



ARL-TR-10064 • MAR 2025



Framework for Assessing Autonomy Research

by Craig Lennon

DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.



Framework for Assessing Autonomy Research

Craig Lennon

DEVCOM Army Research Laboratory

REPORT DOCUMENTATION PAGE

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
March 2025	Technical Report	START DATE 8 February 2024	END DATE 6 December 2024
4. TITLE AND SUBTITLE Framework for Assessing Autonomy Research			
5a. CONTRACT NUMBER		5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
5d. PROJECT NUMBER		5e. TASK NUMBER	5f. WORK UNIT NUMBER
6. AUTHOR(S) Craig Lennon			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) DEVCOM Army Research Laboratory ATTN: FCDD-RLA-JC Aberdeen Proving Ground, MD 21005			8. PERFORMING ORGANIZATION REPORT NUMBER ARL-TR-10064
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.			
13. SUPPLEMENTARY NOTES ORCID: Craig Lennon, 0000-0002-2229-2919			
14. ABSTRACT This report provides a framework for assessing the value of low technology readiness level autonomy research and examples illustrating the framework. After defining autonomy as a type of system function, we propose that system behaviors implementing such functions can best be assessed by evaluating the performance, robustness, and adaptability of the behavior at a task with military relevance.			
15. SUBJECT TERMS Military Information Sciences, autonomy, artificial intelligence, robotics, experimentation			
16. SECURITY CLASSIFICATION OF:		17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT UNCLASSIFIED	b. ABSTRACT UNCLASSIFIED	c. THIS PAGE UNCLASSIFIED	UU 42
19a. NAME OF RESPONSIBLE PERSON Craig Lennon			19b. PHONE NUMBER (Include area code) (410) 278-9886

STANDARD FORM 298 (REV. 5/2020)

Prescribed by ANSI Std. Z39.18

Contents

List of Figures	v
List of Tables	v
1. Introduction and Definitions	1
1.1 Definitions of Experiment, Assessment, and Evaluation	2
1.2 Definitions of Autonomy, System, and Robot	2
1.3 Definition of System Function and System Behavior	3
1.4 Definition of Autonomous Function and Autonomous Behavior	4
1.5 Definition of Task	5
1.6 System Engineering vs. Assessing Research	6
2. Example Research for Illustrating the Framework	7
2.1 Scene Segmentation	7
2.2 Object Detection	8
2.3 Semantic Perception	9
2.4 Semantic Planning	10
2.5 Semantic Navigation	11
3. Integrated Research Assessment	12
3.1 The Assessment Task	12
3.2 The Robot	13
3.3 Autonomous Behaviors	14
4. Performance	15
4.1 Landmark Detection and Observing	16
4.2 Semantic Planning Performance	19
5. Reliability and Robustness	21
5.1 Defining Reliability and Robustness	21
5.2 Measuring Robustness	23

6. Adaptability	25
7. Conclusion	27
8. References	28
Appendix. Questions to Ask When Planning Autonomy Assessments	30
List of Symbols, Abbreviations, and Acronyms	34
Distribution List	35

List of Figures

Figure 1.	Example of scene segmentation.....	8
Figure 2.	Detected gas pumps and traffic barrel (left), and doors (right).....	9
Figure 3.	Example of object detection with lidar fusion, with actual objects (left) and labeled point cloud images (right).....	9
Figure 4.	Example learned cost map for “left of.”	10
Figure 5.	Robot’s view (left) and predictions (right) for a navigation TBS.....	11
Figure 6.	Robot as configured for the 2015 assessment.....	14
Figure 7.	The sites of the experiment.	23
Figure 8.	Example vignette template and robot initial position.	24
Figure 9.	Experimental unmanned vehicle (XUV).	26

List of Tables

Table 1.	Examples of different levels of functions.	4
----------	---	---

1. Introduction and Definitions

Autonomy is a critical technology area for defense (Office of the Under Secretary of Defense for Research and Engineering 2022), so assessing the potential value of fundamental and applied research in autonomy is an important challenge to overcome. This report presents a framework for conducting experiments to assess the value of autonomy research at low technology readiness levels (TRLs), along with examples of past assessments showing how to apply the framework. One should initially assess research at the most isolated or reduced level at which the value of the research can be identified. If you are researching a new type of sensing mechanism, you demonstrate its value in a breadboard configuration before building a prototype sensor, and then test the prototype sensor before attempting to integrate it onto a platform. We propose that autonomy involves the execution of a system function, and as such, is difficult to demonstrate with an isolated component, even if the component is the subject of the research. An evaluation of autonomy research should be based on observing a system perform an action autonomously.

Some autonomy research may use a simulated system, but often we use a surrogate system consisting of a platform that stands in for the tactical platform on which the research might be fielded. We will refer to a surrogate system intended to demonstrate the value of autonomy research as a demonstration system. In assessments of autonomy, the goal is to select a demonstration system and task such that the performance of the system at the task may reasonably be expected to show the potential value of the research on a more mature system. Thus, even in low TRL research into autonomy, we often need to construct a demonstration system, design functions of that system, and identify a task with military relevance that the system can perform. We propose evaluating autonomy research using a framework consisting of the following steps:

1. Identify an autonomous behavior that the research could enable or improve.
2. Select a task in which this behavior could be employed.
3. Design experiments to assess the research by
 - a. Measuring the performance of the behavior with respect to the task, and
 - b. Measuring the robustness of performance with respect to changes in the environment in which the task is performed.
4. If the results of step 3 are promising, continue experimentation to assess the adaptability of the behavior.

In the remainder of this section, we will define relevant terms, including “system function,” “autonomous function,” “autonomous behavior,” and “task.” As this framework is conceived of as being applied to robots, we illustrate its application with an example of research integrated on a single robot. Section 2 presents the research used as an example, Section 3 describes the task and behaviors used to assess the research, and Sections 4–6 explain how the U.S. Army Combat Capabilities Development Command Army Research Laboratory measured performance, robustness, and adaptability in previous experiments.

1.1 Definitions of Experiment, Assessment, and Evaluation

We use the terms “assessment,” “evaluation,” and “experiment” in this report, and define these terms as follows:

Experiment: The structured study of a subject to measure or otherwise understand the subject. It is designed to answer questions or to test hypotheses selected in advance of the experiment. It is an event occurring at a specific time and location.

