has about its environment. In order for the planning component to determine the optimal path over the ground surface and avoid obstacles, it needs to have an accurate representation of the surroundings. There are two main representations which can be implemented for representing this information. One is a vector model which stores objects as vector coordinates in 2D space. The other method is a grid map of cells with values which determine the traversability of the region within each of those cells. While the vector representation has the benefit of usually being smaller, the grid map representation was selected for this work for several reasons. The regularity of the grid, as illustrated in Figure 4, allows for easy consolidation of traversability estimates from various sensor algorithms or even disparate sensors. The grid map is also easier to update as data is collected over a series of time steps. Finally, the grid map representation allows the planning component to perform planning in raster space, a simpler task than planning using a vector map.

The sensor component outputs a traversability value based on two factors:

1. The severity of a non-traversable region (size of the obstacle or roughness of the terrain) or the good terrain characteristics of a traversable region (smoothness of the terrain).
2. The confidence on the evaluated characteristic. The obstacle might be highly non-traversable, but what is the confidence on the presence of the obstacle? Traversability is output on a scale of 2-7 with higher numbers being more traversable.



**Figure 4. Grid Map**

Obstacle detection (OD) is critically important to the autonomy. Without the ability to accurately and quickly detect obstacles, the vehicle, and other people, buildings, and cars could be damaged. The OD algorithm receives raw range data from the LADAR at the bumper level. The OD algorithm is based on the weighted sum of evidences. The traversability value of a cell is computed based on the weighted sum of the evidence of the cell being occupied or free. The evidence of a cell being an obstacle or free space is derived based on the current sensor observation and initial evidences. A sensor observation may be defined as an outcome of the sensor measurement used to evaluate the state of the system.

The sensor observation is managed internally using the two variables, ‘OccupiedHits’ and ‘FreeHits’ for each cell. After each laser scan the range measurements are transformed into observations. For each single coordinate generated from the range value, the cell to which this coordinate belongs in the Traversability Grid is determined, followed by all of the intervening cells between the determined cell and the sensor. Bresenham’s line algorithm is used to determine the indices of the intervening cells. The ‘OccupiedHits’ buffer is incremented by one for the cell which receives the hit and the ‘FreeHits’ buffer is incremented by one for all the intervening cells. For cases where the received range values are beyond the Traversability Grid map, the cell at the intersection of the line formed by the range value and the sensor origin with the bounds of the grid map is found. For all the cells on this line the ‘FreeHits’ is incremented by one. The OD identifies positive obstacles and renders no opinion regarding the smoothness or traversability of areas where no positive obstacle is reported. Hence it reports traversability values from 2 to 7.

The terrain evaluation (TE) algorithm receives raw range data from the top two LADARS. Each of these two LADARS feed data into an individual terrain evaluation grid, which are then combined. A Cartesian elevation map is built from the successive laser scans and the positioning

system readings corresponding to these scans. From the 3-D map, the terrain is classified based on the geometry of the terrain. A set of classification features is generated by performing statistical analysis on the terrain map. The data points are stored in the corresponding cells of the Traversability Grid as a linked list of 3-D Cartesian coordinates. Each cell in the Traversability Grid is evaluated individually and classified for its traversability value. The following geometrical features are used for the classification:

1. The slope of the best fitting plane through the data points in each cell.
2. The variance of the elevation of the data points within the cell.
3. The weighted neighborhood analysis.
4. Negative obstacle algorithm.

Data fusion is the final step before outputting the grid map data. To take advantage of each sensor algorithm and at the same time overcome some of its limitations, the outputs from the above sensors are fused together using a simple rule-based forward reasoning scheme. The uncertainties associated with the two sensors are combined using certainty factors.

### **3.4. World Model Knowledge Store (WMKS)<sup>[4]</sup>**

The WKMS stores and queries dynamic information using a knowledge store. Objects characterized by autonomous robots are often dynamic in nature. A generic knowledge store was developed for the purpose of providing information about these objects to various components. Each classifying component can specify which dynamic model the knowledge store can use to predict the motion or future location of the object. The WMKS is then able to provide querying components the temporally adjusted object definitions calculated from dynamic models.

A spatial database is a relational or object-oriented database which has been enhanced to support spatial data and perform spatial operators on that data. The JAUS WMKS message set is based upon three primary entities; the object geometry, feature classes and feature class attributes. Previous work primarily focused on the query and storage of static geospatial data object. The message set and underlying systems were extended to allow for the addition of temporal knowledge. Specifically, three new messages were created: Query Vector Knowledge Store Objects Future State, Report Vector Knowledge Store Objects Future State, and Modify Vector Knowledge Store Objects. The first allows a client component to query the knowledge store for the predicted state of an object in the future. The other two messages are used to make changes to objects already stored in the knowledge store, something that previous implementations lacked.

To provide a more robust solution, both static and linear predictors were implemented. While a polynomial predictor was the primary prediction method developed and tested, it was realized that no valid solution would exist for data sets prior to the minimum point count defined in the polynomial predictor. Rather than handle this as a special case in the predictor, a flexible predictor was implemented to facilitate all cases being processed. A Postgresql backend was used for storage of the actual objects. Several parsers were used to discriminate between different feature classes and their specific syntaxes. The methodology used to handle knowledge store requests allowed for multiple configurations of specified data to be as flexible as possible. For example, when an object is being created the knowledge store will query for the next valid object ID to assign to the new object if none was added into the incoming object definition. The

Postgis library was also used for querying and storing spatial data. It provided many geometric abstractions necessary such as overlap and intersects algorithms. It also provided a proven and compact methodology for representing spatial objects in a database.

At this point, work with the WMKS focused around integrating motion prediction of spatiotemporal objects. While this provided a robust, application specific system it was not efficient in the general sense. With this in mind, a WMKS system was created based on draft AS-4 specifications. The new messages and system capabilities allowed for a more generic representation of knowledge that could be used in a variety of domain specific problems. The WMKS still relied on a Postgresql backend with Postgis extensions to serve spatial based queries.

