2011 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
ROBOTIC SYSTEMS (RS) MINI-SYMPOSIUM
AUGUST 9-11 DEARBORN, MICHIGAN

A NON-CONTEXTUAL MODEL FOR EVALUATING THE AUTONOMY LEVEL OF INTELLIGENT UNMANNED GROUND VEHICLES

Phillip J Durst
Wendell Gray
US Army ERDC
3909 Halls Ferry Road
Vicksburg, Ms 39180

Michael Trentini
Defence R&D Canada - Suffield
PO Box 4000, Station Main
Medicine Hat, AB, Canada T1A 8K6

ABSTRACT

A simple, quantitative measure for encapsulating the autonomous capabilities of unmanned ground vehicles (UGVs) has yet to be established. Current models for measuring a UGV's autonomy level require extensive, operational level testing, and provide a means for assessing the autonomy level for a specific mission and operational environment. A more elegant technique for quantifying UGV autonomy using component level testing of the UGV platform alone, outside of mission and environment contexts, is desirable. Using a high level framework for UGV architectures, such a model for determining a UGV's level of autonomy has been developed. The model uses a combination of developmental and component level testing for each aspect of the UGV architecture to define a non-contextual autonomous potential (NCAP). The NCAP provides an autonomy level, ranging from fully non-autonomous to fully autonomous, in the form of a single numeric parameter describing the UGV's performance capabilities when operating at that level of autonomy.

INTRODUCTION

The field of robotics and intelligent systems has grown explosively over the last decade, and unmanned ground vehicles (UGVs) are being fielded with increasing frequency for military applications. However, as a consequence of this rapid advancement, a lack of agreed upon standards, definitions, and evaluation procedures for UGVs exists. Specifically, no agreed upon method for assessing an intelligent UGV's level of autonomy has been established. Several models have been proposed, including the Autonomy Levels for Unmanned Systems (ALFUS) [1] and Performance Measures Framework for Unmanned Systems (PermFUS) [2], which together provide a means of retrospectively assessing a UGV's performance for a specific mission and environment. The drawback of the ALFUS is that it is a highly context-sensitive method, and it requires metrics to be measured not only for the UGV but also for its operator and mission environment. Moreover, many of the metrics needed to evaluate environmental and operator concerns for the ALFUS have yet to be determined. A simpler method for measuring a UGV's autonomy level which is derived from only the UGV itself is desirable, because such a measure could be calculated without first performing extensive operational level testing, and this autonomy level could be compared across platforms without the added caveats of environmental factors.

In search of a simple model for UGV autonomy level, an in depth review of current UGV architectures and their components was undertaken. This review, presented in Section 2, led to the development of a high level, generic model for intelligent UGV architectures. This new UGV architecture model is also presented in Section 2, and it forms the backbone of the new model for assessing UGV autonomy level presented in Section 4.

Similarly, Section 3 presents a review of current UGV performance assessment methods, including current test procedures and metrics. The shortcomings of current methods are highlighted, and models for developing UGV performance assessment tests are discussed. UGV performance assessment methods are discussed within the context of the architecture model developed in Section 2.

Using the UGV architecture model presented in Section 2 and the performance evaluation procedures presented in Section 3, a new model for measuring UGV autonomy level was developed. The model provides a predictive measure of a UGV's ability to perform autonomously rather than a retrospective assessment of UGV performance. UGV autonomy level is determined outside of a mission or environmental setting, and is therefore termed the non-contextual autonomy potential (NCAP). Section 4 provides details about the derivation and application of the NCAP for determining UGV levels of autonomy.

UNMANNED SYSTEMS (UMS) ARCHITECTURES

The architecture of a UMS defines what components are present within the UMS and the relationships between these components. UMS components are the stand-alone hardware and software units of the UMS, such as a sensor, a computer processor, or a software algorithm. Each UMS has its own unique architecture which has been optimized based on the UMS's mission and platform. Several examples of mission/platform specific architectures can be found within the literatures, such as [3] and [4]. Furthermore, a literature review reveals two high level models of UMS architectures: the Joint Architecture for Unmanned Systems (JAUS) and the 4D/RCS reference architecture.

JAUS

The Joint Architecture for Unmanned Systems (JAUS) is a standard messaging architecture for unmanned systems [5]. JAUS was designed to promote interoperability between UMS subsystems and provide reusability and standardization for UMS platforms [6]. JAUS is based on a hierarchical organization, an overview of which is shown in Figure 1.