Assessment/Evaluation: A judgment about how much progress has been made with respect to a research objective. It may examine the results of one or several experiments to provide evidence as a basis for the judgment.

1.2 Definitions of Autonomy, System, and Robot

Our perspective is that the term “autonomous” does not make sense as a description of a system but does make sense in describing things that systems do. To explain why, we need to present current definitions of autonomy, system, and system function. Definitions of autonomy and autonomous system, respectively, are as follows:

Autonomy: “[T]he level of independence that humans grant a system to execute a given task in a given environment. The condition or quality of being self-governing to achieve an assigned mission based on the system’s own situational awareness (integrated sensing, perceiving, analyzing) planning and decision-making. This independence is a point on a spectrum that can be tailored to the specific mission, level of acceptable risk, and degree of human-machine teaming” (Hicks 2023).

Autonomous system: “A weapon system that, once activated, can select and engage targets without further intervention by an operator. This includes, but is not limited to, operator-supervised autonomous weapon systems that are designed to allow operators to override operation of the weapon system but can select and engage targets without further operator input after activation” (Allyn 2017).

We propose that the term “autonomy” should not be applied to a system, but only to system functions. A DoD-accepted definition of system follows.

System: “[A]n aggregation of system elements and enabling system elements to achieve a given purpose...” and those system elements may include “hardware, software, people, data, processes, facilities, and tools” (Office of the Deputy Director for Engineering 2022).

By this definition, humans using a robotic platform are part of the same system within which the platform operates, so it is incorrect to say that the human grants the system independence to perform a task. Rather, certain system functions may be executed autonomously while the system performs the task. The framework proposed here was conceived of for robots, operating individually or as a group, where “robot” is defined as follows:

Robot: “[A] powered machine capable of executing a set of actions by direct human control, computer control, or both. It is composed minimally of a platform, software, and a power source” (Allyn 2017).

A robot is an element of a system that includes the robot, users, means of controlling and maintaining the robot, and other elements needed for the robot to perform its intended system functions.

1.3 Definition of System Function and System Behavior

System function: An action or activity intended to be performed by the system.

Systems have higher-level functions, such as refueling a vehicle, and system elements, such as a single robot, have lower-level functions, such as retrieving an object from an area. Components of the robot have even lower-level functions, such as a robotic arm grasping the object to be retrieved. Table 1 gives examples of functions assigned to a system and to parts of that system to illustrate how system functions might be assigned to system hardware and software during design.

Table 1 is provided to illustrate the concept of different levels of system function, but this report will only distinguish between these different levels when the distinction matters. When the distinction is not relevant to the point, the generic term “system function” may be used to refer to a function assigned to a system, subsystem, or any other level of system decomposition.

Table 1. Examples of different levels of functions.

Relative level	System elements	Function
System	Whole system	Place all fueling equipment in assigned places.
Platform	Robot	Retrieve fuel cans and place them on shelves.
User	Human	Check to make sure no fuel has spilled and cans are sealed.
Subsystem	Imaging	Identify the location and orientation of a gas can in the robot's field of view.
Part of imaging subsystem	Object detection	Receive images and place a bounding box around all instances of the object "gas can."
Component	Algorithm	The object detection algorithm that detects gas cans.
Component	Processor / GPU	Runs object detection algorithm.

A system function describes something the system is intended to do, and systems can be designed in different ways to accomplish these functions. When the system is complete, the system will act in some way to execute its function, and we refer to the actions the system takes to execute a function as the behavior of the system.

System behavior: An action or activity performed by the system when executing a system function.

If the system function is to retrieve a fuel can, the robot retrieving the fuel can will execute a sequence of actions that constitute the behavior, and which should accomplish the function of retrieving the can. With definitions of systems, system functions, and behaviors, we can define autonomous functions and autonomous behaviors.

1.4 Definition of Autonomous Function and Autonomous Behavior

We believe the following definitions of autonomous function and autonomous behavior are consistent with the definition of autonomy from Hicks (2023) presented in Section 1.2.

Autonomous function: A system function intended to operate without manual control.

Operating without manual control does not mean operating without human control. A system executing an autonomous function is still supervised by a human, who may even have the option of terminating the autonomous execution to take manual control, or who may constrain or direct the execution of the function in other ways. A system function is an action or activity the system is intended to perform, and may be implemented in a variety of ways, with the behavior of the system depending on the implementation.

Autonomous behavior: An action or activity performed by the system when executing an autonomous function.

The system's behavior is a set of actions that will be observed when a system is accomplishing a task for which the system function would be needed. Evaluating performance will require measuring autonomous behavior and comparing it with an expected execution of the function.

1.5 Definition of Task

For the application of our framework, we define a task as follows:

Task: A specified goal or problem for a system to accomplish or solve.

A task has criteria by which one can determine if it has been accomplished correctly and required conditions (preconditions) under which it can be undertaken. For those interested, there is an Institute of Electrical and Electronics Engineers standard for how to represent robot tasks (IEEE 2023), and our simple definition is consistent with it and with the definition in Huang (2008).

The Army has defined many military tasks for individuals and units. As an example of an individual task, consider the following task from the *Soldier's Manual of Common Tasks* (Headquarters, Department of the Army 2009), which is identified by name, required conditions, and the standards for accomplishing it.

Task name: Navigate from one point on the ground to another point while dismounted.

Conditions: Given a standard topographic map of the area, scale 1:50,000, a coordinate scale and protractor, a compass, and writing materials.

Standards: Move on foot to designated points at a rate of 3,000 m/h.

As an example of a collective task, consider area reconnaissance. This is a task a scout platoon would perform, which requires the platoon to “[find] and report all enemy forces, reconnoiter specific terrain, and [report] all reconnaissance information...” (U.S. Army Publishing Directorate 2024). This collective task has several pages worth of subtasks that must be executed to successfully complete it, with two examples being:

- Identifies enemy elements, and
- Locates and reports bypasses around all urban areas, obstacles, and contaminated areas.

