

# A Framework for Predicting Future System Performance in Autonomous Unmanned Ground Vehicles

Stuart H. Young, Thomas A. Mazzuchi, and Shahram Sarkani

**Abstract**—The development of complex self-adaptive systems has accelerated rapidly over the past decade, led by the Department of Defense, which has sought to develop and field military systems, such as unmanned aerial vehicles and unmanned ground vehicles, with elevated levels of autonomy to accomplish their mission with reduced funding and manpower. As their role increases, such systems must be able to adapt and learn, and make nondeterministic decisions. To field such systems requires extensive testing, evaluation, verification, and validation—a challenging task. To address this, we apply a novel systems perspective to develop a framework to predict future system performance based on the complexity of the operating environment using newly introduced complexity measures and learned costs. In this paper, we consider an autonomous military ground robot navigating in complex off-road environments. Using our model and experimental data from Defense Advanced Research Projects Agency-led experiments, we demonstrate the accuracy with which our model can predict system performance and then validate our model against other experimental results.

**Index Terms**—Mobile robots, modeling, systems engineering, unmanned ground vehicles (UGVs).

## I. INTRODUCTION

THE previous decade has seen explosive growth in the deployment of unmanned systems, including unmanned aerial vehicles, which have routinely handled tasks such as surveillance, and unmanned ground vehicles (UGVs), which have regularly provided a safe standoff in the fight against improvised explosive devices, as well as aided in humanitarian search-and-rescue efforts. However, such systems still exhibit only limited autonomy, despite a significant research investment in developing autonomous behaviors. Leading the development of such complex systems is the Department of

Manuscript received November 6, 2015; revised February 6, 2016; accepted April 19, 2016. This paper was recommended by Associate Editor F. Wang.

S. H. Young is with the U.S. Army Research Laboratory, Adelphi, MD 20783 USA, and also with the Engineering Management and Systems Engineering Department, George Washington University, Washington, DC 20052 USA (e-mail: syoung1@gwu.edu).

T. A. Mazzuchi is with the School of Engineering Management and Systems Engineering, George Washington University, Washington, DC 20052 USA (e-mail: mazzu@gwu.edu).

S. Sarkani is with the Engineering Management and Systems Engineering Off Campus Programs, George Washington University, Washington, DC 20052 USA (e-mail: sarkani@gwu.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSMC.2016.2563403



Fig. 1. Crusher autonomous navigation platform from the DARPA UPI program, developed by CMU NREC. Photograph by Mike Perschbacher.

Defense (DoD), which has increased the levels of autonomy, thus reducing the level of human interaction [1]. For these systems to perform autonomously in all environments, recent research has suggested that they must be programmed to adapt and learn instead of having their behaviors hard-coded [2], [3]. As a result, adaptable systems employ machine-learning techniques that enable them to react to their operating environments to achieve desired capabilities [2], [3]. Many techniques have been studied, such as imitation learning, probabilistic planning, model learning, learning from demonstration, and reinforcement learning [4]–[6]. For example, image classification algorithms are used to characterize terrain based on classifiers that have been trained to recognize specific terrain or objects in the environment, such as roads, grass, or foliage. Using training data from one region, these algorithms apply that knowledge to other similar regions, allowing a robot's navigation planners to make decisions about where to move next. Because it is impossible to completely model all possible situations, modern robot algorithms exploit these trained learning algorithms, which benefit from human annotation, demonstration, or prior successful autonomous runs [7], [8].

The Defense Advanced Research Projects Agency (DARPA), along with the military service laboratories, is leading the push to evaluate the performance of autonomous ground robot systems [9]–[12]. One such example, the Crusher autonomous navigation platform,

is shown in Fig. 1. Crusher was developed by the National Robotics Engineering Center (NREC), part of the Carnegie-Mellon University (CMU) Robotics Institute, for the DARPA Unmanned Ground Combat Vehicle (UGCV), Perception for Off-Road Robotics (PerceptOR) Integration (UPI) program [13].

In support of such efforts, the DoD has focused more intensely on developing autonomy requirements for its future systems; however, such systems must still undergo the DoD testing, evaluation, verification, and validation (TEV&V) process, or civilian equivalents, prior to fielding. Thoroughly understanding and evaluating these systems is challenging, not only due to the rigors of TEV&V, but also due to the complexity and changing nature of the environments such systems face. Quantitatively measuring performance requires one to delineate all of the general operational domains of unmanned systems and decide how to determine when an unmanned system satisfies capability specifications and expectations.

When developing TEV&V techniques, the Defense Science Board recommends that such measures focus on the unique challenges of autonomy, specifically predicting system behavior and characterizing the environment in which autonomous system will operate [1]. As such, this paper explores the problem from a novel systems perspective, using a statistical approach in a model-based framework to quantify the environmental complexity and then, from that, predict the future performance of complex, self-adaptive autonomous robot systems. The statistical approach in this paper is agile and could be further extended using many of the widely applied physics-based models.

Many in the systems engineering community believe that, given our present understanding of complex behavior, truly complex systems cannot be designed with a degree of confidence that is acceptable given our current expectations. Consequently, systems engineers as a whole must investigate this issue as a matter of priority and urgency, and develop approaches to respond to the challenge [14]. In response, we have taken a systems engineering approach to create a tool that can predict system performance (as part of the system TEV&V process) and serve as a framework for testing.

To develop our model, we introduce two measures that encapsulate environmental complexity. The first applies to the resistance a system encounters where it operates, including factors such as vegetation, terrain (including elevation data), environmental conditions (including weather), manmade objects, and other agents (human and unmanned). In this paper, we focus on vegetation and elevation data, since the other factors were not present in our testing environment. The second recognizes that most global planners seek to find the minimum cost path through an environment; of course, this measure also relates to resistance within the environment.

To create our model, we use supervised learning, and then demonstrate our ability to predict system performance and validate our model against experimental results of a live robot from the DARPA UPI program. Section II presents related work in robot evaluation efforts and descriptions of environmental complexity and its ability to predict system performance. Section III details our approach and the development

of the model. Section IV describes our results and Section V shows a validation of our results. Finally, conclusions and future work are presented in Section VI.

## II. RELATED WORK IN ASSESSING AND PREDICTING THE PERFORMANCE OF UNMANNED SYSTEMS

To develop our framework for predicting performance in autonomous unmanned ground systems, we examined the literature related to the two principal components of our framework, namely **performance prediction of autonomous vehicles** and **environmental complexity models**. Prior work has sought to characterize the performance of unmanned systems in terms of levels of autonomy, environmental complexity, and experimental efforts; however, **quantitative results have been difficult and scarce**. Further, **approaches that predict system performance are limited and often focus on binary environmental representations**. Because of the limited previous work in predicting the performance of autonomous systems, we investigate related work in robotics evaluation efforts to understand the systems and their performance measures, and in environmental complexity for approaches to model the environment and its complexity. In the following sections, we describe related work in these areas and discuss how our approach leverages and extends this research.

### A. Robotics Evaluation Efforts

Many efforts have attempted to evaluate robot system performance using a variety of performance metrics. Performance comparisons against a human baseline, in challenges and competitions, or using autonomy metrics (such as system speed, level of interaction, or communications bandwidth) are common. Overarching themes include the following: 1) **system performance is strongly linked to the complexity of the environment**; 2) **systems must employ machine-learning approaches to improve system robustness and overcome brittleness**; and 3) **the nature of testing navigation in unstructured, off-road environments makes accurate, objective measurement of progress challenging**.

DARPA has advanced the state of the art in the autonomy of unmanned systems as exemplified by their efforts to push the envelope of autonomous off-road navigation. The UPI program has sought to develop a highly mobile UGV capable of autonomously driving through various demanding, off-road environments. To gauge system performance, DARPA used measures such as system speed, interventions, and bandwidth consumed. These data were made available to us and serve as the basis of the results presented in this paper. The lead system developer, CMU-NREC, relied heavily on machine-learning algorithms, such as imitation learning, to improve system performance; previously, they used hand-tuned parameters, which resulted in system performance that was brittle and often did not achieve the desired results in new environments [3].

Prior to UPI, DARPA focused its research through several programs, including Learning Applied to Ground Robot (LAGR) and PerceptOR. In the PerceptOR program, DARPA conducted six experiments to quantitatively

evaluate the performance of several robots operating on off-road courses of varying terrain types **using a human-operated vehicle as the baseline**. The performance metrics for autonomy encouraged minimal operator interaction: measured system speed, the number of interactions, and the downlink data required between the operator and the robot [9]. The LAGR program took an alternate approach emphasizing learned behaviors. The program used common platforms, where performers provided software upgrades and DARPA measured the performance against a state-of-the-art code base [10]. As is common in robotic navigation planners, the LAGR performers projected observed 3-D maps, with identified obstacles, onto a 2-D map with traversal costs. Although for a different application, the approaches taken by the performers in these programs to develop onboard planners provided significant insight into the approach presented in this paper. However, **our approach shifts focus, instead on predicting performance in different environments based on environmental factors**.

Two common themes emerged from the above evaluation efforts: **1) such efforts are expensive to conduct and 2) they are highly dependent on the environments in which they are conducted**. While these efforts are tremendously useful for advancing the state of the art, they do not provide sufficient confidence in predicting system performance or bridge the gap to the point that such systems can be fielded. More exhaustive testing, in many environments, is essential to the TEV&V process to create trusted fieldable systems. The model framework presented in this paper seeks to bridge that gap.

### B. Environmental Complexity

Recent research has suggested that **environmental complexity is a strong predictor of performance in mobile robot systems** [15], [16]. Measuring the performance of autonomous robots operating in complex and changing environments has been a long-standing issue, and currently there are no satisfactory metrics to measure how precisely a robotic system satisfies performance specifications or expectations, nor are there any metrics that can generally describe the operational domains of the robot, specifically environmental complexity. A comprehensive survey [17] of 3-D terrain traversability analysis methods describes traversability based on the theory of affordances; main streams of research are principally based on sensing approaches focusing on geometry and appearance. The most prominent approaches have evolved from occupancy grid maps and represent terrain as cost maps using grid-based search techniques. These approaches are still widely used in robot planning algorithms and serve as a basis for our model. We combine the tradeoffs between geometry- versus appearance-based approaches in our model, since we consider a component of each, namely, elevation data (including slope) and vegetation (from orthoimagery including texture from which we extract vegetation).

A widely publicized thread of research, the Autonomy Levels for Unmanned Systems (ALFUS) model, seeks to describe the capabilities of robots in terms of levels of autonomy. It is enhanced by the Performance Measures Framework

for Unmanned Systems (PerMFUS) model, which extends the ALFUS model by introducing contextual metrics. In the case of autonomous mobility, the environmental context is related to the performance metrics [18]. The framework uses a multiaxis model based on the mission requirements, the environments in which the missions are performed, and the characteristics of the unmanned systems. However, these frameworks imply that there are discrete levels of intelligence to perform entire missions [1], and further, they do not include quantitative methodologies for predicting system performance.