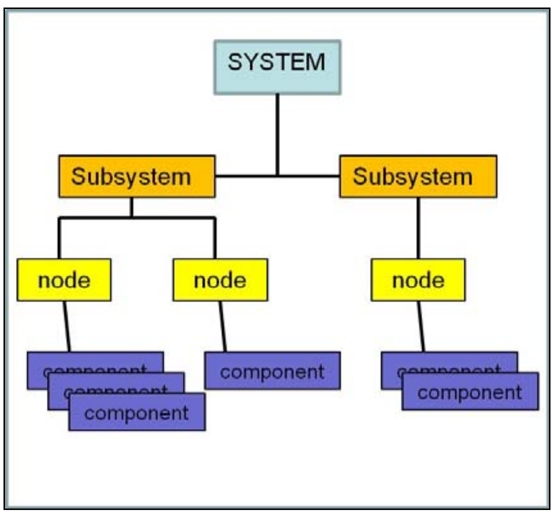

Figure 1: The hierarchical structure of JAUS. JAUS defines the format and structure of the messages passed between the components of the architecture.

In JAUS, a system is comprised of multiple subsystems, with subsystems being self-contained entities. Examples of subsystems would be a single UGV or an operator control unit (OCU). Subsystems contain nodes, which are control systems. The nodes control the components, and the components are the physical systems that perform specific functions. An example of a node and component would be a motor controlling a panning LADAR sensor.

The ultimate purpose of JAUS is to define how hardware and software components within a UMS interact and how UMSs interact with each other by defining standard message formats. JAUS lies on the lines connecting the components to the nodes, the nodes to the subsystems, and the subsystems to the system. The actual internal structure of the entities within the JAUS framework (UGVs, OCUs, sensors, algorithms, etc) is outside the scope of JAUS. A model for these internal architectures is provided by another source: the 4D/RCS.

4D/RCS REFERENCE ARCHITECTURE

The 4D/RCS architecture is a high level architecture model developed for UGVs under the Demo III Experimental Unmanned Vehicle program [7]. Unlike JAUS, the 4D/RCS does provide a framework for the internal structure of a UMS. The 4D/RCS architecture is a multi-layered hierarchy of computational nodes, with each node containing four layers: sensory processing, world modeling, value judgment, and behavior generation. Each node contains a planner module which accepts command inputs. The planner proposes plans, the world modeling predicts the outcomes of these plans, the value judgment evaluates these outcomes, and behavior generation selects the best plan [8].

Figure 2 shows the internal structure of a node. Each subsystem within a UGV would contain nodes, such as a subsystem controlling communication with an OCU or the subsystem controlling the movements of the UGV platform. The main benefit of the 4D/RCS model is that it closes feedback at every level of the architecture, and an open interface exists between deliberative and reactive execution in every node at every level.

The 4D/RCS approach can be applied to individual subsystems within a UGV, and it has been successfully deployed on the Demo III experimental ground vehicle (XUV) for world modeling and map building purposes [9] as well as path planning for an autonomous mobile robot [10]. The modular nature or the 4D/RCS allows specific components and their interactions to be placed inside the architecture to create a best solution for a given UGV platform and mission. However, the 4D/RCS itself forms a subsystem architecture that can be implemented within a larger UGV framework. A higher level, more general model of the architecture for a complete UGV entity is desirable.

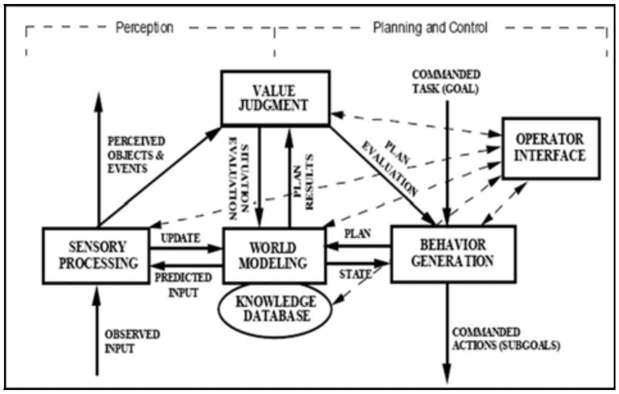

Figure 2: Internal structure of a node within the 4D/RCS architecture. Data is passed between each architecture layer, thus providing feedback at every level. A complete UGV architecture would be comprised of a hierarchy such nodes, each one controlling a subsystem of the UGV.