Military tasks, with accompanying standards, provide inspiration for tasks used to assess the value of autonomous behaviors, but one cannot evaluate a robot based on the standards of the military tasks, especially in the case of low TRL research. Common Soldier tasks presuppose that the Soldier executing the task can talk, understand orders, identify and pick up objects, and take other actions common to humans. These abilities common to humans are often the subject of autonomy and AI research, so rather than evaluating the robot at a military task, we identify aspects of the task that are useful for evaluating the research and design the experiment to focus on the research-related aspects, while compensating for the robot's inability to perform common human functions.

1.6 System Engineering vs. Assessing Research

Since our framework proposes to evaluate autonomy research based on the performance of a system function at a militarily relevant task, we introduced system engineering terminology. To avoid confusion, this section will clarify some differences between system engineering and assessing research. One might engineer a system to solve a task as follows:

- Identify the task and what needs to be done for a solution.
- Design system functions that would allow the system to solve the task.
- Identify existing technology, or promising research, that could be used to implement those functions as behaviors.
- Verify that the correctly executed behaviors solve the task.

For the system engineer, the task the system must perform is important, and research is only useful if it can implement the solution. By contrast, one proceeds as follows when assessing research:

- Identify an autonomous behavior with military relevance that the research can enable.
- Select an appropriate task that provides performance parameters for the experiment.
- Determine the appropriate environmental and experimental conditions.
- Conduct the experiment.

When assessing research, the behavior subject to experimentation is the key point, and the task and other system functions support the experiment. Because of this, the task or overall system functionality may change during the experiment if needed to better evaluate the research, which would be bad practice for system engineering, but the correct choice for assessing research.

Sections 2–5 illustrate steps 1–3 of the framework from the Introduction. In Section 2, we describe research at TRL 1–3 conducted by the ARL Robotics Collaborative Technology Alliance (RCTA), which needed to be assessed. Section 3 identifies the tasks selected to assess that research, and Sections 4 and 5 present how performance and robustness were assessed with respect to the task.

2. Example Research for Illustrating the Framework

The process for evaluating research begins with understanding the research being assessed and what potential applications it might have. This section will describe three research projects related to robotic perception and navigation conducted by members of the RCTA between 2011 and 2015, and the potential applications identified for the research at the time of the assessment. Fundamental research can be applied to many different applications, but we choose one application to evaluate in an experiment. The three research projects were algorithms used in scene segmentation, object detection, and semantic planning. Eventually, all three algorithms were integrated onto one robot for an integrated research assessment, consisting of the set of experiments described in Lennon et al. (2015a).

2.1 Scene Segmentation

The first fundamental research project was scene segmentation. An algorithm was trained on image data in which each pixel had been given a class label. The trained algorithm then took an image as input and produced as output a class label for each pixel of the input image. Figure 1 illustrates the process, by showing an image of a building in which each pixel has been shaded by a color corresponding to its class label. An application identified for scene segmentation was the labeling of terrain and landmarks, and Figure 1 shows this application of the algorithm. This application was considered useful because given labeled terrain and landmarks, a robot could plan routes related to terrain (e.g., avoiding some types of terrain) and relative to landmarks (e.g., being told to go to a goal left of or behind a landmark). With this application in mind, the scene segmentation algorithm was tested on terrain and buildings (Lennon et al. 2013), with the classes it was trained on shown in Figure 1. In later experiments, classes of fire hydrant, traffic barrel, and vehicle were also used as classes in the scene segmentation algorithm (Oh et al. 2017).

To avoid confusion, we note that although terrain-based navigation was intended as an application of scene segmentation, the system used in the integrated assessment described in Section 3 did not perform terrain-based navigation because this behavior could not be integrated into the demonstration system. At the integrated assessment, because terrain-based navigation was not functional, scene

segmentation was assessed based on its ability to detect landmarks rather than terrain. The integration issues preventing terrain-based navigation were not anticipated when the experiment (Lennon et al. 2013) was conducted, as this was prior to attempts to integrate the research into the system. As our intent is to illustrate our framework, we present the assessment process as it played out, including the fact that a behavior we intended to assess (terrain-based navigation) did not function at the time of the integrated assessment. This is why behaviors related to terrain-based navigation were identified and planned for, and appear in Section 3.3, but the performance of such behaviors is not described in Section 4.

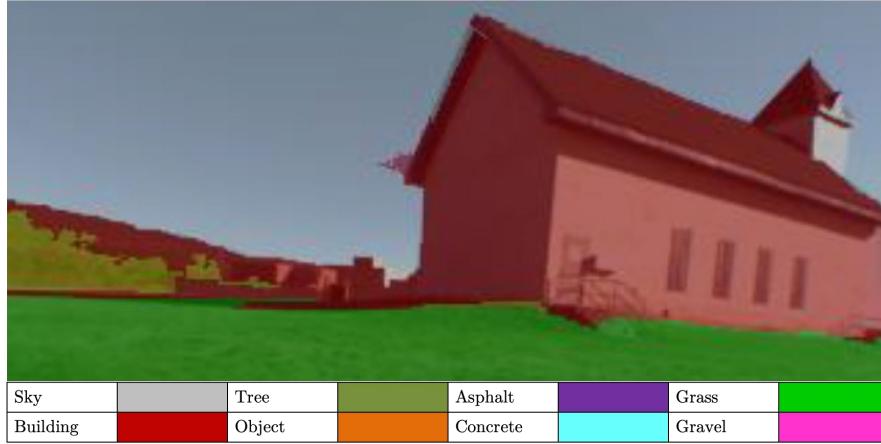


Figure 1. Example of scene segmentation.

2.2 Object Detection

The second fundamental research project was the detection of objects, along with the parts and orientation of the object. In this research thrust, objects were identified in an image using a deformable parts object detection algorithm (Zhu et al. 2014), which could detect and label an object and the parts of the object. While the scene segmentation algorithm labeled every pixel, the deformable parts model (DPM) just identified the object and relevant parts. The detection of objects using the DPM allowed the robot to navigate with respect to, manipulate, or otherwise act on objects. Three applications were considered: the first was locating an object and identifying the handle of the object to be grasped by a robotic arm, the second was the detection of landmarks, such as gas pumps and traffic barrels, and the third was detection of doors and people. Figure 2 shows images in which the objects were detected, classified, and outlined with a bounding box. In the left image of Figure 2, two gas pumps and a traffic barrel have been correctly detected and outlined by white bounding boxes, while two false positives in the clouds are outlined by blue dashed bounding boxes. The blurry text above the bounding boxes was a label including the object class and a reference for the object as it was recorded in the

robot's database. On the right image, two correctly detected doors are outlined by white bounding boxes.