The work of [16] developed a method for predicting the performance of human–robot teams by determining the system’s neglect tolerance and interface efficiency as functions of world complexity. They estimated world complexity for a navigation task in a maze, focusing on the branching factor and clutter in the environment [19]. The branching factor is calculated from the topological map of the environment and is based on the number of decision points in the maze, which is not directly applicable to the continuous approach presented in this paper. To estimate clutter, directional entropy and velocity changes were used. They combined the contributions of the branching factor and clutter into a single value between 0 and 1 to represent world complexity. We extend the concept of [16] by considering continuous traversal costs in the navigation task to predict system performance using average system velocity as a function of complexity.

Other efforts [20], [21] have investigated the interdependencies between the principal components of autonomous robots: 1) perception; 2) planning; and 3) task execution. However, it is still difficult to assess the effect of environmental variations on system performance given the scarcity of performance indicators. One effort [20] developed a probabilistic model that learns the interdependencies between the three above components based on learning a Bayesian network that identifies the coherence between performance indicators and system outputs. However, the approach requires subsystem performance indicators that were not available for our work, but are likely applicable to our proposed future work. Additionally, the approach is applied to a simple environment represented by a binary cost map, which we extend to a more complex environment using a continuous cost map.

Another quantitative method used simulations of a search task to determine that a combination of entropy and sensor compressibility can describe environmental complexity to a robot [22], [23]. The complexity measure also allows the system designer to determine quantitatively the most important information for improving a robot’s performance [22]. This approach was then used to predict the average number of steps a robot needed to complete a navigation task [23] and employed a discrete/binary map to describe the traversability of the cells of the map. In our model, we extend this idea by using continuous costs to describe the traversal costs of the map cells.

Another method [15], [24] considered the performance of a robot with respect to its environment through the interaction of a set of metrics learned in a dynamic Bayesian network and placed in a probabilistic systems model called a system map. The system map was used to clarify how the metrics

related both to each other and a partially known environment. The results demonstrated how the model identifies environmental dependencies and how performance can be predicted in unknown environments. The interaction of the robot with its environment is considered as a single, complex, and dynamic system. In this paper, environmental complexity was defined using an entropy measure [15]. This research is similar to the approach presented in this paper, but we extend this idea to more complex rough terrain and use continuous measures of environmental complexity.

### III. APPROACH

Many modern autonomous robotic platforms can be generally decomposed into perception, planning, and navigation subsystems. It is the combination of these subsystems and their interaction with the environment that determines system performance. Our focus is on predicting aggregate system performance based on the environment.

It is intuitive that system performance is inversely proportional to the complexity of the environment; therefore, our hypothesis is that by modeling the complexity of the environment in which a system will operate, we can predict the system's performance.

To validate our hypothesis and predict system performance, we developed a framework for modeling the operational environment and the system performance in that environment. From the accepted IEEE definition [25], “a model is an approximation, representation, or idealization of selected aspects of the structure, behavior, operation, or other characteristics of a real-world process, concept, or system, i.e., an abstraction.” Further, model-based systems engineering is defined by the International Council on Systems Engineering as the formalized application of modeling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases [26]. Based on these definitions, we first modeled the environment as a 2-D graph with learned costs of the environmental components from a linear regression approach. Second, we modeled the system performance in the environment model using a graph search, namely, the A\* algorithm to determine the lowest cost path through the graph, from which we computed complexity measures to predict the system performance. Finally, we validated our model by comparing the predicted results with actual results. Details of the model development are as follows.

Our model is based on algorithms used onboard many autonomous robots to perform navigation tasks. An example of such a system is the previously discussed NREC’s Crusher autonomous navigation platform [2], [3], [13]. In that effort, the Crusher system was required to autonomously navigate dense trees, thick vegetation, surfaces with poor traction, rocks, steep slopes, and ditches while maximizing speed. Autonomy was measured by minimizing the communications bandwidth consumed and user interventions [27]. In order to meet system requirements, the Crusher system architecture comprised the physical platform (designed to survive

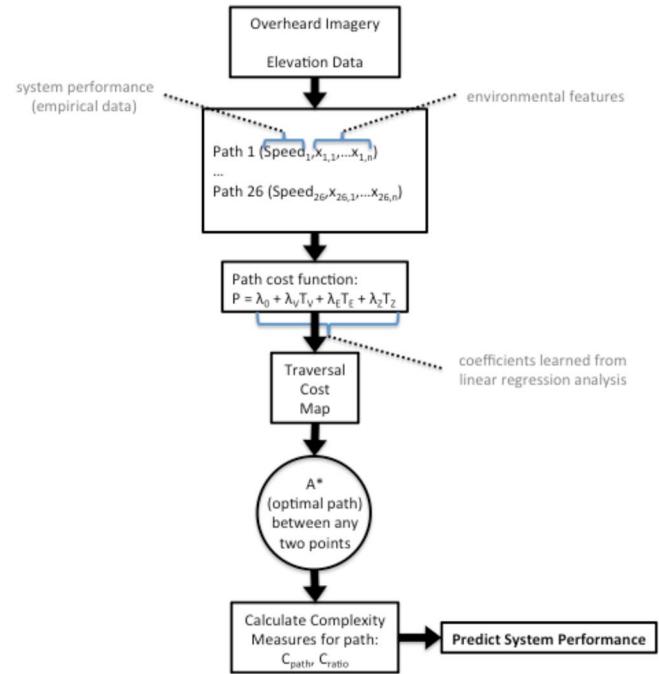


Fig. 2. Summary of approach. The arrows indicate the flow of our process.

undetected obstacles), the associated autonomy sensors, and the software (an overhead map subsystem, perception subsystem, and planning subsystem). The overhead map subsystem primarily provided input to the global planner in the planning subsystem, and the perception subsystem provided local trajectory planning input to the planning subsystem. The planning subsystem then used an A\*-based algorithm (specifically, Field D\*) to globally plan the vehicle’s path [28]. Many robot planning algorithms commonly use an A\*-based, or a similar global planner, to develop an initial course plan for the robot, and then refine its navigation using a local planner that acts based on onboard sensors.

The theoretical basis for this paper leverages the components of the model developed by [3], which features an algorithm that runs an adaptive onboard planner for the Crusher autonomous navigation platform [2] and which implements learning from demonstration to improve system performance and better understand the impact of continuous costs on the autonomous system.

Fig. 2 summarizes our approach, where overhead imagery and elevation data are combined with empirical data to learn the traversal costs. This results in a traversal cost map from which we can predict the system performance of a path based on the A\* estimate of the path through the learned environment. The results section demonstrates that we can predict system performance with less than 10% error on average.

#### A. Modeling Traversal Costs

Each traversal area is divided into cells of  $0.5 \times 0.5$  m by registering overhead imagery data with elevation data in MATLAB. We estimate the traversal cost for each cell by combining contributions from various environmental factors into a continuum of values, where lower costs are associated with more easily traversable routes than higher costs.

These continuous traversal costs are more representative of the costs used by the onboard planners of current autonomous mobility platforms [2] and are an extension of previous work [15], [16], [19], [22]–[24] that used binary cost models, where a cell is either traversable or nontraversable. Overall, the traversal cost is an estimate of the cost for the unmanned system to traverse that cell. Onboard planners determine these costs from *a priori* information, such as overhead imagery and elevation data, and combine this information with onboard sensors to plan the route of the autonomous system [3]. We use only *a priori* data to estimate the traversal costs and then combine them into a traversal cost map on which we estimate the complexity of the environment and predict system performance within that environment. This is analogous to the work of [3] where they used overhead imagery to estimate costs at ranges beyond the limit of their onboard sensors.

In order to develop a traversal cost map for our model, we first decompose the traversal cost into relevant elements. Many 3-D terrain traversability analysis methods, including [3], consider the contribution of geometry- and appearance-based components, thus elements of both are combined in our model, in our case, elevation data (first and second derivatives) and vegetation (overhead orthoimagery). We model the system performance,  $P$ , which consists of system performance data, including speed, number of interventions, bandwidth consumed, and so on

$$P = \lambda_0 + \lambda_V T_V + \lambda_E T_E + \lambda_Z T_Z + \varepsilon. \quad (1)$$

This linear relationship of terms permits us to perform a multivariate linear regression analysis to learn the equation coefficients and develop our model [30]. The first term, following the intercept term,  $T_V$ , is the traversal cost component based on the vegetation in the environment. The next terms,  $T_E$  and  $T_Z$ , are based on the elevation data of the region, specifically, the slope of the terrain. The  $T_E$  term represents regions of high slope that are difficult to navigate and are characterized by terrain such as steep ravines, gullies, or cliffs. The  $T_Z$  term represents flat points in the terrain, where the derivative of the elevation is zero, and is characterized by terrain such as streams, washes, or mud. The  $\varepsilon$  term represents the random error in the approximation [31]. The values  $\lambda_0$ ,  $\lambda_V$ ,  $\lambda_E$ , and  $\lambda_Z$  are coefficients estimated using multivariate linear regression analysis, which is the basis of our approach as described in Fig. 2.

Additional terms may be added to (1) to represent other environmental factors; however, for our region of interest, we are assuming the traversal costs are dominated by the above three factors, thus we assume that any additional terms would have coefficients,  $\lambda$ , of zero. A residual analysis of the model proposed in (1) is then used to confirm this assumption.

1) *Vegetation Costs*: In order to compute the vegetation traversal costs,  $T_V$ , we imported overhead images from Google Earth [32] and considered several state-of-the-art image segmentation approaches to segment the vegetation types, including Otsu's method [33],  $k$ -means [34]–[36], and Gaussian mixture models (GMMs) [37], [38]. We selected a  $k$ -means approach, based on its ease of implementation, and extracted

clusters of terrain that are similar in color, intensity, and texture in a manner similar to [34]–[37] with some influence from the ideas of [39]–[42]. To reduce variance in lighting conditions, we converted the image from the red–green–blue (RGB) color space to the Commission Internationale de l'Eclairage—the International Commission on Illumination (CIE) 1976  $L^*a^*b^*$  color space (CIELAB) [43], which, per the CIE, is a reference that mimics the nonlinear response of the eye to color. In the  $L^*a^*b^*$  color space, the first channel is  $L^*$  is the lightness (or intensity) of the pixels varying from 0 (black) to 100 (white). The color channels are  $a^*$  and  $b^*$ , representing red minus green and yellow minus blue, respectively [44].

Additionally, we computed the entropy of each of the  $L^*$ ,  $a^*$ , and  $b^*$  color channels as a means to detect vegetation based on texture [44]. Since we are interested in extracting similar textures and not in the entropy of the entire image, we determine the entropy,  $H$ , using a statistical method [44] by computing the histogram, (2), of image intensity values in a 15-by-15 patch around each pixel for each of the 3  $L^*$ ,  $a^*$ , and  $b^*$  color channels [45]. In (2), we use 256 bins ranging over the image intensity values (normalized between 0 and 1) for the color channel

$$h_i = \text{no. of pixels in bin } i \quad (2)$$

$$H = -\sum_{i=1}^n \frac{h_i}{N} \log_2 \frac{h_i}{N}. \quad (3)$$

In (3),  $h_i$ , is calculated from (2) and  $N$  is 225 ( $15 \times 15$ ). Fig. 3 shows the 3  $L^*$ ,  $a^*$ , and  $b^*$  channels and the three associated entropy values for each of the channels for a representative region.