A GENERIC, HIGH LEVEL UGV ARCHITECTURE MODEL

Looked at from a high level, all UGV architectures can be broken down into four basic layers: perception, modeling, planning, and execution. Sensors provide the UGV with raw data related to the UGV's operational environment. Software onboard the UGV then abstracts this raw data into an internal model of the UGV's surroundings. This model is then used by other software algorithms to generate a plan of action for the UGV. Finally, a plan is chosen and executed. This high level model, shown in Figure 3, provides a non-hierarchical, broad description of how an intelligent UGV operates.

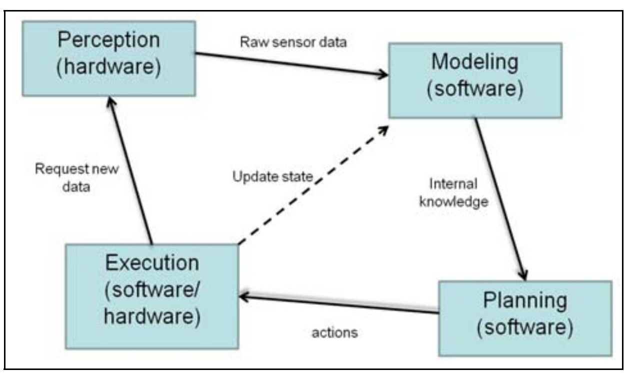

Figure 3: A high level, non hierarchical model for intelligent UGV architectures.

The perception layer of the architecture involves the sensing of the physical environment. It is made up of hardware sensor systems, such as LADAR for sensing the environment's geometry or GPS for determining UGV position. The perception layer of the architecture produces raw data, and this data is sent to the UGV's software systems.

The modeling layer of the UGV architecture is where the raw sensor data is processed. Software is used to turn the raw data into an abstract model of the UGV's surroundings. Modeling includes tasks such as map generation, obstacle detection, or any mission specific software, such as IED or pedestrian detection. After the UGV has created an internal knowledge of its surroundings, it uses this model to plan possible actions.

The planning aspect of the UGV architecture is comprised of the software that is responsible for making decisions based on the UGV's internal knowledge. This layer of the architecture fuses the UGV's world model with higher level knowledge, such as mission goals and safety concerns (i.e. rules of the road). The planning software must pick a best course of action based on mission goals and the UGV's immediate surroundings.

After a suitable plan has been chosen by the planning level of the architecture, it falls to the execution layer to make this plan happen. The execution layer of the architecture is made up of both hardware and software systems. It includes the UGV platform (motor servos, wheels, etc.) and the software used to control the UGV platform. After execution, the UGV must update its state within its world model and return to the perception level of the architecture.

This model presents a coarse understanding of how an intelligent UGV operates. Of course, there are many exceptions that do not fit perfectly within this framework. For most UGVs, there is not such a clear delineation between each level of the architecture. Often, perception, modeling, planning, and execution all happen simultaneously. Still, the presented model provides an elegant break out of the four basic layers necessary for a UGV to operate autonomously and their interactions. This architecture model provides the basis for the autonomy level discussed in Section 4.

UGV PERFORMANCE ASSESSMENT

Currently, UGV performance assessment is done at multiple levels using many different methods. Individual component level testing is done on UGV hardware and software systems, while overall UGV performance assessment is typically done at the mission level. In general, component level testing can be categorized using the architecture model presented in the previous section.

Testing of UGV hardware involves testing the components of the perception and execution aspects of the UGV architecture. This includes the testing of individual sensors, the capabilities of the UGV platform, and human/robot interaction concerns. Evaluation of UGV software systems takes place at the modeling and planning levels and involves testing algorithms for accuracy, efficiency, etc. Performance metrics and standard test procedures for component level UGV testing are lacking. Testing is usually done on a case by case basis with wide variation existing between experiments. Many modified legacy tests exist for evaluating UGV platforms addressing issues like mobility [11], and much effort has been given to develop metrics for quantitatively assessing UGV algorithms, for example [12] and [13].