The application of identifying a handle for grasping was done in a separate set of experiments (Childers et al. 2016), while the detection of landmarks, doors, and people is used as the example application in Section 3.



Figure 2. Detected gas pumps and traffic barrel (left), and doors (right).

2.3 Semantic Perception

When scene segmentation and object detection algorithms were integrated onto a robot, bounding boxes (from DPM) or pixels (from scene segmentation) with a class label were mapped via a process in which lidar returns corresponding to the bounding box or pixels of the object or terrain were given the appropriate class label and placed into the map. The robot used this process to develop a metric map with objects and terrain designated by class labels. Figure 3 shows an example of labeled point clouds, with the left side showing the image and the right side showing class labels applied to lidar points.



Figure 3. Example of object detection with lidar fusion, with actual objects (left) and labeled point cloud images (right).

We referred to a robot populating its map of the world with features that had meaning to a user as semantic perception, since the robot’s perception was intended to be a meaningful reference for the user. Using this map, the robot could interpret commands given with reference to local objects, like “go to the car behind the building.” In addition to making navigation commands easier, this mode of instruction and planning could be done with the robot having only a local map, which would work in a global positioning system (GPS)–denied environment and in an area in which up-to-date maps are unavailable. Semantic perception was applied research, which took two fundamental research projects (scene segmentation and object detection) and integrated them into the sensing and mapping capabilities of a platform to demonstrate their value in an integrated research assessment.

2.4 Semantic Planning

The third fundamental research project was semantic planning, which used inverse optimal control to map navigation constraints like “stay left of the building” into cost maps used by lower-level path planners (Boularias et al. 2015). The application identified for semantic planning was the planning of paths based on navigation instructions given in terms of objects in the robot’s environment. Figure 4 shows the type of cost map generated for the command “left of,” which places high costs on paths to the right and low costs on paths to the left of whatever it is told to go “left of.” We refer to reference objects or terrain features as landmarks, so in the commands “go left of the vehicle” or “go behind the building,” the vehicle or building would be the landmark.

The command provided to the semantic planner was required to be in a structured language, called a tactical behavior specification (TBS), the grammar of which is described in (Boularias et al. 2015).

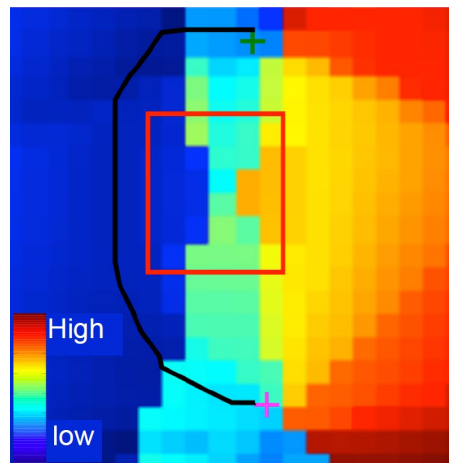


Figure 4. Example learned cost map for “left of.”

2.5 Semantic Navigation

Semantic navigation refers to the applied research needed to integrate semantic planning with existing robot capabilities, such as mapping and low-level path planning. With semantic perception placing landmarks into a map, semantic navigation could use these landmarks for navigation. Upon receiving a TBS like “go to the traffic barrel behind the building which is right of the vehicle,” the robot could:

- Identify referenced objects that are in view as the referenced objects in the command, which we refer to as grounding to the objects. With this TBS, the vehicle and building should be identified as having been referenced.
- Identify when no objects of the type referenced are within its map and hypothesize the position of a referenced object relative to observed objects. In this example, the traffic barrel behind the building should be identified as missing and be hypothesized as being somewhere behind the building.
- Plan a path from its current position to its goal based on its current map, replanning at regular intervals. In this example, the path should go from its current position to the hypothesized traffic barrel. The robot should continue replanning until it is behind the building and perceives a traffic barrel. Then it should ground to that traffic barrel as the referenced goal object and stop near it.

Figure 5 shows an example of scene perception (left) and the path planned on the semantic map (right). The starting position of the robot is marked by a green triangle, pointing in the direction of the robot’s starting orientation. The robot could see a traffic barrel in front of the building but was directed to go to an out-of-sight traffic barrel behind the building, which it predicted a position for, as shown in the right image of Figure 5. In the right side of Figure 5, the building walls that the robot could see are shown in bright blue, and the light gray walls were hypothesized by the robot, as was the barrel labeled “predicted traffic barrel.”

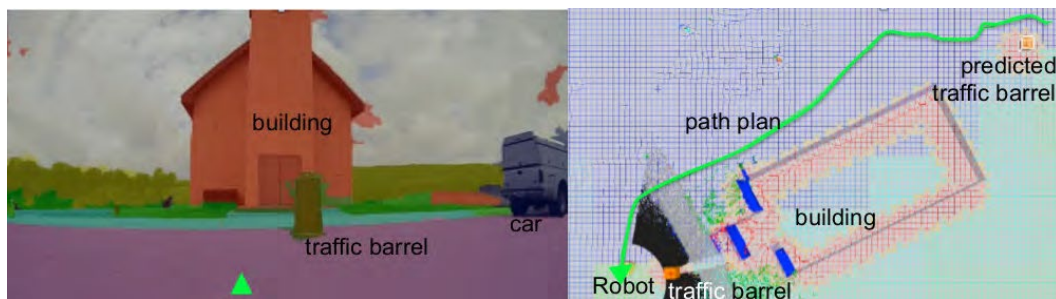


Figure 5. Robot’s view (left) and predictions (right) for a navigation TBS.

This section presented research and applications of that research which would enable autonomous behaviors on a robot. To review, the following is a list of each research thrust and the behavior it enabled:

- Fundamental research in scene segmentation, in which an algorithm classifying pixels enabled perception of terrain and landmarks.
- Fundamental research in object detection, in which a deformable parts model, detecting objects and their parts, enabled perception of landmarks, doors, and people.
- Fundamental research in semantic planning, in which an inverse optimal control algorithm mapped structured language into cost maps and goals.
- Applied research in semantic perception, in which scene segmentation and object detection were integrated into existing robot capabilities.
- Applied research in semantic navigation, in which semantic planning was integrated into existing robot capabilities.