Several notable extensions were made which helped with integration for several projects the WMKS has been used. The first major enhancement was the development of a subscription/server model wherein querying components did not have to pole for information continuously but rather sent just one message to get updates. The standard JAUS event message did not support all of the underlying terminology of the required information in setting up each subscription, so several new messages were formalized including Create Knowledge Store Event Subscription Message and Delete Knowledge Store Event Subscription Message and their corresponding response messages. The subscription approach was instrumental in completion of both the Motion Detection and Perimeter Security task and the Robotic Range Clearance Competition (R2C2) Demonstrator project. Message traffic and response time were significantly improved in both situations along with a decrease in required development time.

Improvements were also made to allow for rapid definition and system integration of new object types and definitions. A robust graphic user interface (GUI) was created to allow even the novice user to easily modify and distribute object definitions across the system which allowed for rapid turnaround for system topology modifications. The WMKS was also rewritten to allow for runtime changes to object definitions through the use of an xml configuration file so that quick changes could be made without holding up development. This also allows for global distribution of system definitions in a common, human readable and editable format.

Another area of improvement was the addition of support for object dependencies within the WMKS. While working towards the Motion Detection and Perimeter Security task, it was found that many of the objects needed to be correlated with other objects that were already stored in the knowledge store or needed to be created with those connections. The idea of dependencies was implemented by using a globally unique identifier inserted into every object definition. This allowed a definition to include one or more pieces of metadata that correlated its value with another object stored in the WMKS. This opened up room for improvements in responsiveness like allowing for querying an object and all of its dependents. Without the idea of dependencies, several query/response messages would have to be negotiated which increases latency and decreases overall system integrity. To facilitate dependent object implementation, several techniques were invented such as allowing a create message to specify all the objects to create and which to implicitly modify including the new objects as dependents.

### **3.5. Simultaneous Localization, Mapping (SLAM) and Object Tracking<sup>[5]</sup>**

It is critical that a robot be able to detect and understand elements in its environment. LADAR sensors have been popular for use in object detection applications such as SLAM and the detection and tracking of moving objects (DATMO) due to their high range accuracy, low cost, and low processing demands. However, these applications have commonly been treated separately despite evidence that they are related. The presence of a moving object adversely affects SLAM systems while static objects are commonly misidentified in DATMO applications. One approach to address these shortcomings has been to combine the two applications in a Simultaneous Localization, Mapping, and Moving Object Tracking (SLAM+DATMO) method.

Past efforts to combine these two tasks have relied on grid-based approaches which require greater memory and processing power due to the use of image processing techniques. In addition, no previous work has attempted to use multiple LADAR to provide a wider field of view. The wider field of view allows the robot to understand more of the world and avoid threats. The work presented here addresses some of the shortcomings described.

A novel SLAM+DATMO approach represents the detected objects using line segments and polygons which are more concise and can be processed more quickly than earlier methods. Also, a formal approach for fusing data from two laser range finders provides a low cost and simple solution for improving sensing capability. Finally, a mechanism for sharing detected object data to other software components is outlined through the use of a centralized WMKS.

The process involves a series of steps. After a new scan from the LADAR is collected, a clustering and feature extraction process is performed. Once the new objects are extracted from the scan, the objects are checked against the list of previously known moving objects to see which new objects can be associated with old moving objects. The new objects which can be matched to old moving objects are used to update their state in the moving object list and also to predict the motion of the objects for the next iteration. The objects which are not matched to previously known moving objects are then tested against the static object list to see if any can be associated with previously known static obstacles. If so, the static obstacle's geometry is updated if necessary. If not, a new object is created and saved in the static object list (The object can later be upgraded to a moving object if motion is detected in the future.).

The software also takes into account other common occurrences such as objects being occluded by moving objects as well as the progressive updating of objects as more and more of their geometry is exposed to the LADARs as a result of a changing vantage point. The use of multiple LADAR sensors for SLAM+DATMO, a previously unutilized feature in other research, is accomplished by creating sufficient error margins to allow for scan discrepancies. This implementation also interfaces with the Knowledge Store, a JAUS component which allows generic objects to be stored, retrieved, and modified in a database. The moving objects detected by this algorithm are constantly saved and modified in the knowledge store as objects with a centroid, outline geometry, and velocity.

### **3.6. Vision Based Lane Finder and Path Finder<sup>[6]</sup>**

A main goal of the vision sensor for an AGV is to provide continuous and precise perception information about traversable paths, future trajectory estimations, and lateral position error corrections within a lane. To accomplish these objectives, multi-camera based Path Finder and Lane Finder Smart Sensors were developed and utilized. These systems create traversable area information for both an unstructured road environment and an urban environment in real time.

The Terrain Estimation Smart Sensor uses classification techniques applied to the image to determine which areas are traversable roads and which are not. The normalized color statistics of the road area directly in front of the vehicle are used to train a classifier and then are applied to the whole image. By using a mixture of Gaussians to define the class which corresponds to the road and the class which corresponds to non-road, a Gaussian decision boundary is generated. This allows the software to make a determination of whether a particular region of the image is part of the road or not. Once this determination is made, the resulting map is smoothed by interpolation and then transformed to the global coordinate system so that the information can be sent to other components as a traversability grid. The resulting estimation of the traversable road area is also converted to vector geometries and saved to the WMKS for long term storage.