In addition to component level testing, several models have been proposed for assessing overall UGV performance as a function of mission effectiveness and autonomy level. The most commonly referenced model for assessing UGV performance is the ALFUS, which provides a framework for determining a UGV's level of autonomy. The ALFUS is not a specific metric, but rather a model of how several different metrics could be combined to generate an autonomy level. The autonomy metric provided by the ALFUS is called the Contextual Autonomy Capability (CAC), which is a three axis system that combines tests related to mission complexity, environmental complexity, and human independence. Figure 4 shows the CAC.

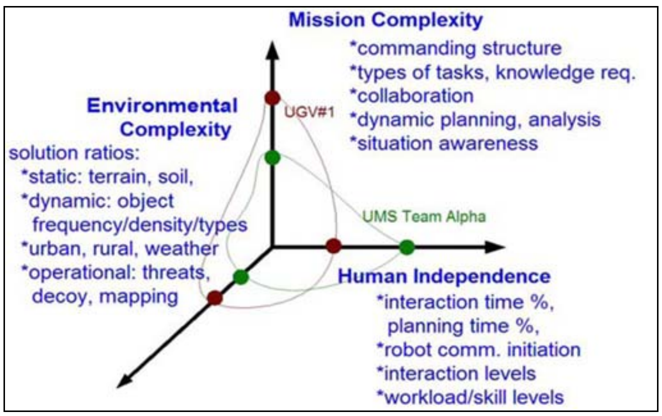

Figure 4: The ALFUS Contextual Autonomy Capability (CAC). For a given mission and environment, metrics are measured for the mission complexity, environmental complexity, and human independence of the UGV, and these metric scores are combined to form a level of autonomy.

The ALFUS faces several outstanding issues. The test required to generate scores for each axis of the CAC do not exit. The PermFUS workgroup is attempting to create these metrics, but very little has been filled in to date. Moreover, a mathematical model for combining the scores along each CAC axis to create a single number for level of autonomy has yet to be established.

Another model for generating UGV performance evaluation procedures was proposed in 2010 by the US Army Engineer Research and Development Center (ERDC) [14]. The ERDC model attempts to derive component level tests from a mission specific context. By observing UGVs performing a given mission, metrics are created based on mission typical events. The goal of the model is to find simple metrics that, when measured for a UGV, provide higher metric scores for better mission performance capabilities.

For now, most UGV testing is carried out on a case-by-case basis. All the metrics currently found in the literature come from this type of testing. For a given UGV and mission, testers guess at some mission aspects to measure (time to complete the mission, operator strain, etc.), and make some general guess at UGV effectiveness based on the results of these tests. A tool that could provide a level of autonomy based on these case by case tests while allowing comparison across systems without having to explicitly address all possible mission and environmental issues is desirable. Such a framework for determining UGV autonomy level is presented in the next section.

THE NON-CONTEXTUAL AUTONOMY POTENTIAL (NCAP)

Current work is ongoing in developing an autonomy level metric using the generic UGV architecture model presented in Section 2. While the ALFUS provides a robust performance assessment tool, a simpler metric that can be applied to current, case-by-case testing methods is desirable. Using the generic UGV architecture model presented in Section 2, a non-contextual, quantitative metric for UGV level of autonomy has been derived. The presence and complexity of each level of the architecture presented in Figure 3 determines the UGV's level of autonomy. Each aspect of the UGV architecture contains its own metrics, measured from component level tests, and the metrics measured at each architecture level are used to generate an overall autonomy level metric. As this autonomy level is measured outside of a mission and environment specific setting, it is termed the non-contextual autonomy potential (NCAP).

The NCAP defines four autonomy levels (AL). The AL ranges from 0, fully non-autonomous, to 3, fully autonomous. A UGV's AL is defined within the context of the generic UGV architecture model. A UGV that only contains perception, i.e., a teleoperated UGV with an onboard camera, has no autonomy. The UGV simply collects data about its surroundings but does nothing with this data. A UGV that generates some sort of world model or retains an internal knowledge of its surroundings is considered semi-autonomous. At this level, the UGV is interpreting the raw sensor data on its own. A UGV that uses its world model to form a plan of action is considered autonomous. At this level, the UGV is making a judgment based on its internal knowledge base. Finally, a UGV that chooses a best action based on its modeling and planning and performs that action without any operator input is considered fully autonomous. Figure 5 shows the NCAP and AL within the context of the architecture model.