3. Integrated Research Assessment

The preceding section described three fundamental research projects and the applied research needed to integrate them into existing robot capabilities, which needed to be assessed. This section will identify the task selected for the assessment of the research in an integrated research assessment, which was the term used to describe the assessment of research integrated onto a platform. Then we will briefly describe the platform used in the experiment and the autonomous behaviors we expected to observe during the experiment.

3.1 The Assessment Task

The perception and navigation applications described in Section 2.3 seemed suited to reconnaissance and surveillance tasks, so the RCTA leadership decided to conduct experiments assessing the performance of these behaviors at a surveillance task referred to as “screen the back,” in which a robot is instructed to navigate to the back of a building, observe doors in the building, and identify when people come out of the doors. Successfully accomplishing the task would require the robot to navigate along a path, while adhering to TBS constraints, to a referenced goal point, and then to detect and observe doors on a referenced building, and detect people coming out of those doors. To accomplish this task, the robot would need these functions:

1. Classify terrain based on images.
2. Detect and classify objects, including landmarks, doors, and people.
3. Produce a semantic map of its environment, which it can use to navigate.

4. Translate a structured language command into a goal and cost map, including making predictions for unseen landmarks.
5. Plan a path to the goal using the cost map.
6. Follow the planned path, replanning as needed.
7. Detect doors and people coming out of the doors.

It is worth noting that while functions 1–7 were needed for the task, the fundamental research to be assessed at the experiment contributed directly only to functions 1, 2, 4, and 7. The other functions were needed for the task, but do not depend on the research presented in Section 2. As mentioned in Section 1.6, the desire to demonstrate the value of the research at a relevant task led to the need for system functions that were not part of the research being assessed but were needed to perform the task. To evaluate the research, the experiment had to be conducted so that subsequent analysis could evaluate the research of Section 2 based on observing the robot’s autonomous behaviors during the execution of functions 1, 2, 4, and 7.

3.2 The Robot

The robot used in the assessment was a Clearpath Husky (described in more detail in Lennon et al. [2015a]). Figure 6 shows that robot, which had about 5 inches of ground clearance. The small Hokuyo lidar on the front was used for obstacle avoidance. The Adonis camera produced images for object detection and semantic segmentation. The output of those algorithms was used, in combination with the XR lidar on top, for mapping.

The platform in Figure 6 was used to demonstrate the value of autonomous behaviors, which could someday be employed by a military system. Because we were experimenting to assess the behaviors (and the algorithms that produced those behaviors), and not the platform, we made accommodations for the platform during experiments. We chose experiment sites that could be traversed by the platform, started the platform in positions that were within lidar range of navigation landmarks, removed false positive obstacles, like tall grass, which would have hindered the platform, and made sure each goal could be safely navigated to if the robot planned a path according to an appropriately interpreted TBS. The point of the assessment was not to evaluate a developmental or operational system performing a military task, but to assess research level autonomous behaviors integrated onto a demonstration system performing a relevant task.

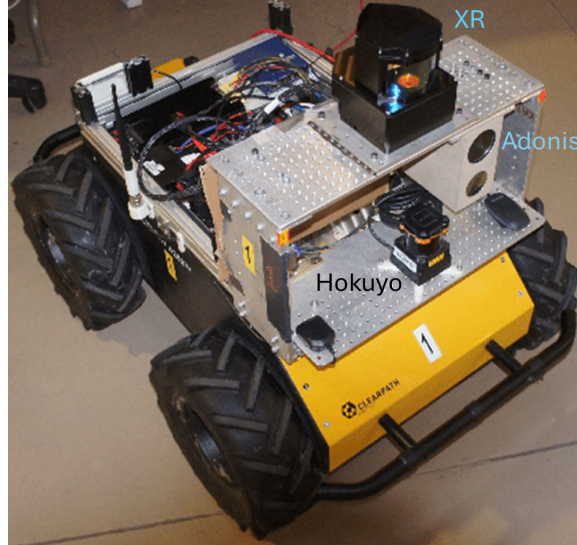


Figure 6. Robot as configured for the 2015 assessment.

3.3 Autonomous Behaviors

The behaviors we expected to observe as we prepared for the integrated research assessment in 2015 follow.

1. *Terrain classification*: Segment images by terrain type and fuse the resulting segmentation with lidar to map buildings and terrain. The map would be local, independent of GPS or prior maps, but buildings and terrain should be correct in terms of their relative distance from one another. Scene segmentation was the only research used in this behavior.
2. *Landmark detection*: Detect objects (e.g., fire hydrants, vehicles, and traffic barrels) within an image, and fuse the resulting detection with lidar to place them into a map. As above, the map is local but should be correct in terms of relative distance. Both the deformable parts model and scene segmentation were used in this behavior.
3. *Mapping*: A semantic map is created by the fusion of object detection and scene segmentation data. Objects seen multiple times should be identified as the same object, and objects and terrain should be mapped with correct relative distances. No research from Section 2 was used in this behavior, so the map produced by this behavior was only used as a means of evaluating other behaviors.
4. *Semantic planning*: Given a TBS, produce a goal and cost maps consistent with the TBS. This behavior used the inverse optimal control research.
5. *Path planning*: Given a goal and cost map, produce a plan for navigating to the goal using the cost map. This behavior did not use any research from Section 2, but the proper function of this behavior made semantic planning

easy to evaluate, since the path it produced could be inspected to see if it adhered to the TBS.

6. *Navigating*: The robot should follow the planned path, stopping to replan as needed, and should not collide with obstacles. It should stop within a predetermined distance from the goal. This behavior did not use any research from Section 2, only the output of the planner and a preexisting obstacle detection behavior.
7. *Observing*: When the robot is positioned facing a building, it should detect doors in the building and detect people emerging from those doors. This behavior used the DPM research. Observing was similar to landmark detection, except that in landmark detection, the robot needed to repeatedly detect and classify the landmark as it moved, while observing required detection from a single position.

We expected the robot to execute behaviors 1–7 without any human intervention, and we designed experiments evaluating the robot’s ability to do so. While the robot was graded on its performance of the whole task, the assessment in Lennon et al. (2015a) was primarily used to evaluate the research used for behaviors 2, 4, and 7. As mentioned in Section 2.1, terrain-based navigation was not functional at the 2015 assessment, so behavior 1 was not assessed. Next, we explain how performance was evaluated at the integrated research assessment.