The Lane Finder Smart Sensor is designed to visually detect painted road lines in order to provide an estimate of the lane center. It uses two cameras mounted on either side of the vehicle to provide a better vantage point, even if there is another vehicle in the lane ahead. These lane center estimates provide a navigational aid to the planning component which is independent of GPS bias or other GPS inaccuracy. The process of extracting lane lines begins with normalizing the image data to a normalized RGB space. This aids in alleviating the problems caused by shadows and other lighting irregularities. Then a Canny edge filter is applied to the whole image which serves to highlight edges in the image such as between color boundaries. Once the image with highlighted edges is generated, a Hough transform is used to identify dominant lines in the image. This returns several candidates. By applying a filter which isolates the longest and most likely candidates based on a lane line model, the lane lines can be identified in the image. The color of each of the lane lines is also determined as this can be useful information. Finally, the location and orientation of the lane lines are transformed into vehicle coordinates so that the location of the lane center can be determined. This is then used to generate lane center estimates at several distances ahead of the vehicle. These estimates are packed into a JAUS message and sent to the path planning component.

Both components are run at a rate of 20 Hz. One of the assumptions of the Path Finder is that the vehicle is currently on the road since it uses the area directly beneath the vehicle to train its classifier. Similarly the Lane Finder is able to report the lane center with some amount of offset, but if the vehicle deviates completely out of the lane, the Lane Finder will either start to report the lane center of the next closest lane or fail if it is not in a lane at all.

### **3.7. Reactive Driver (RD)<sup>[7, 8]</sup>**

The RD is the component in the CIMAR JAUS architecture that derives the vehicle control inputs from the mission specification. Typically, the mission specification is in the form of a list of waypoints or mission points expressed in latitude and longitude. The mission is expected to be

executed in the listed order, with consideration for obstacles and vehicle limitations. An idealized mission execution would be a straight line path from point to point. The environments that the unmanned ground vehicle (UGV) is tasked in are initially unstructured. Typical UGV's path capabilities are restricted by configuration, precluding sharp turns and lateral motion. These issues make the ideal path difficult or impossible to achieve. Work has been performed to develop a RD that addresses these challenges.

Motion planning and control for autonomous vehicles are complex tasks that must be done online in order for a UGV to operate in a cluttered environment. The first phase of this work addresses the theory, implementation, and test results for some new and novel receding horizon control (RHC) techniques that allow these tasks to be unified into one online approach. The first new method is called heuristic receding horizon control (HRHC), and uses a modified A\* search to fulfill the online optimization required by RHC. The second is called dual-frequency receding horizon control (DFRHC), and is used to simplify the trajectory planning process during the RHC optimization.

The DFRHC is a HRHC implementation applied to a tessellated representation (grid) of the environment immediate to the UGV. Classified obstacles and traversability are represented in the grid representation. The DFRHC utilizes a modified A\* optimization procedure to find the best trajectory through the immediate environment, considering vehicle kinematics, obstacles, and traversability capabilities. Both methods are combined together to form a practical implementation. The AGV named the NAVIGATOR, developed at the CIMAR, serves as a platform for the implementation and testing discussed.

The introduction of moving obstacles into a robot's environment presents added complexity to the motion planning task. This work examines the need for and development of a representation which incorporates the dynamic nature of the environment and presents a novel motion planning method which utilizes this representation to facilitate the generation of optimal trajectories among moving obstacles. This is termed the predictive temporal motion planning (PTMP) method. This new method provides an advanced approach to the problem of generating solution trajectories in dynamic environments by elegantly connecting the tasks of obstacle detection and prediction, environment mapping, and motion planning.

The dynamic environmental representation takes the form of a typical grid which is extended into the time dimension by adding temporal layers to the grid structure. The layers of this temporal grid represent distinct time-steps into the future. These time-steps are determined by considering how the motion planning algorithm calculates its discrete control commands. Obstacle motion prediction is incorporated into the temporal grid by estimating future positions of moving obstacles and displaying these estimates in the layer of the temporal grid associated with the prediction times.

The new motion planning method then can use this predictive temporal grid to investigate potential control input sequences to generate an optimal trajectory to achieve its goal. As the algorithm evaluates potential control commands at various time-steps in the future, it does so by exploring the various temporal layers of the new grid structure corresponding to distinct control times. By considering the estimated future motions of any obstacles, the motion planning

algorithm can more intelligently calculate its control sequences to avoid the objects in an efficient manner.

## 4. RESULTS AND DISCUSSION

### 4.1. APF<sup>[1]</sup>

The NAVIGATOR vehicle built for the Defense Advanced Research Projects Agency (DARPA) Grand Challenge 2005 carried on it an embryonic implementation of the APF, focusing on a pair of Situation Assessment Specialists delivering a handful of Findings to enable the Decision Broker to set the maximum speed of the vehicle.

The architecture for the Urban Challenge version of the NAVIGATOR includes extensive adoption of the APF. The two behaviors chosen for implementation were basic roadway navigation (RN) and an n-point turn (NPT). Three Specialists were identified for the Reference Implementation, the RN Behavior Specialist, the NPT Behavior Specialist, and the Close Range Safety Specialist, each tasked with the monitoring of a behavior and/or responsible for a variety of relevant Findings. Five Decision Broker Protocols were identified for the implementation: one that provides the overarching monitoring of behaviors and invocation of other Protocols and two pairs that transition into and out of the two available behaviors.

Several key concepts were demonstrated and their viability confirmed during the course of testing the APF Reference Implementation for the Urban Challenge version of the NAVIGATOR:

- The notion of a Decision Broker interactively and autonomously orchestrating the behavior of a complex, full-scale AGV.
- The notion of Specialists, implemented as software entities, autonomously determining, using, and exchanging their Findings.
- A hybrid of both deliberative and reactive behaviors cooperating to pursue a mission.
- The use of a granular, distributed knowledge representation scheme.
- The use of a granular, distributed reasoning mechanism operating in near-real-time.