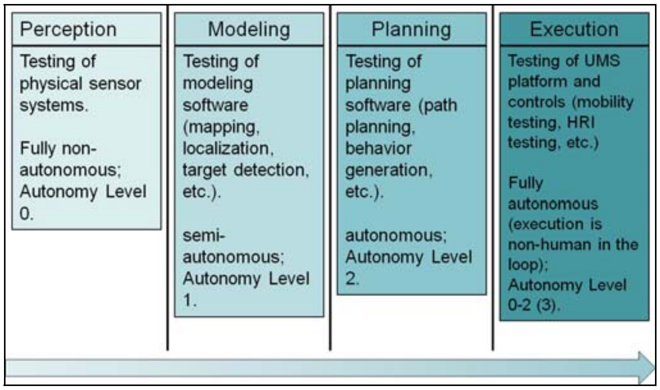

Figure 5: The NCAP Autonomy Levels within the framework of a generic UGV architecture model. Also shown are the types of component level tests done at each architecture layer.

Because execution is implicit in all UGVs, regardless of autonomy level, a UGV's AL is defined by the architecture level at which a human interacts with the UGV. So, a UGV with LADAR and camera sensors that is driven entirely by teleoperation would be AL 0. If that same UGV used its LADAR and camera data to generate a world map but still required teleoperation to move through the environment, its AL would be 1. If software were added which enabled the UGV to plan paths using a world model and subsequently asked the user to select the best path, it would have an AL of 2. A UGV would only be considered AL 3, fully autonomous, if it required no human input during its mission.

The NCAP uses test scores for each architecture aspect to form a single metric for UGV AL. Testing done within the NCAP framework are performed on individual UGV components and do not require mission level evaluations. The goal of the NCAP is to provide a means of combining component and engineering level tests into a predictive measure of UGV autonomous performance. Therefore, the NCAP does not provide an evaluation of UGV autonomous performance; rather, it encapsulates a UGV's potential to operate autonomously.

Scores for testing done at each architecture level can be combined to generate an overall NCAP score. This combined score could be compared between UGVs to assess which UGV is more capable of autonomous mission performance. For example, a fully autonomous UGV with very high test scores could have an NCAP score of 3.7, while another UGV that performs very poorly might have an NCAP score of 3.1. However, even a UGV that fails 100% of the time at its mission but is operated fully autonomously would still have an NCAP score of 3.0. The variation in score between ALs is based on UGV performance, but AL itself is fixed by the level at which a human must interact with the UGV.

COMPARISON OF THE NCAP TO THE ALFUS

The NCAP differs from the ALFUS in several ways and has many benefits over the ALFUS. Most importantly, the ALFUS is a retrospective measure of autonomy. For a given mission and environment, metrics based on mission performance must be measured and combined to generate the autonomy level at which the UGV performed the given mission. The ALFUS is primarily a tool for evaluating UGV performance. On the other hand, the NCAP is based solely on the UGV platform itself. Metrics based on component level testing of the UGV are combined to provide the highest level of autonomy attainable by the UGV. The NCAP is meant to serve as a tool for predicting UGV autonomous performance potential.

The benefit of the NCAP is that it provides a simple tool for rapidly determining a UGV's autonomy level (AL). Because the NCAP is based only on the UGV hardware and software, it is much easier to evaluate. The NCAP reduces the emphasis on the hard to quantify "human element" and constrains the nearly endless environmental and mission concerns. Testing done on the UGV hardware and software systems will encompass human and environmental factors without having to make them implicit concerns. Testing of the UGV sensor systems will be determined by the environment. Testing of the OCU, if present, will include any necessary human-robot interaction concerns, and so on.

The greatest benefit of the NCAP is that it can be determined using case-by-case testing, which is how all UGVs are currently being tested. Tests can be developed for the UGV's hardware and software based on the desired mission. These tests can be broken out according to the architecture layer they involve (testing of sensor benchmarks, evaluation of mapping algorithms and path planners, etc.). After running tests, the separate test scores can be combined to generate an AL and NCAP score. The NCAP has been designed such that it does not require the way UGVs are being tested to change radically but instead creates a framework for taking current UGV test methods and generating a more meaningful metric.