4. Performance

Performance describes the execution of a system function, typically by means of one or more measures of performance by which the execution of the function can be evaluated (Space & Missile Systems Center 2004). For example, if a platform’s function is to drive, performance might be measured by the distance and speed it can travel across different terrain types. Testing system functions at high TRL measures the observed performance of the function against the required performance. Low TRL experimentation provides evidence for the value of research by demonstrating how the research products perform, and by providing information about the benefits and limitations of the research, which can guide decisions about future research. Therefore, at low TRL there is not an established level of task performance that needs to be met for the experiment to be successful; rather, an experiment is considered successful when the contribution of the research to task performance is correctly measured. The RCTA integrated assessment presented in this section provides an example of how we measured the performance of a demonstration system executing an autonomous behavior at a relevant task.

In Section 3.3, seven behaviors were identified, four of which were needed for evaluating fundamental research conducted by RCTA members and the applied research needed to integrate that fundamental research into existing system capabilities: terrain classification, landmark detection, semantic planning, and observing, with the other behaviors supporting the evaluation by enabling the robot to complete the task used to demonstrate the research. As mentioned in Section 2.1, terrain classification was not evaluated as a behavior on the robot due to integration challenges, so performance measures for only three of these four behaviors are presented here. We will compare performance measures used for the algorithms prior to integration on the system with performance measures of system behaviors used after the algorithms were integrated onto the system.

4.1 Landmark Detection and Observing

The scene segmentation and deformable parts algorithms were both used for the landmark detection behavior. Prior to integration, scene segmentation was assessed in Lennon et al. (2013). In that evaluation, pixels of each image received a class label assigned by a human labeler, and pixel class labels generated by the algorithm were compared to the class labels assigned by the human, with the latter considered correct. A variety of metrics were used to measure performance, including measures such as class precision and recall and F1 score. As the assessment was of an algorithm independent of a system behavior, the metrics were not related to an autonomous behavior. It did not matter if incorrect pixels were clustered together, which might result in terrain looking navigable when it was not (for terrain classification) or in the false positive detection of a landmark, or if the incorrectly classified pixels were scattered throughout the image in a way that might be corrected by filtering. It also did not matter what types of class confusions occurred. Labeling sky as grass counted the same as labeling gravel as asphalt. The algorithm was evaluated with metrics unrelated to its intended use, but which had good property: that a perfect score on all the metrics implied good performance in any behavior. But a better score on a pixel-based metric does not necessarily mean better performance of the behavior if the score is not perfect. For example, if terrain classification is to be used for navigation, then pixel class errors for sky, or at the tops of buildings, may not be relevant, and the labels of pixels closer to the robot may matter more than errors far from the robot. Isolated pixel errors may matter less than pixel errors that are clustered. While the terrain classification behavior was not part of the integrated assessment, the scene segmentation algorithm was used for landmark detection behavior in that assessment.

Since the DPM algorithm was also used for landmark detection, we will describe how that algorithm was evaluated prior to integration before describing

performance evaluations for the landmark detection behavior. The DPM algorithm was not designed to segment an entire scene, but to identify specific objects within the scene, using parts of the object as cues. This research thrust was pursuing the detection of objects that might subsequently be manipulated by the robot, hence the focus on parts of the objects. The assessment of the algorithm was performed on a standard academic dataset, as described in Zhu et al. (2014), in which the algorithm classified objects in images. This assessment was a reasonable proxy for landmark detection, since in this behavior the algorithm bounds and then applies a class label to objects. Like with the pixel-based evaluation of scene segmentation, a perfect score suggests the algorithm will be useful in any behavior, but otherwise, a higher score may not correspond to better performance of a behavior.

To see how the acceptable level of performance of a subsystem can depend on the rest of the system, consider landmark detection, in which a robot might repeatedly see the same landmark from different perspectives. The robot may not need to classify the landmark correctly in most images, just enough to map where it is, and if objects are retained in the map after detection, it may be that false positives are of much greater importance than false negatives. Further, landmark detection may not be hindered by the bounding box being mislocated, and perhaps it is enough if the true bounding box overlaps with the box placed by the robot. Under such conditions as these, false positives are more important than false negatives, and measures like intersection over union (IOU) are of limited value because it only matters if IOU is greater than zero. By contrast, if landmarks are only retained in the map if they are seen with consistency, false positives could matter less than false negatives. Thus, the acceptable level of performance depends on how the research is integrated into the system. Acceptable performance also depends on the task. For example, if an algorithm were being used to track an object—e.g., a platform on which a drone was to land—it would be important that the landing pad was correctly bounded, and IOU might be important as a measure of performance.

We note in passing that the DPM algorithm was also used in an autonomous grasping experiment, in which its ability to detect parts and object orientation was evaluated. In that experiment, a DPM model detected a gas can and identified the handle of the gas can which was autonomously grasped by a robotic arm. Further detail is available in Childers et al. (2016).

Since both the scene segmentation and DPM algorithms could classify traffic barrels and fire hydrants, we chose to evaluate the landmark detection behavior based on these two classes. Choosing objects detectable by both algorithms let us compare the algorithm’s performance, while choosing objects that we could rearrange (the fire hydrant was not attached to the ground) allowed us to vary the environment. Since the relevant behavior was the detection of landmarks by which

to navigate, the evaluation was based on the false positive and false negative detections of traffic barrels and fire hydrants. This performance measure was task relevant, but it relied on the output of a mapping process in which landmark detections could be interpreted as new or as re-detections by the mapping algorithm. To evaluate the landmark detection behavior separate from mapping, the team needed to distinguish between errors in perception and errors in mapping.