### 4.2. CRMF<sup>[2]</sup>

Prior to performing the field testing, the Reference Implementation underwent a series of stress tests to demonstrate its viability under extreme resource taxing conditions. Approximately 1500 unique Job Requests were created over a two minute time interval. At times, the PE Job Queue reached in excess of 1000 jobs and remained at that level for a prolonged time period. Even under these intense conditions, the PE component update rate did not significantly deviate from its target rate of 40 Hz. The framework continued to function effectively despite the large quantity of jobs to process during a given runtime iteration. These results reinforce those of the performance benchmarking tests previously discussed which indicate stable system performance in the presence of a large backlog of Job Requests.

During field testing of the CRMF Reference Implementation on December 6, 2009, multiple Moving Object (MO) snapshots were used to test the CRMF. Looking at a single MO component snapshot, approximately 30 new objects were detected and added to the WMKS. The test demonstrated that the CRMF provides mechanisms for discovery, modeling, and monitoring of sensing resources while providing a knowledge representation scheme which enables goal abstraction and a uniform approach to job submissions. These are all critical aspects needed to achieve intelligent autonomous real-time resource management.

Several key concepts were demonstrated while proving their viability in field testing the CRMF Reference Implementation:

- Application of Plant Engineering analogy with hybrid scheduling theory paradigms to provide near real-time resource management of a full scale autonomous vehicle platform undergoing reconnaissance surveillance and target accusation (RSTA) activities
- The notion that distributed cooperating analysts, implemented as software entities, distill critical system information into knowledge attributes which result in actionable behavior generation through resource brokering
- Application of 3-dimensional, dynamic, spatiotemporal model of common AGV sensing resource capabilities
- Development of a modular and extensible Resource Object defining Capability and Performance Attributes which embodies the essence of a resource
- Framework promotes technology reuse through the use of a uniform knowledge representation scheme and reasoning mechanism thereby reducing integration costs
- Use of a distributed deliberative reasoning mechanism operating in near real-time
- Framework components and tools aid in design, implementation, and evaluation of complex multi-mission unmanned systems
- Integration with existing AGV's standards and frameworks to provide intelligent resource management

#### 4.3. TSS<sup>[3]</sup>

The OD algorithm can be used only for positive obstacles. It gives no opinion on the smoothness of the terrain and negative obstacles. With hilly roads, the OD generates a lot of ground noise. Most of the noise is due to misclassification of an approaching uphill slope or due to going down a hill transitioning into flat ground. However the OD algorithm is not as complex as the TE. The OD algorithm does not have to create a 3-D point cloud. Figure 5 shows that the OD is very reliable in identifying positive obstacles. The grid is updated about 35 times a second in the region bounded by the field of view of the sensor. In this region moving obstacles which have passed are cleared in the grid, and, if a moving obstacle shows up in the grid, the grid will be updated. Since the OD algorithm does not depend on mapping the true coordinate of the point; but just checks to see if the point belongs to a cell, the error in mapping the obstacle is very small compared to the TE algorithm. The main concern with the TE algorithms is modeling the ground plane. Since the data is collected in successive scans, the ground plane of the vehicle changes. Thus each time the points are registered, they have a different reference plane. In cases where the vehicle is on a flat smooth terrain this is not a problem. However, in cases of uneven terrain, it is very difficult to relate the data to a common ground plane. Although the points are registered in a fixed global frame, there is some error associated with the registration process, and experiments have shown the magnitude of the error depends on the condition of the terrain. Since the look ahead distance of the Terrain LADAR's is limited by the tilt angle of the laser, in the present case the TE algorithm is effective for a range of only 18 m. It does not provide any information on obstacles further than this distance. The TE algorithm maps a moving obstacle as part of the terrain and hence does not clear them after they have passed from the grid. In spite of the above disadvantages of the TE, the algorithm actually maps the surroundings into a 3D point cloud and characterizes the terrain based on slope, variance, and discontinuities. Hence the classification is based on more detailed information of the surroundings as compared to the OD algorithm.



**Figure 5. Sample Output from a Road with Barrel Obstacles**

#### 4.4. WMKS<sup>[4]</sup>

Several general scenarios were tested using the different tracking predictors to give a good sense of real world results. Tests were run while the NAVIGATOR robotic platform was stationary and the target was moving; while the target was stationary and the platform was moving; and while both were moving. A previously developed Terrain Smart Sensor component was used to classify the location of the target relative to the robotic platform. This grid information was then abstracted into data which was stored as an object in the knowledge store. The polynomial predictor was shown to be flexible enough to be used for a large variety of data in the prediction of future state for the tracked objects. It yielded favorable results in tracking not only objects, but also attributes as well. The linear prediction method was compared to the polynomial predictor and was shown to yield less favorable results overall and, in general, provided much less stable behavior when used to predict future states. Each scenario was tested at least five times to decrease the possibility of statistical aberrations in recorded results.

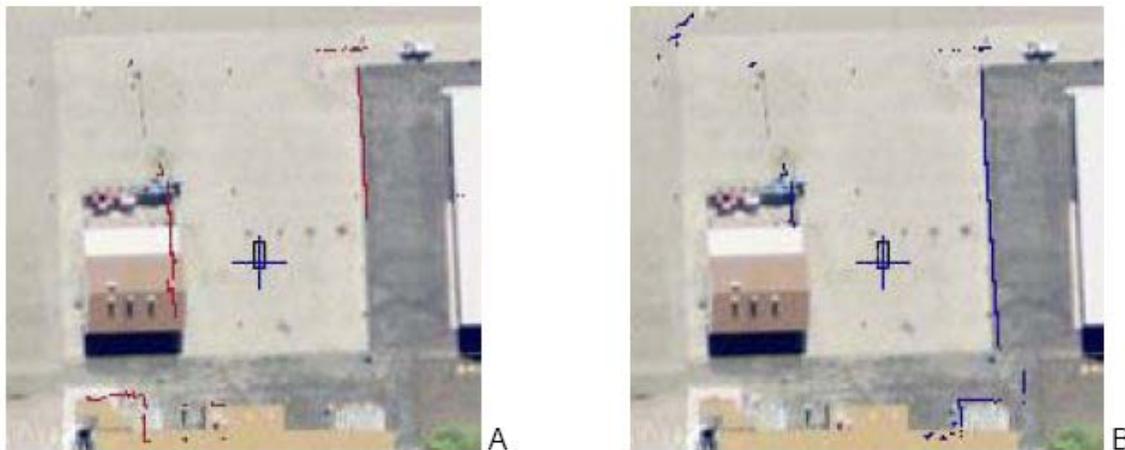
Following the earlier work, the WMKS continued to evolve. The Motion Detection and Perimeter Security task required a responsive yet dynamic centralized knowledge store. The WMKS was used to store objects detected and classified by the LADAR sensors and cameras. Because the knowledge store acted as a central repository of information, the camera component was able to respond to detection events rapidly and store those pictures globally. The information about detected inconsistencies could then be queried and displayed to the operator so that they were able to make the decision whether the detected object was a threat. The knowledge store was extended for this task to efficiently store and serve binary image data. Several transport issues were resolved regarding dropped packets and large, user datagram protocol (UDP) message traffic that at first prevented relay of images through the system.