CONCLUSIONS AND FUTURE WORK

Autonomy level has proven to be a very difficult metric to quantify, let alone measure, for intelligent UGVs. While much effort has been given to this question, an agreed upon method for defining what makes one UGV more autonomous than another and quantifying its performance at that level of autonomy has yet to be established. In an effort to provide an answer to this question, a new model for assessing UGV autonomy level was developed.

Using a high level framework for intelligent UGV architectures, a non-contextual measure of UGV autonomy level, the NCAP, was developed. The NCAP provides a predictive, single number measure of UGV autonomous performance potential. It can be measured using testing of only a UGV without having to address environment and mission concerns explicitly, and it provides a simple means of comparing the autonomous capabilities of different UGVs.

As with the ALFUS, specific tests and metrics for the NCAP have yet to be established. And also like the ALFUS, the mathematical model for combining test scores into a single number metric is, as of this writing, undefined. As testing and evaluation of UGVs for military applications continues, the results of these tests will be incorporated into the NCAP framework. The ultimate goal of the NCAP is to provide a simple tool which can drive the design of UGV testing and provide a simple, efficient, one number descriptor for UGV autonomy level. If it succeeds at this task, the NCAP will serve to greatly enable the testing, development, and deployment of intelligent UGVs.

ACKNOWLEDGEMENTS

Permission to publish was granted by Director, Geotechnical and Structures Laboratory.

REFERENCES

[1] H. Huang, Kerry Pavek, James Albus, and Elena Messina, “Autonomy Levels for Unmanned Systems (ALFUS) Framework: An Update”, 2005 SPIE Defense and Security Symposium.

[2] H. M. Huang, E. Messina, and A. Jacoff, "Performance measures framework for unmanned systems (PerMFUS): initial perspective", Proceedings of the 9th Workshop on Performance Metrics for Intelligent Systems, 2009.

[3] P. Doherty, P. Haslum, F. Heintz, T. Merz, P. Hyblom, T. Persson, and B. Wingman, “A Distributed Architecture for Autonomous Unmanned Aerial Vehicle Experimentation”, Proceedings of the 7th International Symposium on Distributed Autonomous Robotic Systems, 2004.

[4] M. Caccia, R. Bono, G. Bruzzone, and G. Veruggio, “Unmanned underwater vehicles for scientific applications and robotics research: the ROMEO project”, Marine Technology Society Journal, vol. 34 no. 2 pp. 3-17, 2000.

[5] AS-4D Unmanned Performance Measures Committee, http://www.sae.org/servlets/works/postDiscussion.do?comtID=TEAAS4D&docID=&forumID=17351&resourceID=134092&inputPage=showAll.

[6] Steve Rowe and Christopher R. Wagner, “An Introduction to the Joint Architecture for Unmanned Systems (JAUS)”, Proceedings for the 2007 Fall Simulation Interoperability Workshop, SISO, 2007.

[7] J. S. Albus, "4-D/RCS: A Reference Model Architecture for Demo III, Version 1.0", NISTIR 5994, National Institute of Standards and Technology, 1997.

[8] J. S. Albus, "4-D/RCS Reference Model Architecture for Unmanned Ground Vehicles", Proceedings of the SPIE 16th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls, 2002.

[9] Tsai Hong, Stephen Balakirsky, Elena Messina, Tommy Chang, and Michael Shneier, "A Hierarchical World Model for an Autonomous Scout Vehicle", Proceedings of the SPIE 16th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls, 2002.

[10] A. Lacaze, "Hierarchical Real-Time Path Planning for Obstacle Avoidance and Path Optimization for One or More Autonomous Vehicles", Proceedings of the SPIE 16th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls, 2002.

[11] Adam Jacoff, “Urban Search and Rescue Robot Performance Standards: Progress Update”, June 2007.

[12] B. Balaguer, S. Carpin, S. Balakirsky, “Towards Quantitative Comparisons of Robot Algorithms: Experiences with SLAM in Simulation and Real World Systems”, IROS 2007 Workshop, 2007.

[13] Ioana Varsadan, Andreas Birk, and Max Pfingsthorn, “Determining Map Quality through an Image Similarity Metric”, RoboCup 2008: Robot Soccer World Cup XII, 2009.

[14] P. Durst, “Performance Evaluation and Benchmarking for Unmanned Ground Vehicles", Proceedings of the 2010 NDIA Ground Vehicle Systems Engineering and Technology Symposium, 2010.