To evaluate the integrated semantic perception and mapping subsystem (combining behaviors 2 and 3), the assessment team compared the final maps produced by the robot with ground truth maps created by the team. This allowed for a comparison of the robot's representation of its environment with the actual physical environment, with performance being measured by the correct placement of detected objects into a map. Objects shown in the map that were too far from real objects of the correct class were considered false positive, and real objects with no detected object mapped nearby were considered false negatives. To assess the semantic perception of applied research separate from mapping, recordings of runs were replayed, with evaluators watching the robot's user interface to observe detections as they were transmitted to the mapping component, in addition to examining the final map. Instances in which the robot faced an object without detection were counted as false negatives, and instances in which the robot recorded more objects present than were present were counted as false positives. This also allowed errors to be separately tracked for the scene segmentation and DPM algorithms. Both performance measures were relevant to the behavior, as observing detections provided more information about the fundamental research enabling landmark detection, and evaluations of the final map provided information as to how landmark detection worked in combination with the established mapping behavior to evaluate the applied semantic perception research.

Since the same method of tracking false negatives and false positives was used for both the scene segmentation algorithm and for the DPM algorithm, proportions of false negatives and positives could be compared for the barrel and fire hydrant classes. The algorithms were not compared because they were interchangeable, as each algorithm had a different purpose, but the comparison improved the understanding of both algorithms. For example, we discovered that scene segmentation produced a high proportion of false positives (0.78) for traffic barrels and no false negatives, but not for fire hydrants, for which it had a high proportion of false negatives (0.53) and no false positives. The DPM algorithm had a low proportion of false positives for barrels and fire hydrants (.06 / 0) and a moderate proportion of false negatives for both (0.22 / 0.27). The difference in errors between the algorithms suggested that the errors were attributable to the training and

parameter tuning of the algorithms rather than to anything inherent in the environment in which the experiment took place.

The DPM algorithm was also used in observing behavior, which required the detection of doors and pedestrians. In this behavior, performance was again assessed based on whether correct or incorrect detections were made. The behavior was evaluated by comparing detections shown on a user interface to ground truth (also shown on the interface) during the experiment. The assessment is in Lennon et al. (2015a), but we note here that there were no false positives for doors and pedestrians, only false negatives. This was consistent with the performance of the same DPM algorithm at the landmark detection task, in which false negatives were common, but false positives were rare. That is not surprising since it was the same algorithm, but the conditions of use in the behaviors differed, since doors were generally recessed and in shadow, while the landmarks were generally prominent and out of shadow. Similar performance on similar tasks suggests an algorithm might be generally well (or poorly) suited for a particular type of task.

4.2 Semantic Planning Performance

Recall that the semantic planner produced a goal point and a cost map for visible terrain based on the TBS provided to it, as shown in Figure 4. In that figure, which showed a cost map produced in simulation, the simulated robot had a map of the entire region, so the cost map could cover the entire navigation path from start to goal. Figure 5 shows an actual run, in which the robot had not seen all of the area through which it had to navigate, and hypothesized the location of the goal point and the parts of the building it could not see. The goal and cost map were input to a lower-level planner, which would plan a minimal cost path from its position to the goal. This path was visible on the robot’s user interface. Since the lower-level path planner was relatively mature technology, we assumed that it was correctly producing a minimal cost path given the cost map provided. So contingent on perception functioning properly, we felt confident evaluating the semantic planner by examining this lower-level path and attributing the correctness or incorrectness of this path to the semantic planner. For many runs in the experiment, the goal was out of sight at the start, so regular replanning was needed throughout the run to reach the goal. In these instances, successful performance required all plans generated during the run to adhere to constraints, and for the robot to reach the goal.

The assessment of semantic planning prior to the integrated research assessment was conducted in simulation and with a real robot in a simplified environment (Boularias et al. 2015), so that failures of perception would not interfere with the assessment of the algorithm. Perception did not matter in simulation, since a map

was provided to the robot, and correct perception of landmarks and goals was monitored on site in the simplified environment, so the performance of the algorithm could be evaluated based on whether the robot reached its goal while adhering to constraints. In this simplified environment, the goal was in sight at the start, and commands were no more complicated than “navigate to building left of fire hydrant.” In the integrated assessment, a goal might be out of sight and a command might include constraints on how the robot reached its goal, like “stay left of the building; navigate to a traffic barrel behind the building.”

At the integrated assessment, semantic planning was evaluated by examining the proposed path plan, and semantic navigation was evaluated by observers who watched the robot move and evaluated the robot’s adherence to constraints and its success in reaching the goal. Successful semantic navigation implied the success of semantic planning, but semantic planning could succeed even if semantic navigation failed. Rather than grade semantic navigation as pass/fail, the observers assigned a grade between 0 and 100, which allowed for an element of subjectivity in case the robot did something unexpected during the run. For grades of less than 100, graders documented the reason for the grade to identify failures observed when the behavior was being executed. In addition to informing researchers as to failure modes associated with the behavior, this allowed evaluators to identify when failures to reach the goal resulted from behaviors related to perception or to navigation. For example, a robot might fail to see a landmark or incorrectly perceive an obstacle that prevented it from reaching the goal. In many runs, the robot stopped too far from the goal, which was attributed to a parameter in the planning algorithm that could be adjusted, as opposed to a failure of the semantic planner.

The measure of performance for semantic planning in the integrated assessment was similar to the performance measure of the algorithm prior to integration, and the same is true for the landmark detection behavior and the evaluation of the DPM algorithm. The measures of performance for scene segmentation differed substantially from the measures used for landmark detection, which is partially attributable to the fact that scene segmentation was intended to be used in terrain-based navigation, but shortfalls in integration precluded terrain-based navigation, so semantic perception could only be measured by landmark detection. In the next section, we define reliability and robustness, describe the difference between the two, and explain why we are concerned about robustness rather than reliability for low TRL research.

5. Reliability and Robustness

System performance can change over time, as the components of the system change, or with a change of the conditions under which the system is used. In this section we define reliability and robustness, and present examples of how one can assess robustness of autonomous behaviors for low TRL research.

5.1 Defining Reliability and Robustness

We start with a DoD-accepted definition of reliability.

Reliability: “The probability that [a system] will perform its intended function for a specified interval [of time] under stated conditions” (Office of the Deputy Assistant Secretary of Defense Systems Engineering 2018).