The R2C2 Demonstrator project showed that the knowledge store was a highly robust, easily modified system. The reuse of the same component while simply changing the configuration file highlighted the utility of having this software. After the overall system was designed, including what objects and properties would be stored along with interdependencies of those objects, the configuration file to the knowledge store was accordingly modified. The short turnaround allowed other software to be developed more quickly. The simplicity of the schema modifications also facilitated further modifications when the scope of the task changed several

times during development. Only one major bug was discovered regarding an incorrect size limit of a data type. This underlines the reliability of using the software over a span of several months.

#### 4.5. SLAM and Object Tracking<sup>[5]</sup>

Testing involved both simulated data and live data taken aboard the vehicle. Static testing with a stationary vehicle and moving objects was performed as well as the vehicle in motion among other moving objects. Figure 6 shows data that was collected from two different LADAR sensors on the vehicle. The average execution time for the code was about 0.19 seconds giving updates at about 5 Hz. One test involved moving the vehicle through a static environment and observing the correctness of the algorithms. It was found that the object tracking and updating algorithms worked well, but there were a number of issues. First, the effect of vehicle pitch and roll greatly affected what objects could be detected by the LADAR. Also, there were inconsistencies introduced into the object representation by the object resolution algorithm. Sometimes a stored point would be kept when it should have been removed from the representation because it caused the object shape to become distorted. Although the updating algorithm was sometimes inconsistent, it also produced some promising results. It can be seen that the algorithm does successfully outline some of the objects in the environment. Some errors can be explained by errors in the global position and orientation sensor (GPOS) position estimate.



**Figure 6. Overlay of Data Collected with the Passenger Side LADAR (red) and Driver Side LADAR (blue)**

During dynamic vehicle testing, the system was also able to simultaneously track the moving object and update the static object representations. The problems of inconsistent static object updating and incorrect classification of static objects as moving objects seen when the platform moved through a static environment were also seen during this stage.

Finally, the position estimation system was tested in the presence of moving objects. To evaluate its effectiveness, the vehicle was fixed in place, and a moving object was allowed to move through the environment. In general, the corrected position seems to behave in a similar manner to the previous tests with a static vehicle. Although the corrections in the Universal Transverse Mercator (UTM) y position and rotation appear to be noisier, it is unclear if this was caused by

the presence of the moving object or not. However, the error of the system is still reasonable. The change in the UTM x and y positions are less than 0.30 m and 0.15 m respectively and the change in the yaw is less than 0.5 degrees.

#### 4.6. Vision Based Lane Finder and Path Finder<sup>[6]</sup>

Both the Lane Finder and the Path Finder have been shown to perform well in urban environments. The Lane Finder is probably the more useful of the two and has been tested in both daytime and nighttime conditions with the head lights on as shown in Figures 7 and 8. It has shown its robustness since it can handle both solid lines and dashed lines. It is also able to handle cases when there is no center line by estimating the lane center from a single line and an assumed standard offset.



Figure 7. Lane Finder with the Detected Lane Lines Drawn over the Image



Figure 8. Lane Finder Operating at Night

Some challenges still include misidentifying the right line if a bike lane is present and handling large deviations from the lane. This component also does not provide any additional navigational help when traversing intersections because of the absence of painted lines.

#### 4.7. RD<sup>[7, 8]</sup>

The DFRHC has been implemented on several UGVs. The purpose built NAVIGATOR, developed by CIMAR for the second DARPA Grand Challenge, utilized a basic DFRHC to perform the low level reactive control of the vehicle while undergoing autonomous operation. This implementation ran on a 2 GHz single core AMD Athlon based computer. The control

frequency was fixed at 20 Hz. The planning frequency was implicitly variable in that the A\* generational step was fixed at a distance of 3 to 5 meters. The velocity of the vehicle determined the instantaneous planning frequency, but since the maximum speed of the NAVIGATOR is less than 35 mph, the planning frequency was always significantly lower than the control frequency.

The A\* was implemented with three children in each generation, one straight ahead, one at maximum left turn, and one at maximum right turn. The trajectory cost was evaluated with a straight line model applied to the optimizing heuristic. Performance was adequate on the hardware deployed up to approximately 25 mph.

The DFRHC was modified for the Urban Navigator, a vehicle based on a 2007 Toyota Hybrid Highlander. This vehicle was developed by CIMAR for use in the DARPA URBAN Challenge and continued work in surveillance and high speed autonomy. A form of feed forward compensation was added to the RHC by planning from a location along the current trajectory of the vehicle. This location was found by estimating the travel of the vehicle for a fixed time from the current location. This allowed compensation for system lag. Another modification of note is the addition of a history based trajectory in the first generation of the A\* search for each iteration. This lowered computational requirements by driving the solution towards the previous one, if it was still valid. The trajectory cost evaluation was augmented to consider the vehicle trajectory, including segmented curve representations. This improved obstacle avoidance in curves. This DFRHC was implemented with a control frequency of 40 Hz on a 2.4 GHz, dual core Athlon processor based computer.