Combining the image intensities and textures (measured by the entropy value) in each of the three color channels yields a six-dimensional vector,  $V$ , that describes the vegetation, which is then used to compute the  $k$ -means image segmentation, where  $I_L$ ,  $I_a$ , and  $I_b$  are the normalized image intensity values (ranging from 0 to 1) for the  $L^*$ ,  $a^*$ , and  $b^*$  channels, respectively [see (4)]. Similarly,  $H_L$ ,  $H_a$ , and  $H_b$  are the entropy values (normalized, ranging from 0 to 1) for each of the  $L^*$ ,  $a^*$ , and  $b^*$  channels, respectively. The coefficients,  $\phi_L$ ,  $\phi_a$ ,  $\phi_b$ ,  $\phi_{eL}$ ,  $\phi_{ea}$ , and  $\phi_{eb}$  are the values that we can adjust to give the best clustering of vegetative regions from the image

$$V = (\phi_L I_L \quad \phi_a I_a \quad \phi_b I_b \quad \phi_{eL} H_L \quad \phi_{ea} H_a \quad \phi_{eb} H_b). \quad (4)$$

The  $k$ -means algorithm clustered the pixels in the map by iteratively minimizing the squared Euclidian distance of our  $n$  observations and the cluster centroid [34]. In our case, we have an observation for each of  $N$  pixels,  $V_n$ ,  $n = 1, \dots, N$ . Our goal is to minimize the objective function,  $J$ , such that the observations are clustered into  $k$  clusters, where the squared Euclidian distance between the cluster centroid,  $\mu_k$ , and the observations in the cluster are minimized [see (5) and (6)]. The assigned cluster,  $r_{nk}$ , may change after each iteration until the objective function,  $J$ , is minimized [30]

$$J = \sum_{k=1}^K \sum_{n=1}^N r_{nk} \|V_n - \mu_k\|^2 \quad (5)$$

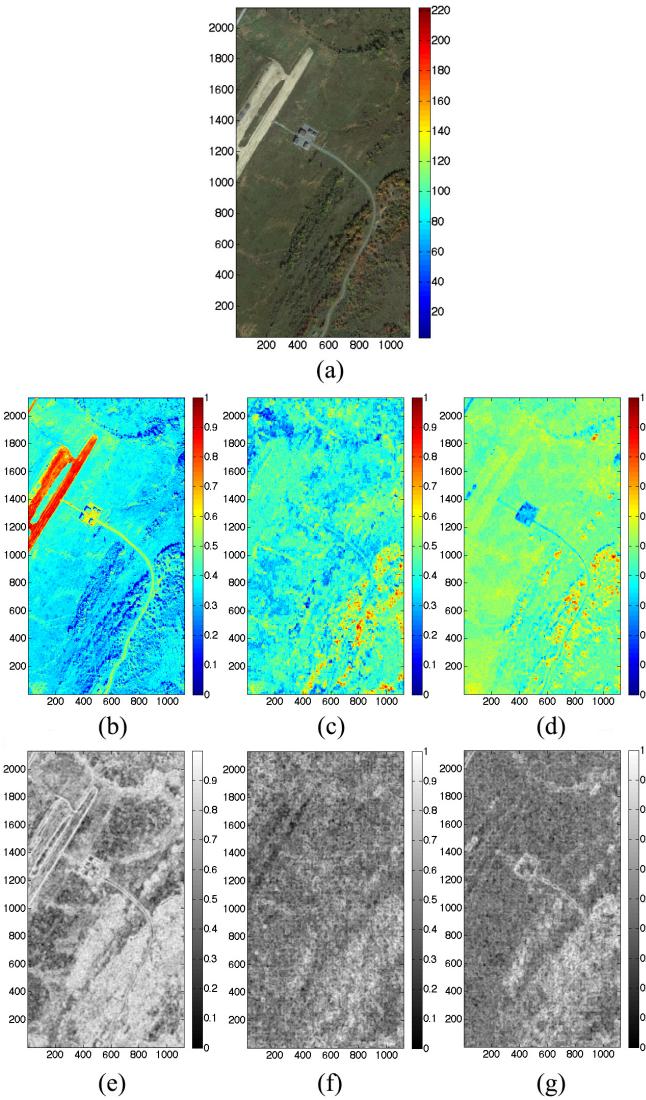


Fig. 3. Example input image and the 3  $L^*$ ,  $a^*$ , and  $b^*$  channels and the three associated entropy values (all images normalized 0 to 1) for each of the channels for a representative region (course 1, waypoints 7 to 8) that comprise the 6-D vector in (4). (a) Input image. (b)  $L^*$  lightness channel. (c)  $a^*$  color channel. (d)  $b^*$  color channel. Entropies of (e)  $L^*$  lightness channel, (f)  $a^*$  color channel, and (g)  $b^*$  color channel.

$$r_{nk} = \begin{cases} 1 & \text{if } k = \operatorname{argmin}_j \|V_n - m_k\|^2 \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

Once the pixels are assigned clusters from the  $k$ -means algorithm, we can segment the image into identifiable vegetative regions. The vegetative regions are assigned traversability costs,  $T_V$ , based on the empirically determined values in Table I and represented as a function in (7), where the vegetative traversability cost,  $T_V(x, y)$ , at any point  $(x, y)$  in the region of interest is a function of the vegetation, which is determined from an image segmentation of an overhead image of the region. The traversability costs in Table I were determined from initial values in [3] and [27] and then iteratively adjusted based on several representative segments, such that the predicted paths from our A\* simulation were similar to the actual paths taken by the Crusher autonomous mobility

TABLE I  
ITERATIVELY DERIVED TRAVERSABILITY COSTS FROM EMPIRICAL DATA FOR VEGETATION BASED ON  $k$ -MEANS IMAGE SEGMENTATION

$T_V$	VEGETATION
32	ROADS
32	DIRT
48	GREEN GRASS
96	THICK GRASS
96	DRY GRASS
192	CREEK/MUD
1000	FOLIAGE
2000	TREES

platform for the same segment. As an alternative to this empirically derived approach, we attempted to assign costs based on instantaneous velocity data; however, the data were too noisy to reliably assign costs on this basis. Improving the vegetation traversal costs,  $T_V$ , is the subject of future work

$$T_V(x, y) = f(\text{vegetation}; \text{image segmentation}). \quad (7)$$

2) *Elevation Costs:* In order to determine the components of the traversal cost based on the elevation data, we selected the national elevation dataset (NED) (April 2013 1/3-arc-second data) [46] digital elevation model (DEM), which consists of data from various sources including LT4X, Lidar, and airborne stereo imagery. The NED averages 1.55-m absolute elevation accuracy and 0.81-m relative elevation accuracy in the continental U.S. [47]. We also considered the SRTM DEM and ASTER Global DEM (GDEM), which are 1-arc-second datasets with nearly global coverage; however, a comparison by [47] showed SRTM and ASTER GDEM data to be less accurate than the NED data, and “the NED exhibits lower overall error than SRTM across the range of slope categories.” Lidar was also considered, but Lidar is not available in all areas and was not available for use in this paper. From the elevation,  $E$ , we compute the gradient and Hessian of the elevation (8). We then determine the magnitude of the gradient vector and use this value for the slope,  $M(x, y)$ , at each node location  $(x, y)$

$$\nabla E = \begin{pmatrix} \frac{\partial E}{\partial x} & \frac{\partial E}{\partial y} \end{pmatrix} \quad \nabla^2 E = \begin{pmatrix} \frac{\partial E}{\partial x^2} & \frac{\partial E}{\partial xy} \\ \frac{\partial E}{\partial yx} & \frac{\partial E}{\partial y^2} \end{pmatrix} \quad (8)$$

$$M(x, y) = \sqrt{\left(\frac{\partial E(x, y)}{\partial x}\right)^2 + \left(\frac{\partial E(x, y)}{\partial y}\right)^2}. \quad (9)$$

With a value for the slope at each point, we compute the component of the traversability cost based on regions of high slope,  $T_E$ , representing regions that are difficult to navigate, such as steep ravines, gullies, or cliffs. We found that a linear relationship most closely matched the performance of the robot on the test segments. The resulting equation for  $T_E(x, y)$  at each node location  $(x, y)$  was determined empirically and is given in (10). We applied a scaling factor,  $c_E$ , to (10) such that the order of magnitude of the slope values was similar to that of the vegetation values. We found empirically that when  $c_E = 400$ , the A\* simulated path and the actual robot path was most consistent; therefore, we set  $c_E = 400$  for (10) when conducting the subsequent linear regression analysis

$$T_E(x, y) = c_E M(x, y). \quad (10)$$

In contrast to regions of high slope, regions of zero slope also present navigational challenges for robots in the form of streams, washes, or muddy terrain. Consequently, these regions add a component to the traversability cost and are characterized by local minima in the slope, where the slope,  $M$ , is zero or nearly zero (in our case, when  $M < 0.2$ ). To determine these local minima, we apply the second partial derivative test over our region of interest [see (11)], where the spatial gradient equals zero and the spatial Hessian is positive definite (determinant is positive). Since this would result in discontinuous points, we apply traversal costs for all slope values,  $M$ , less than 0.2 according to (12), to determine  $T_Z(x, y)$  at each node location  $(x, y)$ , which is the zero-derivative component in (1). The component of the traversal cost representing these zero-slope regions,  $T_Z$ , was scaled in a manner similar to (10); therefore, we also set  $c_Z = 400$  in (12), resulting in a traversal cost,  $T_Z$ , which is inversely proportional to the slope and is at a maximum when the slope is zero

$$\nabla E = \mathbf{0} \text{ and } \det(\nabla^2 E) > 0 \quad (11)$$

$$T_Z(x, y) = c_Z(-5M(x, y) + 1). \quad (12)$$

*3) Developing Traversal Cost Map:* Once the traversal costs are calculated for each cell based on (7), (10), and (12), they are combined into a traversal cost map that represents the costs to traverse the environment. Using (1), we learn the values of the coefficients  $\lambda_0$ ,  $\lambda_V$ ,  $\lambda_E$ , and  $\lambda_Z$  based on a linear regression analysis. Then, using (1) explicitly for each cell  $(x, y)$ , the traversal costs for each cell within the map,  $T(x, y)$ , are computed from

$$T(x, y) = \lambda_0 + \lambda_V T_V(x, y) + \lambda_E T_E(x, y) + \lambda_Z T_Z(x, y). \quad (13)$$

### B. Environmental Complexity Measures

To characterize the complexity of the environment and predict system performance, we introduce two measures of environmental complexity to our model. The theoretical basis of the first measure is the observation that the complexity of the environment is a measure of the resistance a system encounters where it operates. Equation (14) represents this complexity ratio,  $C_{\text{ratio}}$ , which is the ratio of the distance travelled to the Euclidian distance between a start and end point. If one were to travel in a straight line with no deviation in the path, then  $C_{\text{ratio}}$  would equal 1.0, indicating no resistance

$$C_{\text{ratio}} = \frac{\sum_{n=1}^k \sqrt{(x_n - x_{n-1})^2 + (y_n - y_{n-1})^2}}{\sqrt{(x_k - x_0)^2 + (y_k - y_0)^2}}. \quad (14)$$

In (14), the numerator is the sum of the Euclidian distance between all adjacent nodes, which are the centroids of the cells from the discretization of the traversal cost map, along the path,  $(x_n, y_n)$  and  $(x_{n-1}, y_{n-1})$ , and the denominator is the Euclidian distance between the starting waypoint,  $(x_0, y_0)$ , and the goal waypoint,  $(x_k, y_k)$ .