The PTMP approach to consider moving obstacles extended the DFRHC implementation. Testing was performed on the Urban Navigator system during high speed navigation work. The previous modifications were maintained, and the addition of a time varying representation of the traversability grid was added. This allowed planning that considers the future location of classified moving objects. The hardware was consistent, but the software was more aggressively threaded to better utilize resources.

## 5. CONCLUSIONS

### 5.1. APF<sup>[1]</sup>

The APF has been shown to be both a viable method for representing and managing complex, situation-dependent behavior on an AGV and a valuable contribution to researchers tasked with developing and fielding such a vehicle. The value of the architecture can be measured by the major role it contributed to the architecture and design of the AGV fielded by Team Gator Nation for the 2007 DARPA Urban Challenge and the platforms that followed.

To underscore this latter point, it is useful to mention the initial architecture that was presented to DARPA by Team Gator Nation. The growth of behaviors, and the Behavior Specialists needed to assess them, compared to the 2005 version shows the available expansiveness of the architecture. Likewise, there is an extensive proliferation of Situation Assessment Specialists needed to derive the many Findings, which are needed to properly understand and respond to the situation at hand. This architecture (and its use of the APF) continues to evolve as the team migrates towards a detailed design and as the team members become more directly involved in the details of how the framework operates and how it should be used. Having a team of researchers dialoging about communicatory implementation, Findings, Specialists, and Decision Protocols further underscores the viability and usefulness of the framework. This emergent adoption of the ideas and innovations resulting from the subject research has already strengthened and improved the framework. For example, the notion of allowing the duties of the Decision Broker to be distributed into layers of abstraction (i.e., using Decision Protocols within a Behavior to select sub-behaviors) was a direct result of team discussions. In addition, the direction in which this framework's possible development can lead is already being addressed by members of the team.

### 5.2. CRMF<sup>[2]</sup>

The CRMF has been shown to be both a viable method for modeling and managing a distributed collection of heterogeneous resources and resource-requests and a valuable contribution to the researchers whose algorithms depend on the reliable availability of resources to fulfill mission operating requirements. The framework's viability and contributions were demonstrated in both the Reference Implementation and during the CIMAR Environmental Mapping and Change Detection demonstration for AFRL conducted at the Gainesville Raceway test facility. This work will not only continue to benefit researchers at the University of Florida (UF) who will build upon the foundation created by the Reference Implementation, but could impact the robotics community as a whole by gaining acceptance in standards bodies such as the Society of Automotive Engineers (SAE)'s AS-4 committee.

The framework's value is evidenced by the pronounced role it has assumed in the Urban NAVIGATOR software architecture. Researchers at CIMAR will continue utilizing the CRMF while developing advanced technologies that broaden the frontiers of AGV application. Researchers are developing algorithms for simultaneous localization mapping, and object tracking using multiple 2-D laser scanner sensors. Not only can the framework be utilized to model and monitor these resources, it can intelligently select the most appropriate resources to utilize as inputs to the mapping and tracking algorithms should other applications require the intermittent use of a particular scanner. Furthermore, the PE value judgment algorithm in the

Reference Implementation was designed to incorporate object tracking data provided by the MO component.

### 5.3. TSS<sup>[3]</sup>

The TSS has been proven to be able to detect obstacles such as cars and pedestrians as well as give smoothness information about the road surface. Curbs are detected and marked as less traversable than the smoother road surface. While a car traveling within the road lanes is not expected to come across rough terrain, the ability to detect this condition is worthwhile in the case of pot holes, debris on the road, or other unexpected objects. The obstacle detection features are also critical to avoiding cars, pedestrians, signs, fences, k-rails, cones, and other obstacles, all of which are easily detected and marked.

While many aspects of navigation in an urban environment can be accomplished using a priori data such as the GPS coordinates of lane centers and intersections, sensors are required to gather unknown information about the roads and account for the highly dynamic environment caused by other vehicles and pedestrians. LADAR sensors have proven to be excellent devices at gathering this information real time, and the accuracy of the sensor and relative invariance to ambient lighting makes them conducive to use in autonomous vehicle navigation.

### 5.4. WMKS<sup>[4]</sup>

The dynamic world model does not provide any reasoning or analysis of the estimator solutions. Rather, the solution, as found by a prediction algorithm, is reported as-is. Many times this is the proper behavior. However, some form of oversight or regulation functionality could provide added value to the system. For example, while an object may move hundreds of meters through the course of its observed behavior, it is very often not going to do so instantaneously, or near-instantaneously. The capability in the world model to detect situations where a prediction has a high likelihood of being incorrect could either prevent those situations or at least inform a client that such an error may exist. Similar capabilities could (and perhaps should) be implemented in the prediction algorithms themselves. Building this into the primary world model framework would provide basic oversight to all prediction methods deployed.

Another key part of future work for the polynomial predictor algorithm is the evaluation of different solutions. With the current approach of analyzing a number of different “windows,” the one with the lowest order is selected as discussed previously. The reason the lowest order polynomial is selected is because higher-order polynomials tend to exhibit much larger errors when used for extrapolation. However, it is possible that these higher order polynomials might provide better estimation around trend changes because they address the nonlinearity of the data at those points. Therefore it is hypothesized that some other metric for the evaluation of the appropriate solution could be used and yield better results than the current method.

In more recent work, the WMKS has developed into a reliable and elastic, centralized repository of data. Several projects have relied on its ability to relay and define information for successful autonomous robotic mission completion. The software and the framework developed to modify system schemas will be used in the future to model autonomous robotic systems and speed their development.

Currently, the knowledge store serves as a centralized repository for subsystem components. Future work will involve extending the knowledge store so that system level components can accurately meld information through established techniques already relied upon in enterprise data storage environments. Work can also be done to update messaging and behavioral aspects when the final AS4 knowledge store specifications are published. No significant messaging changes are expected, but draft protocol behavioral definitions were not available while developing the knowledge store, so messaging changes will most likely need to be modified.

### **5.5. SLAM and Object Tracking<sup>[5]</sup>**

A novel method for performing simultaneous localization, mapping, and moving object tracking has been tested. Also, formalized methodologies for using an external WMKS and fusing data from multiple LADAR were introduced. In general, SLAM and DATMO have been treated separately with little consideration given to the interaction between static and dynamic elements. Also, most SLAM approaches generated either a point or feature map, which had no real-world interpretation. They also did not attempt to detect differences between the map and the sensed environment and simply matched the detected points or features for localization purposes.

The presented work introduced a method for generating an object map which has a real-world interpretation and can be enhanced with contextual attributes such as object classifications, images, etc, using a spatial reconstruction approach. The approach extended and refined an object's representation as the viewing angle changed and more information about the object was known. The map was constructed in the presence of moving objects while simultaneously estimating the vehicle's current position and orientation. Currently, the execution time of the code (which can reach up to .35 seconds) is still a limiting factor, and data fusion and the accuracy of object representation have room for improvement.

### **5.6. Vision Based Lane Finder and Path Finder<sup>[6]</sup>**

The Lane Finder and Path Finder Smart Sensors are valuable tools which can aid in urban navigation. Cameras are relatively inexpensive sensors and provide a wealth of information about the surroundings. The Lane Finder's ability to operate in both daylight and night time make it very versatile. While the vehicle's navigation is still primarily GPS waypoint driven, the use of the painted lane lines is perhaps more reliable for staying within the lane and is how a human would drive.

There are still problems that prevent the navigation from relying on this output completely for staying within the lane (shadows, faded paint, and absence of lines), but it does provide an excellent addition to the planning component's inputs. This can be especially useful when a GPS offset is experienced because the planner can follow the lane center provided by the vision system as opposed to blindly following the GPS coordinates.

### **5.7. RD<sup>[8]</sup>**

The DFRHC based planner has proven reliable in generating control commands for a UGV in an unstructured environment. The implementations have controlled UGV's through proscribed

missions while avoiding obstacles and areas of reduced traversability. Computational requirements are low enough to be met with commercial off the shelf (COTs) type resources. This planner has been augmented by improving the A\* search and augmenting the traversability representation to include time varying traversability issues (moving obstacles).

## **6. RECOMMENDATIONS**

### **APF**

Naturally, during the course of the current work there were a number of areas identified that present opportunities for further research. One ongoing research topic is how the framework will address conflict resolution, such as would be the case if two Specialists were arriving at opposing or incompatible conclusions. For conflicts that are foreseeable, this can be addressed by devising rules that explicitly resolve the conflict. This might be appropriate when two different styles of perception could reach conflicting Findings. Another research area is that of truth maintenance, which refers to the viability and “shelf life” of Findings and decisions over time. Future researchers may want to explore the benefits of periodic confirmation of the current state and selecting a “safe” or “conservative” default state when no state can be definitively chosen. A final area of continuing research has to do with the assurance of continuity, stability, and safety during behavior transitions. The framework in its current state does not address how the transition from one behavior to another would affect the performance of the vehicle (or its individual components) during the transition. Future researchers should attempt to devise a mechanism that incorporates the resolution of discontinuities and instabilities into the Adaptive Planning Framework as they pursue the design of more complex behaviors and contemplate how one would transition among them.

### **CRMF**

Researchers at the CIMAR will continue utilizing the CRMF while developing advanced technologies that broaden the frontiers of autonomous ground vehicle application. Currently researchers are developing algorithms for simultaneous localization, mapping, and object tracking using multiple 2-D laser scanner sensors. Not only can the framework be utilized to model and monitor these resources, it can intelligently select the most appropriate resources to utilize as inputs to the mapping and tracking algorithms should other applications require the intermittent use of a particular scanner. Furthermore, the PE value judgment algorithm in the Reference Implementation was designed to incorporate object tracking data provided by the MO component. In addition to refining the object tracking and motion classification sensor, new technologies such as an object classifier have yet to be deployed. This classification capability will use the image captured using the Photographer Application Analyst and Panning Camera component along with other knowledge in an attempt to discern the nature of the object. Subsequently, the classification process may require the use of other resources onboard to gain a better understanding of the object. The creation of such requests is now possible using the Reference Implementation and the framework tools developed in this work as a blueprint.

### **TSS**

Further development of the TSS is merited. Improvements to the system such as time stamping the incoming data points would provide better registration of the points in 3D space and thus more accurate results. Time stamping points would also provide a better way to decay old data in order to account for moving objects in the environment. Other areas of study that could be explored include fusing camera data with the LADAR data to improve environment classification, using a 3D LADAR such as those produced by Velodyne or Ibeo, and constructing large scale terrain maps of the environment.

## **WMKS**

The WMKS developed under this contract currently supports all existing requirements. It has seen significant testing in real world environments and has shown several years of reliable results. Several areas could be investigated for future development.

One area of interest for future research would be the idea of distributed management and information sharing between multiple groups of robots. Several published and unpublished platforms exist that already supply production level support of distributed data replication across networks. Data sharing between robots is crucial for the future of autonomous robot systems as robotic platforms are given more complex tasks that require data rich representations. Most industry implementations use non-relational database management systems (DBMS) as a backend, so the effort to transition the current WMKS system to these NoSQL paradigms would require significant effort. Another route could be designing an implementation from scratch using these ideas for possible integration with the AS4 standard.