The second measure recognizes that most global planners seek to find the minimum cost path through an environment, as exemplified by [3], but that while it seeks a minimum path, it is not without resistance. Thus, (15) sums up the accumulated

costs along the path as a measure of the resistance the UGV encounters along its path. Thus, the path complexity,  $C_{\text{path}}$ , is the total cost of the path, as calculated from the traversal cost map [see (15)]. Later, we find that normalizing the  $C_{\text{path}}$  metric serves as a highly effective metric for predicting system performance

$$C_{\text{path}} = \sum_{n=1}^k T(x_n, y_n). \quad (15)$$

In (15),  $T$  is the traversal cost of each node along the path and  $C_{\text{path}}$  is the sum of all,  $k$ , traversal costs along the path. Once a traversal cost map is created, the values for  $C_{\text{ratio}}$  and  $C_{\text{path}}$  are determined from (14) and (15), respectively, for each segment for which we have experimental data.

### C. Prediction

For a navigation mission, several performance metrics must be considered, including instantaneous velocity, average velocity, distance travelled, mission duration, volume and duration of communications to/from operator, and mission success rate [15]. We selected average velocity as the metric to predict because average velocity is often a key performance parameter for autonomous systems. We also sought to measure mission success rate, as measured by interventions; however, the number of interventions encountered by the actual system were insufficient to draw meaningful conclusions and will be pursued in future work.

To predict system performance between any two points, as summarized in Fig. 2, we run the A\* algorithm on the same traversal cost map described in the previous section. A\* is a specific case of a best-first search algorithm that determines the optimal, lowest cost path through a graph from a start node to a goal node [29]. Here we use A\* to find the minimum cost path between two points in our traversal cost map. We considered other state-of-the-art improvements on the original A\* algorithm, such as D\* [48] and other modified A\* algorithms [49]–[54], but since our goal is to predict system performance from a fully known map, as opposed to an onboard global planner with a partially observed map where D\* would be appropriate, we found that the original A\* approach by [29] was sufficient. As the A\* algorithm traverses the graph, it follows a path of the lowest expected total cost, keeping a sorted priority queue of alternate path segments along the way. Mathematically, the A\* algorithm seeks to minimize the value of  $f_n$ , denoted by

$$f_n = g_n + h_n. \quad (16)$$

In (16),  $f_n$  is the traversal cost at any node  $n$ , and consists of adding two parts,  $g_n$  and  $h_n$ , where  $g_n$  is the cost of the current optimal path from the start node,  $s$ , to the current node,  $n$ ; and  $h_n$  is an admissible estimate of the cost from the current node,  $n$ , to the goal [49]. Upon reaching the goal node,  $g$ ,  $f_n$  in (16) is at a minimum and the A\* algorithm finds the optimal minimum cost path through the graph. In our model, the costs represent the difficulty of traversing a cell. Equation (17) determines the cost,  $T_{1,2}$ , of traversing between any two adjacent nodes,  $(x_1, y_1)$  and  $(x_2, y_2)$ ,

where the Euclidian distance between nodes is multiplied by the one plus the average traversal cost values of the adjacent nodes

$$T_{1,2}((x_1, y_1), (x_2, y_2)) = \left[ 1 + \frac{T(x_2, y_2) + T(x_1, y_1)}{2} \right] \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}. \quad (17)$$

From the A\* algorithm, we compute the minimum cost path through the traversal cost map, and the associated complexity measures,  $C'_{\text{path}}$  and  $C'_{\text{ratio}}$  for each path segment. The  $C'_{\text{ratio}}$  and  $C'_{\text{path}}$  values are from the A\* simulated path and are used to estimate the system average velocity based on the learned model described in the previous section.

#### IV. RESULTS

We implemented our approach using data provided by DARPA from the UPI program to learn the parameters for our model, then used our model to successfully predict the actual performance of the Crusher autonomous navigation platform, and finally validated these results. The UPI program conducted numerous experiments from 2004 to 2008. The results presented here focus on UPI Experiment 10b, which was conducted at Ft. Drum, NY, in May–June 2008 (Fig. 4). As shown previously in Fig. 1, the Crusher is a 13 000-lb autonomous navigation platform developed by CMU-NREC [13] comprising the hardware platform and a software architecture (the perception, overhead map, and planning subsystems). The autonomy system hardware include six SICK LMS LADAR sensors (useful effective range of 30-m); a Velodyne HDL-64E LADAR (effective range of 100 m); four normalized difference vegetation index (NDVI) red and near infrared (NIR) cameras [ $70^\circ \times 52.5^\circ$  field of view (FOV),  $1024 \times 768$  resolution, 15-Hz frame rate], two covering the vehicle front and one on each side; and two stereo camera pairs looking to the vehicle front (52-cm and 1.2-m baselines, respectively,  $1024 \times 768$  resolution, 30-Hz frame rate, and  $43 \times 33$  FOV) [27]. Additionally, the Crusher autonomous navigation platform was provided with digital terrain elevation data (DTED) from the environment, ranging from DTED3 to DTED5, for *a priori* planning. For the results presented here, DTED4 data were used, which have approximately 3-m post spacing, along with 1-m RGB and NIR imagery. A 1-m resolution image means that each pixel represents 1-m  $\times$  1-m square on the ground [55]. The Crusher autonomy software implemented state-of-the-art autonomous navigation and perception algorithms, including online learning algorithms used to improve system performance [3].

The data used were from courses 1 and 1a of the Ft. Drum experiment, resulting in 26 independent segments. The metrics included global positioning system (GPS) tracks of the system for the runs, from which instantaneous and average velocities were determined. Interventions were also collected, but the data were insufficient to make meaningful predictions of interventions due to the limited number of interventions encountered. For the segments, the minimum waypoint spacing was 154 m and the maximum spacing was 842 m,



Fig. 4. Ft Drum, NY, DARPA UPI experiment 10b, course 1a. [28].



Fig. 5. Input overhead image along with actual path for the segment: course 1a, from waypoints 14 to 15 [28]. The axes are Cartesian coordinate axes in the model reference frame. North is up.



with an average spacing of 522 m between waypoints. The system achieved an average speed of 3.12 m/s; with a maximum speed of 6.42 m/s. Fig. 5 shows an input overhead image along with the actual path (blue) for an example segment on course 1a from waypoints 14 to 15. Fig. 6 shows the learned traversal cost map for the same segment as shown in Fig. 5 with the actual path (white) and predicted path (green).

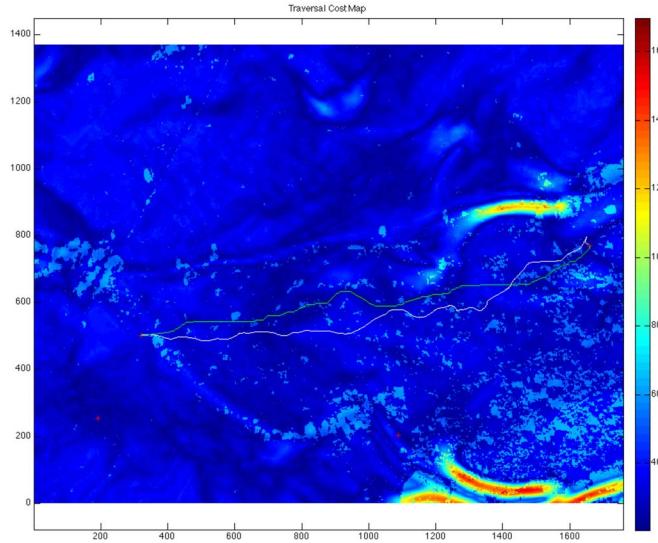


Fig. 6. Final traversal cost map from (13) for the region shown in Fig. 5, where the predicted path (green) is compared with the actual experimental path (white). Coefficients:  $\lambda_V = 0.2748$ ,  $\lambda_E = 2.4337$ , and  $\lambda_Z = 0.8805$ . The axes are Cartesian coordinate axes in the model reference frame. North is up. The heat scale represents regions of higher traversal cost.

#### A. Traversal Cost Map

The following two sections detail how we developed the traversal cost map contributions from a combination of the vegetation and elevation costs.

**1) The Vegetation Costs:** We found that importing an image from an elevation of 2 km provided a reasonably sized image with sufficient vegetative fidelity without exceeding the practical processing limitations of our system (2.8-GHz Core i7 with 16-GB memory). Additionally, when calculating the predicted paths using our implementation of A\*, we found that a discretization of the map not exceeding 1.5E+06 cells for each segment performed efficiently. Once we imported the image from Google Earth, we performed an image segmentation using *k*-means. We found that  $k = 6$  clusters were sufficient to adequately segment the image into identifiable vegetative regions, which was consistent with number of clusters used by [3] in their GMM. Squared Euclidian distance was used as the distance measure, and four independent *k*-means segmentations were performed with varied values for the coefficients,  $\phi_L$ ,  $\phi_a$ ,  $\phi_b$ ,  $\phi_{eL}$ ,  $\phi_{ea}$ , and  $\phi_{eb}$ . We sought to only use the  $a^*$  and  $b^*$  channel images and entropy, since those channels contained the color information in the image, as suggested by many, including [3]. Further, we found that five replicates were sufficient to ensure that we did not converge on a local optimum, with a maximum of 200 iterations [56]. The segmentation usually converged with only three replicates. Fig. 7 shows an example of the results from the *k*-means image segmentation.

Fig. 8 shows the same six clusters independently from the image segmentation in Fig. 7. The clusters are labeled per terrain type and  $T_V$  values from Table I are assigned to the clusters. For example, cluster 2 in Fig. 8 is classified as “roads” and assigned “32”; similarly, cluster 4 is classified as “foliage” and assigned “1000.” Fig. 9 is the resultant vegetation component,  $\lambda_V T_V$ , of the traversal cost map.

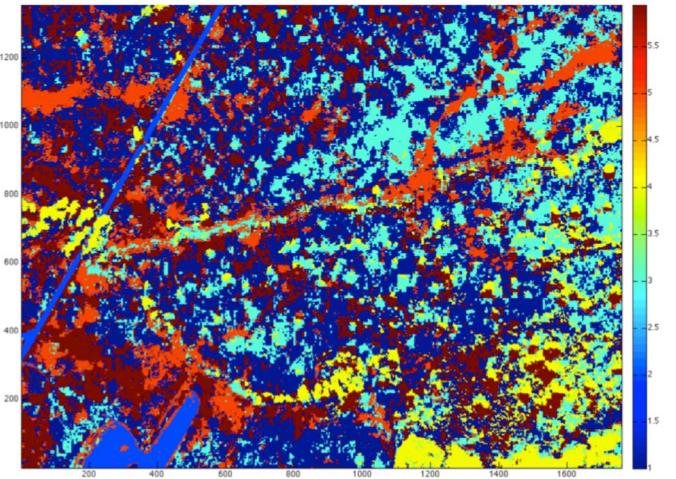


Fig. 7. Example result of *k*-means image segmentation for course 1a, waypoint 14 to 15. The *k*-means coefficients for this example are  $\phi_L = 0$ ,  $\phi_a = 1$ ,  $\phi_b = 1$ ,  $\phi_{eL} = 0$ ,  $\phi_{ea} = 0.1$ , and  $\phi_{eb} = 0.1$ . The axes are Cartesian coordinate axes in the model reference frame. North is up.