Another area of possible future investigation is parallel querying topologies. Certain systems need deterministic first-in first-out queuing of interactions with the WMKS. However, other systems may not require this functionality and could benefit significantly from a parallelized query/response infrastructure. This would allow short, relatively quick interactions to be done while long, complex querying is being done on different threads. This does not reference the actual DBMS interaction which must lock/unlock the database to insure consistency, but rather the packing and unpacking of the information for translation to the JAUS system. This packaging requirement has been shown to be a significant bottleneck for certain queries including large geometric representations.

## **Vision Based Lane Finder and Path Finder**

The Lane Finder has potential to be a primary navigational input instead of simply a corrective input. Some problems need to be overcome before the robustness of the system is good enough to drive the vehicle within lanes without GPS at all. Bike lanes can cause problems as well as other lane markings such as painted medians with high curvature and faded lane lines. One feature which would be valuable is being able to identify all the lanes across the entire roadway and determining which lane the vehicle is currently in. Presently, the Lane Finder only reports the geometry of the currently occupied lane, but producing information about which lane the vehicle is in would be critical to navigational tasks such as changing lanes, entering turn lanes, and passing.

## **SLAM and Object Tracking**

There is ample room for improvement in the performance of the characterization and prediction of moving objects. There were also significant challenges in using multiple LADARs when the data was not synchronized, and their alignment with the vehicle's coordinate system was not perfect. Working towards better registration of data would alleviate some of these problems. Using feedback from the SLAM algorithm to correct the GPOS results being reported to other components would also be a useful feature. Recommendations for future work would first entail improving the performance of the current algorithm before exploring other possibilities such as performing SLAM or detection and tracking of moving objects with 3D LADAR data. Further work to create a prediction of a moving objects trajectory (besides a simple first order

extrapolation) could be accomplished through machine learning techniques. A push to improve the performance of this software to the point where it can be reliably used in an urban environment is merited.

### **Local RD**

The existing implementations of the RD components are functional, but several issues could be improved. Effort to reduce the time to implement the reactive driver on a specific vehicle would simplify the deployment of the technology. It could also improve fault tolerance. Some effort applied to the characterization of obstacles and the UGV in the traversability grid could improve near obstacle maneuvering.

The RD utilizes a parameterized kinematic model to develop control inputs for the UGV. Further, several parameters are utilized to predict the dynamic behavior of the vehicle for use in command generation. A process could be developed to aid in the identification of these parameters. One goal of the procedure would be minimum human input during training. The same process could be modified to run on line and identify when the vehicle performance is outside expected ranges, thus allowing high level autonomous intervention when the vehicle is malfunctioning.

The obstacle avoidance process implemented relies on the dilation of obstacles in the traversability representation and the modeling (for collision purposes) of the vehicle as a point. Current computational resources are available that could allow a more precise model of the vehicle and environment to be used to identify valid trajectories through the planning space. This would also improve the usefulness of the RD with respect to vehicles with payloads such as trailers or tools. The RDs are implemented such that this improvement could be added as an extension/augmentation with small changes to the existing implementation.

## 7. REFERENCES

1. Touchton, R., "[An Adaptive Planning Framework for Situation Assessment and Decision-Making of an Autonomous Ground Vehicle](#)," Ph.D. dissertation, University of Florida, 2006.
2. Garcia, G., "[A Cognitive Resource Management Framework for Autonomous Ground Vehicle Sensing](#)," Ph.D. dissertation, University of Florida, 2010.
3. Solanki, S., "[Development of Sensor Component for Terrain Evaluation and Obstacle Detection for an Unmanned Autonomous Vehicle](#)," Ph.D. dissertation, University of Florida, 2006.
4. Kent, D., "[Storing and Predicting Dynamic Attributes in a World Model Knowledge Store](#)," Ph.D. dissertation, University of Florida, 2007.
5. Johnson, N., "[Simultaneous Localization, Mapping, and Object Tracking in an Urban Environment using Multiple 2D Laser Scanners](#)," Ph.D. dissertation, University of Florida, 2010.
6. Lee, J., "[Vector-Based Ground Surface and Object Representation Using Cameras](#)," Ph.D. dissertation, University of Florida, 2009.
7. Galluzzo, T., "[Simultaneous Planning and Control for Autonomous Ground Vehicles](#)," Ph.D. dissertation, University of Florida, 2006.
8. Thorn, E., "[Autonomous Motion Planning Using a Predictive Temporal Method](#)," Ph.D. dissertation, University of Florida, 2009.

## LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

AA	application analyst
AFRL	Air Force Research Laboratory
AGV	autonomous ground vehicle
AI	artificial intelligence
APF	adaptive planning framework
CA	communicator analyst
CIMAR	Center for Intelligent Machines and Robotics
COTS	Commercial off the shelf
CRMF	cognitive resource management framework
DARPA	Defense Advanced Research Projects Agency
DATMO	detection and tracking of moving objects
DBMS	database management systems
DFRHC	dual-frequency receding horizon control
DSA	diagnostician/systemizer analyst
GHz	gigaHertz
GPOS	global position and orientation sensor
GPS	Global Positioning System
GUI	graphic user interface
HRHC	heuristic receding horizon control
Hz	Hertz
J AUS	Joint Architecture for Unmanned Systems
LADAR	light detection and ranging (also LIDAR)
mph	miles per hour
MO	moving object
NPT	n-point turn
OD	obstacle detection
PE	plant engineer
PTMP	predictive temporal motion planning
RA	resource analyst
RAA	resource appraiser analyst
RD	reactive driver
R2C2	Robotic Range Clearance Competition
RHC	receding horizon control
RN	roadway navigation
RSTA	reconnaissance surveillance and target acquisition
SAE	Society of Automotive Engineers
SLAM	simultaneous localization and mapping
TE	terrain evaluation
TSS	terrain smart sensor
UF	University of Florida
UDP	user datagram protocol
UGV	unmanned ground vehicle
UTM	Universal Transverse Mercator
WMKS	World Model Knowledge Store