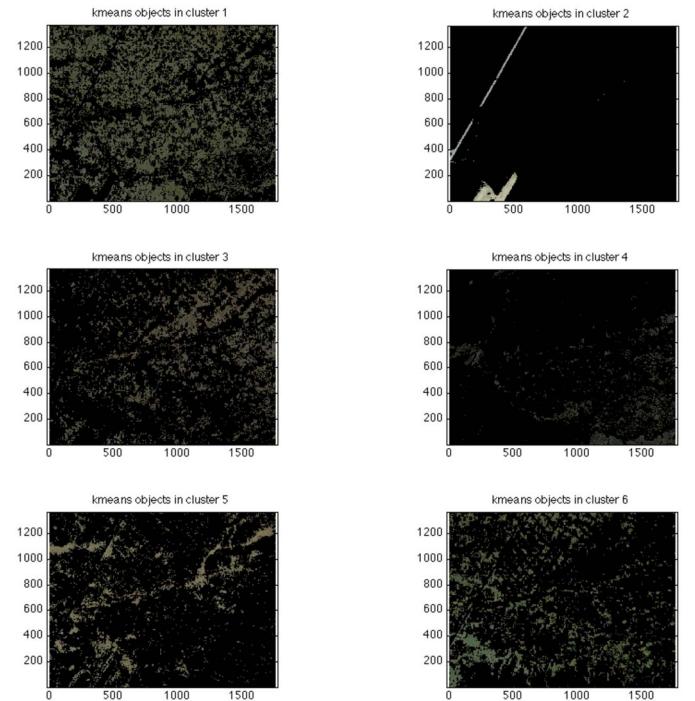


Fig. 8. Classification of regions based on image segmentation, from Fig. 7. Again, the *k*-means coefficients for this example are  $\phi_L = 0$ ,  $\phi_a = 1$ ,  $\phi_b = 1$ ,  $\phi_{eL} = 0$ ,  $\phi_{ea} = 0.1$ , and  $\phi_{eb} = 0.1$ . The axes are Cartesian coordinate axes in the model reference frame. North is up.

**2) Elevation Costs:** We acquired elevation data for the regions of interest [46] and computed the gradients from (8). The costs shown in Fig. 10,  $\lambda_E T_E$ , are the component of the traversal cost based on the high-slope regions from (10) and the learned coefficient. Similarly, Fig. 11 shows  $\lambda_Z T_Z$ , the component of the traversal cost based on the regions of near-zero slope, which represent areas of potential swamp and streams from (12) and the learned coefficient. Figs. 9–11 are then combined to get the final traversal cost map, shown previously in Fig. 6. Fig. 12 shows additional examples of the

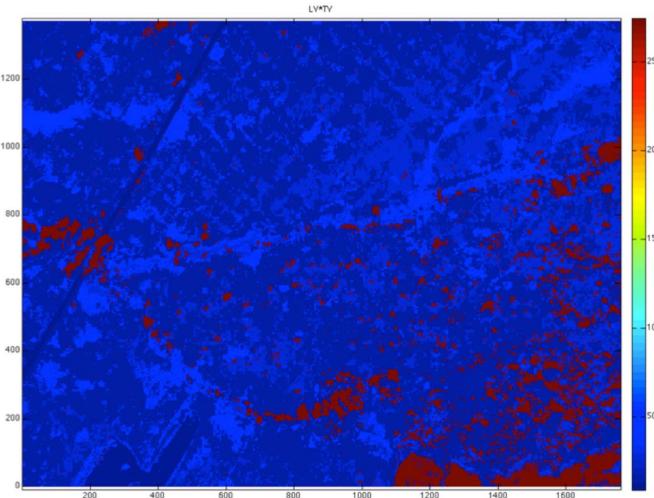


Fig. 9. Resulting vegetation component,  $\lambda_V T_V$ , of traversal cost map (Fig. 6) for segment: course 1a, waypoints 14 to 15, coefficients:  $\lambda_V = 0.2748$ . The axes are Cartesian coordinate axes in the model reference frame. North is up. The heat scale represents regions of higher traversal cost.

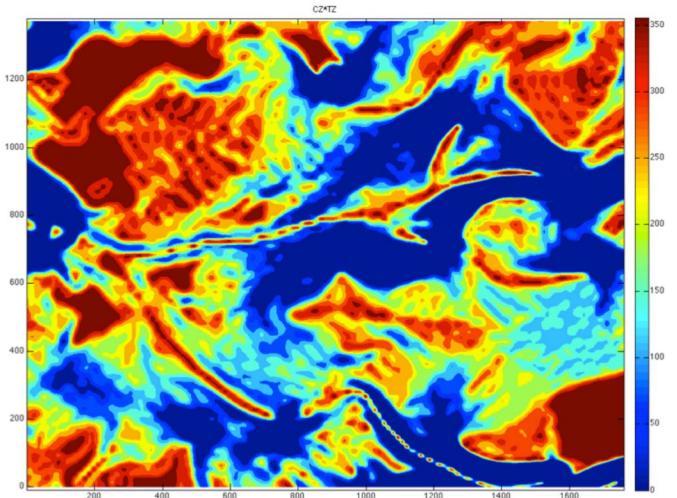


Fig. 11. Resulting zero-slope component,  $\lambda_Z T_Z$ , of traversal cost map (Fig. 6) for segment: course 1a, waypoints 14 to 15, coefficients:  $\lambda_Z = 0.8805$ . The axes are Cartesian coordinate axes in the model reference frame. North is up. The heat scale represents regions of higher traversal cost.

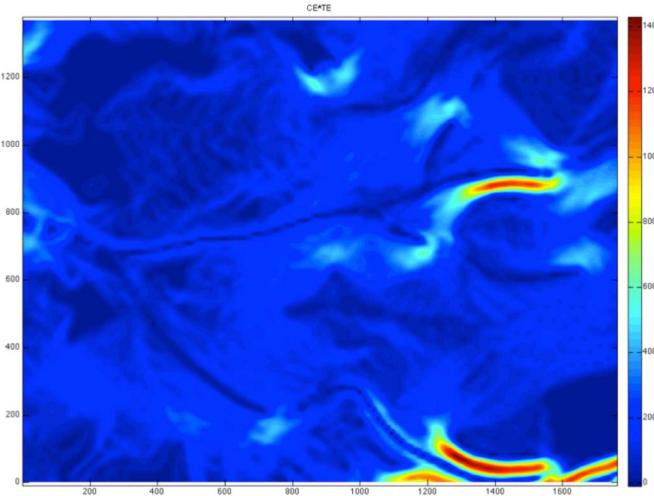


Fig. 10. Resulting high-slope component,  $\lambda_E T_E$ , of traversal cost map (Fig. 6) for segment: Course 1a, waypoints 14 to 15, coefficients:  $\lambda_E = 2.4337$ . The axes are Cartesian coordinate axes in the model reference frame. North is up. The heat scale represents regions of higher traversal cost.

model's predicted path compared with the actual experimental results. Fig. 13 depicts segments that show how our predicted path compares with actual runs that experienced difficulty in the terrain, resulting in extensive replanning, likely due to swampy and muddy conditions.

### B. Model Results

Our multivariate linear regression analysis produced the following values for the coefficients in (1):  $\lambda_V = 0.2748$ ,  $\lambda_E = 2.4337$ ,  $\lambda_Z = 0.8805$ , and  $\lambda_0 = 4.4296$ . These parameters resulted in a linear fit of the data with an  $R^2$  value of 0.8796. We then summed the contributions from the three traversal cost components from (13) to determine the total traversal cost map and the associated learned model (see Fig. 14). Based on this fit and associated residual analysis, we were satisfied that our linear model sufficiently modeled system performance.

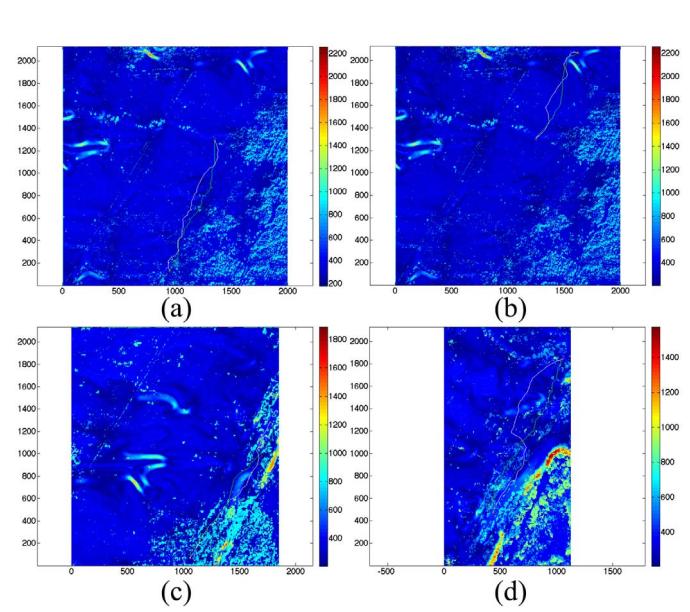


Fig. 12. Additional example final traversal cost maps for selected segments from (13), the predicted path (green) is compared with the actual experimental path (white). Coefficients:  $\lambda_V = 0.2748$ ,  $\lambda_E = 2.4337$ , and  $\lambda_Z = 0.8805$ . (The axes are Cartesian coordinate axes in the model reference frame. North is up. The heat scale represents regions of higher traversal cost). (a) Segment: course 1a, waypoints 9 to 10. (b) Segment: course 1a, waypoints 2 to 3. (c) Segment: course 1, waypoints 6 to 7. (d) Segment: course 1a, waypoints 12 to 13.

We found that the best model for predicting system performance was the normalized  $C_{\text{path}}$  versus system average velocity. In contrast, the relationship between  $C_{\text{ratio}}$  and predicting system performance (average system velocity) was less informative. We found a strong linear relationship between the  $C_{\text{ratio}}$  value and system performance; however, it yielded less than satisfactory predictive results. For example, our prediction results produced mean squared error (MSE) values of 49%. Consequently, the  $C_{\text{ratio}}$  value was not considered further.

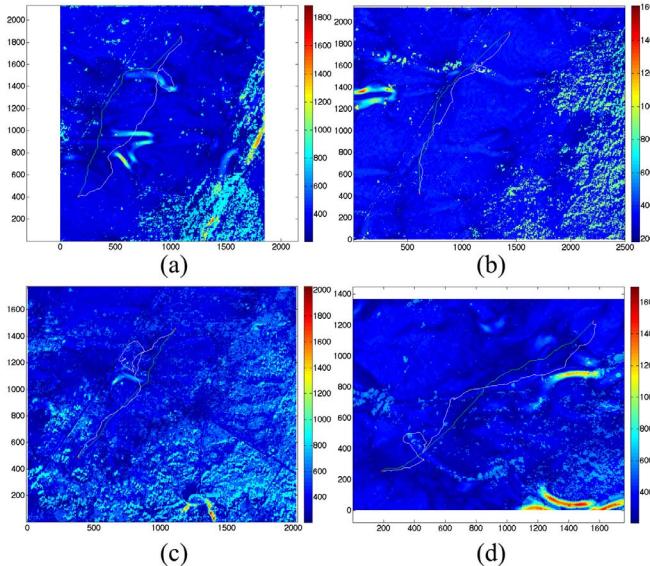


Fig. 13. Additional example final traversal cost maps for selected segments from (13), the predicted path (green) is compared with the actual experimental path (white) for experimental runs that experienced difficulty in navigating the terrain, resulting in extensive replanning, likely due to swampy and muddy conditions. Coefficients:  $\lambda_V = 0.2748$ ,  $\lambda_E = 2.4337$ , and  $\lambda_Z = 0.8805$ . (The axes are Cartesian coordinate axes in the model reference frame. North is up. The heat scale represents regions of higher traversal cost.) (a) Segment: course 1, waypoints 0 to 1. (b) Segment: course 1, waypoints 1 to 2. (c) Segment: course 1, waypoints 2 to 3. (d) Segment: course 1, waypoints 9 to 10.

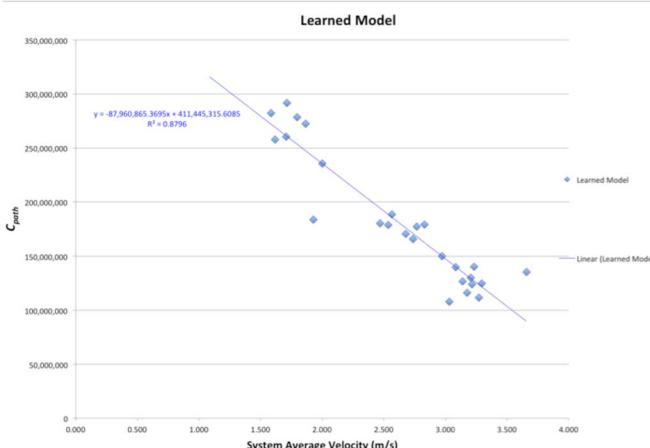


Fig. 14. Learned model of the system based on the coefficients learned from the multivariate linear regression analysis, (1):  $\lambda_V = 0.2748$ ,  $\lambda_E = 2.4337$  and  $\lambda_Z = 0.8805$ , and  $\lambda_0 = 4.4296$ .

### C. Prediction Results

In order to predict system performance, as shown in our approach summary in Fig. 2, we ran  $A^*$  on 26 segments using the learned traversal cost map to determine a value for  $C'_\text{path}$  for each segment. We then used this value of  $C'_\text{path}$  to estimate system performance. We found that our model resulted in prediction accuracy for system performance with an average MSE of 8.22%. Table II shows the results of this model, where the learned model column combined with the actual speed column gives us our equation of the model. For each

TABLE II  
SYSTEM PREDICTION

SEGMENT	LEARNED MODEL ( $C_{\text{path}}$ )	$A^*$ PREDICTION ( $C'_\text{path}$ )	$A^*$ WITH OFFSET ( $C''_\text{path}$ )	ACTUAL SPEED (m/s)	PREDICTED SPEED (m/s)	PERCENT ERROR	MEAN SQUARED ERROR
1P 01	178919274	150655563	173253898	2.831	2.708	4.34%	1.51%
1P 12	135407530	126617027	145609581	3.657	3.022	17.35%	40.24%
1P 23	165977891	156668983	180169330	2.736	2.629	3.89%	1.13%
1P 34	140037308	136439876	156905858	3.085	2.894	6.19%	3.65%
1P 45	272542309	261601553	300841786	1.866	1.257	32.61%	37.01%
1P 56	125000279	98381128	113138297	3.295	3.391	-2.93%	0.93%
1P 67	130316996	114847867	132075047	3.205	3.176	0.90%	0.08%
1P 78	278504326	251044659	288701358	1.796	1.395	22.32%	16.07%
1P 89	116334493	107635637	123780983	3.175	3.270	-3.01%	0.91%
1P 90	178894894	159260495	183149569	2.534	2.595	-2.41%	0.37%
1P 1011	126870781	114005504	131106330	3.141	3.187	-1.48%	0.22%
1P 1112	140141230	98708460	113514729	3.231	3.387	-4.82%	2.43%
1P 1213	149989195	122697605	141102246	2.972	3.073	-3.43%	1.04%
1P 1314	111601384	101794973	117064219	3.273	3.347	-2.25%	0.54%
1P 1415	107776811	92023958	105827552	3.031	3.474	-14.64%	19.68%
1P 1516	124018953	123961685	142555938	3.215	3.057	4.90%	2.48%
C1 01	235693805	184650720	212348328	2.002	2.263	-13.06%	6.84%
C1 12	291492005	230435536	265000866	1.714	1.665	2.86%	0.24%
C1 23	282075399	247508704	284635010	1.586	1.442	9.09%	2.08%
C1 34	170409775	132077176	151888752	2.676	2.951	-10.26%	7.53%
C1 45	257750787	225517120	259344688	1.618	1.729	-6.86%	1.23%
C1 56	260288894	217263936	249853526	1.707	1.837	-7.62%	1.69%
C1 67	177234092	129509144	148935516	2.768	2.984	-7.81%	4.68%
C1 78	180165744	147527504	169656630	2.469	2.749	-11.31%	7.80%
C1 89	188528769	151886592	174669581	2.566	2.692	-4.91%	1.59%
C1 910	183737362	155358240	17861976	1.927	2.646	-37.32%	51.72%
						AVE % ERROR	AVERAGE MSE
						-1.14%	8.22%

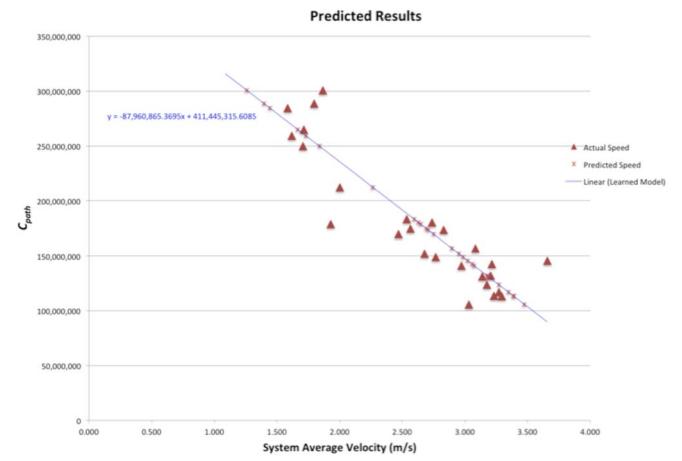


Fig. 15. Actual results from the values in Table II, compared with the learned linear model of the system from (1) and shown in Fig. 14.

segment, the calculated  $C'_\text{path}$  values from  $A^*$  are shown in the  $A^*$  prediction column.

In order to achieve this level of accuracy, we applied an offset to the cost values since  $A^*$  is an optimal path and will always underestimate the cost. We found that the  $C'_\text{path}$  values from the  $A^*$  predicted path were lower than the  $C'_\text{path}$  values from the empirical data, as anticipated. We learned that this bias was 15% lower than the learned model values. We deduced the value of this offset by minimizing the average percent error in the linear fit of the learned model equation and the  $A^*$  predicted equation. With an offset of 15% applied, we achieved an average MSE of 8.22% for the system's predicted average velocity versus the actual velocity. This value is shown in the  $A^*$  prediction with offset column of Table II. From these values, we compute the predicted average velocity (predicted speed), also shown in Table II.

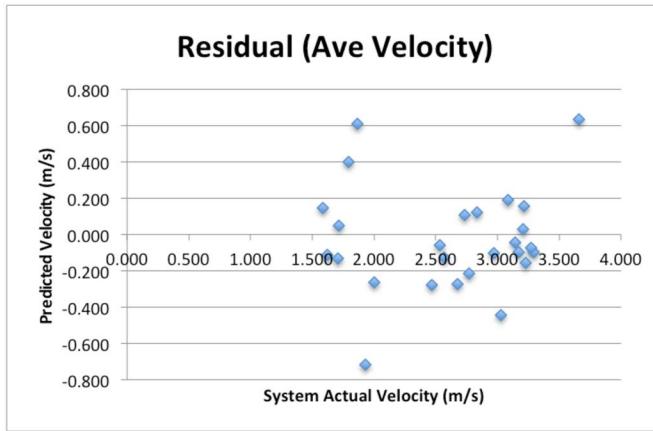


Fig. 16. Residual analysis to validate the linear regression model of the predicted average system velocity.

The predicted results (with the applied offset of 15%) from Table II are plotted in Fig. 15 and compared with the same learned linear model from Fig. 14. However, even if an offset had not been applied, we achieved an average MSE of 15.70%. These values are for a model that trained on all 26 segments and made 26 predictions based on A\* estimates for all 26 segments. In the next section, we perform a cross-validation to demonstrate the accuracy of the model with test segments removed from the model. Examination of the results in Table II reveal that for most of the segments, our prediction of speed for each segment was within 5% MSE of the actual average speed for 18 of the 26 segments and within 8% MSE for 21 of the 26 segments. Analysis of the data for the segments that exceeded 8% error, we believe, was due to our model estimate in regions of zero-slope, characterized by muddy and swampy conditions. The impact of the zero-slope regions is important, because 89% of interventions were due to swamp-like regions [28]. This is an area that we are investigating to improve our model since the effect of mud on the system was not temporally consistent (e.g., sunny versus rainy during the testing). Analysis of weather data in advance of the experimental runs revealed heavy showers within 24–48 h. As the course dried out, the effects changed. Future work will include a sensitivity analysis of the model components and their effects on system performance predictions; a linear model may be insufficient to model these effects.

Fig. 16 shows the residual analysis of the predicted average system velocity. Note that the residual analysis is randomly distributed about the horizontal axis, thus validating our linear regression model.

## V. VALIDATION

To validate our model, we performed a leave-one-out cross-validation (LOOCV) based on the fact that we had a limited number of segments with which to build our model. Table III shows the results of our LOOCV. The first three columns show the learned model based on 25 training segments (1 segment was removed for testing). The remaining columns show the test results of the prediction with the segment that was held out when training the model. The average MSE of the LOOCV

TABLE III  
LOOCV

LEAVE OUT	TRAINING		TEST					
	SLOPE	Y-INTERCEPT	TEST	PREDICTED ADJUSTED ( $C_{PANi}$ )	ACTUAL SPEED (M/S)	PREDICTED SPEED (M/S)	PERCENT ERROR (SPEED)	MEAN SQUARED ERROR (SPEED)
1	-88324102	411734585	1	173253898	2.831	2.700	4.62%	1.71%
2	.93448651	423766857	2	145609581	3.657	2.977	18.60%	46.24%
3	-87883124	411440628	3	180169330	2.736	2.632	3.81%	1.09%
4	-87938598	411395617	4	156905858	3.085	2.894	6.19%	3.64%
5	-85858647	404874759	5	300841786	1.866	1.212	35.06%	42.79%
6	-88188764	411906653	6	113138297	3.295	3.388	-2.82%	0.86%
7	-87990058	411495855	7	132075047	3.205	3.176	0.91%	0.09%
8	-85654658	40435869	8	288701358	1.796	1.350	24.85%	19.92%
9	-87000114	409590807	9	123780983	3.175	3.285	-3.48%	1.22%
10	-88025937	412008454	10	183149569	2.534	2.600	-2.59%	0.43%
11	-87480316	410534953	11	131106330	3.141	3.194	-1.71%	0.29%
12	-88789709	413085509	12	113514729	3.231	3.374	-4.42%	2.04%
13	-87939917	411398458	13	141102246	2.972	3.074	-3.44%	1.04%
14	-87095040	409685151	14	117064219	3.273	3.360	-2.65%	0.75%
15	-86333134	408698888	15	105827552	3.031	3.508	-15.75%	22.79%
16	-87644233	410815356	16	142555938	3.215	3.061	4.78%	2.36%
17	-87918419	411325525	17	212348328	2.002	2.263	-13.05%	6.83%
18	-84797085	401820896	18	265000866	1.714	1.613	5.85%	1.01%
19	-86725645	407760921	19	284635010	1.586	1.420	10.48%	2.76%
20	-87908002	411539353	20	151888752	2.676	2.954	-10.36%	7.69%
21	-89250843	415335680	21	259344688	1.618	1.748	-8.01%	1.68%
22	-88042465	411713249	22	249853526	1.707	1.838	-7.70%	1.73%
23	-88085176	411405716	23	148935516	2.768	2.980	-7.64%	4.48%
24	-88159802	412537787	24	169656630	2.469	2.755	-11.56%	8.15%
25	-87926681	411250335	25	174669581	2.566	2.691	-4.87%	1.56%
26	-92313306	425296232	26	178661976	1.927	2.672	-38.63%	55.42%
					AVE %	AVERAGE		
					ERROR	MSE		
					-0.90%	9.18%		

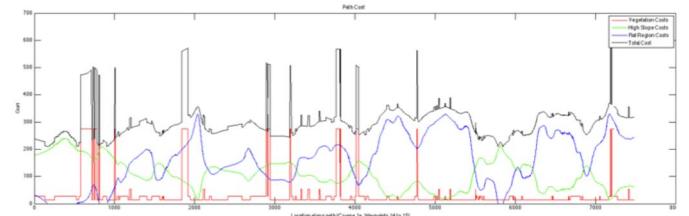


Fig. 17. Two-dimensional graph showing the traversal costs from (13) along the actual path for a selected segment: course 1a, from waypoints 14 to 15 (shown in Fig. 5). The x-axis represents the location along the path from waypoint 14 to 15; the y-axis represents the costs for each cell along the path. The total cost at each location along the path (black); the vegetative costs (red); the high slope costs (green); and the flat region costs (blue).

is 9.18%. This is less than 1% worse than the above results, indicating that the model is robust.

## VI. CONCLUSION

We have developed a quantitative model for predicting the performance of nondeterministic, autonomous UGV systems based on the complexity of the environment from a systems engineering perspective. We validated our hypothesis by modeling the complexity of the environment in which a system will operate and found that we can predict the system's performance. We learned this model using experimental data from an autonomous mobile platform performing a navigation mission in complex off-road terrain. The model characterizes system performance based on a complexity measure that is the function of a traversability cost map. Additionally, we introduced a complexity measure to predict system performance in similar environments. Our model demonstrates the ability to predict system performance based on environmental complexity as characterized by vegetation and elevation change, but the framework can be extended to include additional terms that represent other forms of environmental complexity. Further,

the framework is valid for use by other UGVs, though the model parameters would have to be learned based on the UGV's performance in its environment.

Our model has many uses beyond predicting system performance based on environmental complexity. It can be used to assess improvements in system performance as algorithms and behaviors that are developed by researchers in similar environments. For example, as algorithms improve, one would expect system performance (average velocity) to improve as a function of fixed complexity. Additionally, such a model and extensions may prove useful to the TEV&V communities within the government to assess the performance of autonomous, nondeterministic systems, given that it is intractable to exhaustively test in all possible environments under all conditions. As such, this framework can be used as a platform for testing. This is especially relevant as the government, and military in particular, embark on the fielding of autonomous military robot systems. Another utility of the model lies in its ability to predict system performance in previously unvisited and untested environments based on an analysis of the environmental complexity. This may prove tremendously useful when planning anticipated system performance and logistics requirements, whether for fielded systems or experimentation. Further, the statistical approach in this paper could be combined with many of the widely used physics-based models to create a robust tool for use in TEV&V.

This paper presented here extends the previous work that has characterized complexity as a binary function of local obstacles by providing a continuous function that is tightly correlated to system performance. The model framework can be expanded to consider other environmental factors and combined with predictions in other varied environments to get a broader description of a system's performance across an array of operational environments to build a database for a system that can be used by the TEV&V community to build a better understanding of how nondeterministic systems will perform. Further, the model could eventually be expanded to consider system performance in dynamic environments by simulating dynamic events to update the traversal cost map, in a manner as suggested by [6] and [57], and then by replacing A\* with a dynamic search algorithm, such as D\* [48], for predicting system performance.

Future work includes assessing the model of the system in other classes of environments and for different robotic systems. For example, the model will be run on data for a UPI experiment held in the desert environment of Camp Roberts, CA. Future work will also include improving the  $T_E$  values, which currently use a learned lookup table to determine the values. Such work will include learning cost values based on instantaneous system velocities. Work has begun in this area, but the data from Ft. Drum presented in this paper proved too noisy to extract accurate velocities as a function of instantaneous terrain. Other future work will explore the  $C_{ratio}$  and other measures as better predictors of performance or used in combination with the  $C_{path}$  measure. This will prove a challenge because changing the values in the traversal cost map for trouble areas causes the A\* algorithm, which is seeking the optimal path, to

deviate significantly from the path taken by the actual system. However, this deviation itself may provide additional predictive insight.

Additionally, current and future work will explore the impact of the environment on subsystems rather than the system as a whole. While subsystem data were not available for the work presented here, investigating the environmental impact on subsystems may reveal additional insight into a systems performance. The investigation of the subsystem performance may also benefit from a decomposition of the environmental complexity costs shown in Fig. 17. In Fig. 17 the black line in the graph shows the total cost for each point along a representative path (Course 1a, from Waypoints 14 to 15). The total cost (black) is composed of the vegetative costs (red), the costs from high slope regions (green), and the costs from flat regions (blue). For example, the environmental impact on the perception subsystem may be investigated by comparing classifier results versus ground truth as a function of the environment measured using the overall complexity costs and its components (Fig. 17). Additionally, the impact of the environment to the planning subsystem and other performance measures may be examined by comparing the actual path versus the predicted path as a function of the complexity costs. Further, the environment-based analysis could be applied to other UGV behaviors beyond navigation including exploration, reconnaissance, surveillance, and many others. Understanding the impact of the environment on other behaviors, and the ability to predict system performance will certainly prove a useful tool for use in TEV&V and autonomous system development.

## ACKNOWLEDGMENT

The authors would like to thank E. Krotkov, M. Perschbacher, and J. Pippine for their support in providing the data analyzed in this paper. They would also like to thank the reviewers for their thorough reviews and their suggestions that have contributed to the quality of the publication.

## REFERENCES

- [1] R. Murphy and J. Shields, "Defense science board-task force report—The role of autonomy in DOD systems," Office Under Secretary Defense Acquisition, Technol. Logist., Washington, DC, USA, Tech. Rep. CD 1172, 2012.
- [2] D. J. Silver, A. Bagnell, and A. Stentz, "Learning from demonstration for autonomous navigation in complex unstructured terrain," *Int. J. Robot. Res.*, vol. 29, no. 12, pp. 1565–1592, 2010.
- [3] B. Sofman *et al.*, "Improving robot navigation through self-supervised online learning," *J. Field Robot.*, vol. 23, no. 12, pp. 1059–1075, 2006.
- [4] J. Peters and A. Y. Ng, "Guest editorial: Special issue on robot learning—Part A," *Auton. Robots*, vol. 27, no. 1, pp. 1–2, 2009.
- [5] D. Silver, J. Bagnell, and A. Stentz, "High performance outdoor navigation from overhead data using imitation learning," in *Robotics: Science and Systems IV*. Cambridge, MA, USA: MIT Press, 2008.
- [6] J. A. Bagnell, D. Bradley, D. Silver, B. Sofman, and A. Stentz, "Learning for autonomous navigation," *IEEE Robot. Autom. Mag.*, vol. 17, no. 2, pp. 74–84, Jun. 2010.
- [7] T. Kollar and N. Roy, "Trajectory optimization using reinforcement learning for map exploration," *Int. J. Robot. Res.*, vol. 27, no. 2, pp. 175–196, 2008.
- [8] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.
- [9] E. Krotkov *et al.*, "The DARPA PerceptOR evaluation experiments," *Auton. Robots*, vol. 22, no. 1, pp. 19–35, 2007.
- [10] L. D. Jackel, E. Krotkov, M. Perschbacher, J. Pippine, and C. Sullivan, "The DARPA LAGR program: Goals, challenges, methodology, and phase I results," *J. Field Robot.*, vol. 23, nos. 11–12, pp. 945–973, 2006.

- [11] J. Pippine, D. Hackett, and A. Watson, "An overview of the defense advanced research projects agency's learning locomotion program," *Int. J. Robot. Res.*, vol. 30, no. 2, pp. 141–144, 2011.
- [12] B. A. Boddy and R. Camden, "Technology readiness level six and autonomous mobility," in *Proc. Int. Soc. Optics Photon., Defense Secur.*, Orlando, FL, USA, 2004, pp. 302–313.
- [13] CMU NREC. *National Robotics Unmanned Vehicle Design Projects*. Accessed on Aug. 8, 2014. [Online]. Available: <http://www.nrec.ri.cmu.edu/projects/unmanned/>
- [14] C. N. Calvano and P. John, "Systems engineering in an age of complexity," *Syst. Eng.*, vol. 7, no. 1, pp. 25–34, 2004.
- [15] A. Lampe and R. Chatila, "Performance measure for the evaluation of mobile robot autonomy," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Orlando, FL, USA, 2006, pp. 4057–4062.
- [16] J. W. Crandall, M. A. Goodrich, D. R. Olsen, Jr., and C. W. Nielsen, "Validating human–robot interaction schemes in multitasking environments," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 35, no. 4, pp. 438–449, Jul. 2005.
- [17] P. Papadakis, "Terrain traversability analysis methods for unmanned ground vehicles: A survey," *Eng. Appl. Artif. Intell.*, vol. 26, no. 4, pp. 1373–1385, 2013.
- [18] H.-M. Huang, E. Messina, A. Jacoff, R. Wade, and M. McNair, "Performance measures framework for unmanned systems (PerMFUS): Models for contextual metrics," in *Proc. ACM 10th Perform. Metrics Intell. Syst. Workshop*, New York, NY, USA, 2010, pp. 22–28.
- [19] J. W. Crandall, "Towards developing effective human–robot systems," M.S. thesis, Dept. Comput. Sci., Brigham Young Univ., Provo, UT, USA, 2003.
- [20] G. Lidoris, F. Rohrmüller, D. Wollherr, and M. Buss, "System interdependence analysis for autonomous robots," *Int. J. Robot. Res.*, vol. 30, no. 5, pp. 601–614, 2011.
- [21] F. Rohrmüller, G. Lidoris, D. Wollherr, and M. Buss, "System interdependence analysis for autonomous mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, St Louis, MO, USA, 2009, pp. 696–701.
- [22] G. T. Anderson and G. Yang, "A proposed measure of environmental complexity for robotic applications," in *Proc. IEEE Int. Conf. Man, Syst., Cybern. (SMC)*, Montreal, QC, Canada, 2007, pp. 2461–2466.
- [23] G. Yang and G. T. Anderson, "An experimental study of environmental complexity as seen by robots," in *Proc. IEEE Int. Conf. Man, Syst., Cybern. (SMC)*, Anchorage, AK, USA, 2011, pp. 3102–3106.
- [24] J. Held, A. Lampe, and R. Chatila, "Linking mobile robot performances with the environment using system maps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Beijing, China, 2006, pp. 651–656.
- [25] IEEE Standards Coordinating Committee, *IEEE Standard Glossary of Software Engineering Terminology*, IEEE Standard 610.12-1990, 1990.
- [26] "Systems engineering vision 2020," Int. Council Syst. Eng., San Diego, CA, USA, Tech. Rep. INCOSE-TP-2004-004-02, 2007.
- [27] A. Stentz, "DARPA UPI project," CMU-NREC, Pittsburgh, PA, USA, Tech. Rep., 2008.
- [28] M. Perschbacher and J. Bares, "UPI Ft. Drum, New York—June 2008," CMU-NREC, Pittsburgh, PA, USA, Tech. Rep., 2008.
- [29] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, Jul. 1968.
- [30] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
- [31] S. Chatterjee and A. S. Hadi, *Regression Analysis by Example*. Hoboken, NJ, USA: Wiley, 2012.
- [32] Google Earth. (Oct. 11, 2013). *Ft. Drum, NY. Lat 44.14027, Lon -75.64228. Map Image: Google, Digital Globe. Eye alt 2.00km*. Accessed on May 4, 2014.
- [33] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979.
- [34] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Stat. Probab.*, vol. 1. Los Angeles, CA, USA, 1967, pp. 281–297.
- [35] A. Z. Chitade and S. K. Katiyar, "Colour based image segmentation using k-means clustering," *Int. J. Eng. Sci. Technol.*, vol. 2, no. 10, pp. 5319–5325, 2010.
- [36] C. Chandhok and M. T. Scholar, "Color image segmentation using k-means clustering," *Int. J. VLSI Digit. Signal Process. Appl.*, vol. 2, no. 3, pp. 241–245, 2012.
- [37] D. Silver, B. Sofman, N. Vandapel, J. A. Bagnell, and A. Stentz, "Experimental analysis of overhead data processing to support long range navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Beijing, China, 2006, pp. 2443–2450.
- [38] H. Permuter, J. Francos, and I. Jermyn, "A study of Gaussian mixture models of color and texture features for image classification and segmentation," *Pattern Recognit.*, vol. 39, no. 4, pp. 695–706, 2006.
- [39] M. F. Augusteijn, L. E. Clemens, and K. A. Shaw, "Performance evaluation of texture measures for ground cover identification in satellite images by means of a neural network classifier," *IEEE Trans. Geosci. Remote Sens.*, vol. 33, no. 3, pp. 616–626, May 1995.
- [40] H. Mobahi, S. R. Rao, A. Y. Yang, S. S. Sastry, and Y. Ma, "Segmentation of natural images by texture and boundary compression," *Int. J. Comput. Vis.*, vol. 95, no. 1, pp. 86–98, 2011.
- [41] M. S. Jigar, S. Brijesh, and S. K. Satis, "A new K-mean color image segmentation with cosine distance for satellite images," *Int. J. Eng. Adv. Technol.*, vol. 1, no. 5, pp. 27–30, 2012.
- [42] L.-K. Soh and C. Tsatsoulis, "Segmentation of satellite imagery of natural scenes using data mining," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 2, pp. 1086–1099, Mar. 1999.
- [43] A. R. Robertson, "The CIE 1976 color-difference formulae," *Color Res. Appl.*, vol. 2, no. 1, pp. 7–11, 1977.
- [44] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2008.
- [45] Mathworks. *Entropy of Grayscale Image*. Accessed on Aug. 6, 2014. [Online]. Available: <http://www.mathworks.com/help/images/ref/entropy.html>
- [46] US Geological Survey (USGS). *The National Map: The US Geological Survey (USGS)*. Accessed on Jan. 22, 2014. [Online]. Available: <http://nationalmap.gov>
- [47] D. B. Gesch, M. J. Oimoen, and G. A. Evans, "Accuracy assessment of the U.S. geological survey national elevation dataset, and comparison with other large-area elevation datasets—SRTM and ASTER," U.S. Dept. Interior, U.S. Geol. Survey, Reston, VA, USA, Tech. Rep. 2014–1008, 2014.
- [48] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, San Diego, CA, USA, 1994, pp. 3310–3317.
- [49] J. Pearl, *Heuristics*. Reading, MA, USA: Addison-Wesley, 1984, pp. 27–30.
- [50] A. Nash, S. Koenig, and M. Likhachev, "Incremental Phi\*: Incremental any-angle path planning on grids," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, San Francisco, CA, USA, 2009, pp. 1824–1830.
- [51] A. Nash and S. Koenig, "Any-angle path planning," *Artif. Intell. Mag.*, vol. 34, no. 4, pp. 85–107, 2013.
- [52] K. Daniel, A. Nash, S. Koenig, and A. Felner, "Theta\*: Any-angle path planning on grids," *J. Artif. Intell. Res.*, vol. 39, pp. 533–579, Oct. 2010.
- [53] F. Duchoň *et al.*, "Path planning with modified a star algorithm for a mobile robot," *Proc. Eng.*, vol. 96, pp. 59–69, 2014, doi: 10.1016/j.proeng.2014.12.098.
- [54] D. Ferguson and A. Stentz, "Field D\*: An interpolation-based path planner and replanner," in *Robotics Research*. Heidelberg, Germany: Springer, 2005, pp. 239–253. [Online]. Available: [https://www.ri.cmu.edu/publication\\_view.html?pub\\_id=5122](https://www.ri.cmu.edu/publication_view.html?pub_id=5122)
- [55] USDA. *Why is the CLU Digitized on 1-Meter Imagery?* Accessed on Jul. 12, 2015. [Online]. Available: [https://www.fsa.usda.gov/Internet/FSA\\_File/clu\\_1-meter\\_imagery.pdf](https://www.fsa.usda.gov/Internet/FSA_File/clu_1-meter_imagery.pdf)
- [56] Mathworks. *k-Means Clustering*. Accessed on Jul. 13, 2015. [Online]. Available: <http://www.mathworks.com/help/stats/kmeans.html>
- [57] B. Sofman, "Online learning techniques for improving robot navigation in unfamiliar domains," Ph.D. dissertation, Robotics Inst., Carnegie-Mellon Univ., Pittsburgh, PA, USA, 2010.
- [58] C. Ye, "Navigating a mobile robot by a traversability field histogram," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 2, pp. 361–372, 2007.



**Stuart H. Young** received the B.S. degree in mechanical engineering from the University of Washington, Seattle, WA, USA, in 1991, and the M.S. degree in mechanical engineering from the Naval Postgraduate School, Monterey, CA, USA, in 1996. He is currently pursuing the Ph.D. degree in systems engineering with George Washington University, Washington, DC, USA. Since 2007, he has been the Chief of the Asset Control and Behavior Branch with the U.S. Army Research Laboratory, Adelphi, MD, USA. From 2006 to 2007, he was the Project Coordinator for Unmanned Systems and Countermeasures and Counter IED in the Program Manager-Future Combat Systems, Technology Office. From 1991 to 2006, he was Team Leader, researcher, and engineer with the U.S. Army Research Laboratory, Adelphi, MD, USA. He also served in the U.S. Army Reserve as an Engineer Officer, from 1991 to 2003. He holds four patents. His current research interests include the development of intelligent systems, autonomous behaviors for collaborating robotic teams operating in militarily relevant environments, and in their experimental evaluation.

Mr. Young is an Elected Member of the Tau Beta Pi National Engineering Honor Society and the Pi Tau Sigma International Mechanical Engineering Honor Society.



**Thomas A. Mazzuchi** received the B.A. degree in mathematics from Gettysburg College, Gettysburg, PA, USA, in 1978, and the M.S. and D.Sc. degrees in operations research from George Washington University, Washington, DC, USA, in 1979 and 1982, respectively.

Since 2000, he has been the Chair of the Department of Engineering Management and Systems Engineering, George Washington University, where he is also a Professor of Operations Research and of Engineering

Management, following three and a half years as an Interim Dean with the School of Engineering and Applied Science, from 1997 to 2000, and three years as the Chair of the Department of Operations Research from 1994 to 1997. He also holds responsibility as an Associate Director of the Laboratory for Infrastructure Safety and Reliability, where he conducts reliability research. His current research interests include applied statistics, Bayesian inference, quality control, reliability analysis, risk analysis, stochastic models of operations research, and time series analysis.

Dr. Mazzuchi is an Elected Member of the International Statistics Institute, and remains active in professional society matters. He is an Elected Member of the Operations Research Honor Society, Omega Rho, and the Science Honor Society, Sigma Xi. Since 1985, he has published scientific papers in such scholarly, refereed journals as the *Journal of Risk Analysis*, *Safety Science*, *Technometrics*, *Lifetime Data Analysis*, the *IEEE TRANSACTIONS ON RELIABILITY*, the *Journal of Statistical Planning and Inference*, and *Systems Engineering*.



**Shahram Sarkani** received the B.S. and M.S. degrees in civil engineering from Louisiana State University, Baton Rouge, LA, USA, in 1980 and 1981, respectively, and the Ph.D. degree in civil engineering from Rice University, Houston, TX, USA, in 1987.

He is the Faculty Adviser and the Head of the Engineering Management and Systems Engineering Off-Campus Programs Office, George Washington University (GWU), Washington, DC, USA. He has been a Professor of Engineering Management and Systems Engineering, since 1999, and the Founder and the Director of GWU's Laboratory for Infrastructure Safety and Reliability, since 1993. He served as an Interim Associate Dean for research and development, from 1997 to 2001, and the Chair of the Department of Civil, Mechanical, and Environmental Engineering, from 1994 to 1997. His current research interests include engineering management, systems engineering, civil engineering, and logistics management. He has conducted contract research work with such organizations such as National Aeronautics and Space Administration, Washington, the National Institute of Standards and Technology, Gaithersburg, MD, USA, the National Science Foundation, Arlington, VA, USA, the U.S. Department of Interior, Washington, the U.S. Department of the Navy, Washington, the U.S. Department of Transportation, Washington, and Walcoff and Associates Inc., McLean, VA, USA.

Dr. Sarkani is a member of Sigma Xi (as the GW Chapter President, from 1994 to 1995), a member of the Probabilistic Methods Committee, of Engineering Mechanics Division, ASCE, and served as the Chair of the Committee on Fatigue and Fracture Reliability from 1990 to 1994. He is a member of the Technical Administration Committee of Structural Safety and Reliability, ASCE. He is a Registered Professional Engineer in the State of Virginia.