This definition focuses on time because reliability is frequently estimated for physical systems that change over time as physical stressors within the environment break the system down, and because estimating time between failure is needed to understand the value of a system. The “stated conditions” in the definition could encompass many things, including the source of system components, quality of the components or their assembly, or the how the system is used and maintained. Conditions also include the operational environment within which the system is used, which we will focus on when considering autonomous behaviors. Reliability is a system property, and reliability of a mature system performing a behavior is difficult to infer from a demonstration system. When a demonstration system has hardware or software failures, this does affect the experiment, but is generally not considered in assessing the research unless the research is the cause of the failure. Reliability of a demonstration system may be important for conducting experiments efficiently, but it is not generally considered when assessing research demonstrated on that system.

Operational environment: Everything going on around or in the system that is an input to the system.

For clarity, we note that inputs to the system include conditions relevant to the user of a robotic platform and all other system elements, not just the platform itself, but this report will only provide examples of the effect of environment on the platform. Most properties of the environment have no effect on system performance, so when describing an operational environment, one should attempt to do so in terms of those properties of the environment that do affect the system’s behaviors. Autonomous behaviors based on low TRL research may not be well enough understood for the operational environment to be clearly identified, and two environments that look similar to a human may not be similar to the robot.

Consequently, low TRL experiments typically choose one site at which to conduct experiments and try to keep the environment at that site the same while modifying the behavior so as to be confident that the change in behavior is due to their modifications and not to the operational environment changing. Once modifications of the behavior are complete, changes in the environment are more acceptable, because if an unchanging robot varies in performance with a change in the environment, this suggests that changed aspects of the environment are relevant to defining the operational environment of the system. Research frequently alternates between experiments under constant conditions to improve behaviors and changes to environmental conditions to understand what factors of the environment are relevant to operation. If performance varies substantially due to environmental conditions—e.g., between summer and winter or between forest and desert—that may indicate that these are distinct operational environments. If performance remains the same on average (without changing the robot), that suggests that these are equivalent operational environments, at least with respect to the performance of the behavior being measured.

We define robustness for a system as follows:

Robustness: A system’s ability to maintain a level of performance when the operational environment varies.

For higher TRL systems, the operational domain of the system may be documented—e.g., in an assurance case (Rinehart et al. 2017)—but for low TRL research, one may not know what factors of the operational environment are relevant to performance and will need to rely on the opinion of the researchers whose technology is being assessed.

Having defined robustness as the ability to maintain performance when the operational environment varies, we introduce the term *adaptability* to describe the system’s ability to reestablish a level of performance after changing operational environments. Adaptability is described by both the change in performance between environments and modifications to the system needed to reestablish a given level of performance when changing operational environments. Section 6 provides an example of experimentation at TRL 6, which assessed adaptability. To distinguish between evaluating robustness or adaptability at low TRL, one can proceed as follows:

- Pick an experimentation site and allow researchers to adapt the robot as needed for that site.
- Propose making changes to the site, or changing experimentation sites, and ask the researcher if the robot can perform as well in the changed or new site as currently configured.

- If they say yes, proceed with the changes, understanding that the variation in performance is describing the robustness of the behavior.
- If they say no, ask them to track what they need to do in terms of retraining algorithms, tuning parameters, or modifying the platform to adapt to the new site or conditions. These changes, and the difference in performance between sites after the changes are made, describe the adaptability of the behavior.

In summary, if the researcher thinks no changes to the research system are needed for new conditions, we take that to mean they believe that the new conditions are the same operational environment and we are examining robustness of the behavior. If changes are needed, then the changes and difference in performance describe the adaptability of the behavior. The next section shows how robustness was assessed during the integrated research assessment referred to in Section 4.

5.2 Measuring Robustness

The experiments described in Section 4 were conducted at a single site, but as a series of vignettes in which variation in robot performance was assessed by deliberately modifying the environment and initial conditions of the robot. A vignette consisted of a site, configuration of relevant objects, and robot starting position. Figure 7 shows an overhead image of the experiment area, which was a military training area at Fort Indiantown Gap. Three buildings used as vignette sites are labeled (church, bar, and gas station).



Figure 7. The sites of the experiment.

The relevant objects staged at each site were detectable landmarks, including vehicles, traffic barrels, and a fire hydrant which was movable (not attached to the ground). A template of a vignette from Lennon et al. (2015b) is shown in the left

side of Figure 8, with a robot in starting position shown on the right side of the figure. The positioning of the traffic barrels permitted systemic variation in TBS—e.g., instructions to go to the left, right, or center traffic barrels shown in Figure 8.



Figure 8. Example vignette template and robot initial position.

In addition to varying the location and number of landmark objects, vignettes varied conditions such as whether the robot could see the goal, and whether there was clutter (non-landmark objects) scattered around the landmarks. During the experiment, two different robots were configured similarly, with the intention that the robots should have the same hardware and software. For each vignette, there were two runs, with each robot conducting one run. Using two robots allowed the replication of runs to occur at different times of day, as one robot would do a selection of vignettes in the morning, and the other replicate those vignettes in the afternoon. Additionally, the TBS varied between vignettes, since the TBS depended on the arrangement of landmarks. Vignettes and replications were arranged as described because researchers provided no reason to believe that performance would vary based on the changes between vignettes. Each vignette should be equally likely to succeed, as far as researchers knew, so variation in the vignettes could be an estimate of robustness. After the experiments, factors such as clutter or building location were examined to determine if performance varied with that factor, and as it did not, we concluded those factors were not relevant to the behavior in the environment in which we ran the experiment.

In addition to estimating robustness, experiments at low TRL can support later system reliability processes by documenting failure modes of the system when it engaged in the behavior. This is often required for assessing the contribution of the research to the behavior, as reasons for unsuccessful performance and their relation to the research must be identified. Some failure modes, such as battery failure, may relate to the platform rather than the behavior, and would be resolved in the design of a more mature system, but others, such as grounding to the wrong object, are

modes related to the research being studied, and should be documented so that later consumers of the research will be aware of them. The experiments detailed in Lennon et al. (2015a) systemically varied one experiment site to evaluate robustness, and whether the factors varied were important for defining the operational environment. The next section shows how adaptability was studied across operational environments.

6. Adaptability

Adaptability: A system's adaptability is the system's ability to maintain or reestablish a given level of performance when changing environments. It is described by the effort required to achieve a given (e.g., acceptable) level of performance when changing operational environments, and the difference in performance after those efforts have been made.

While performance and robustness relate to system behavior in one operational environment, adaptability relates to pairs of operational environments. Performance and robustness of a system behavior in one environment could both be described with a number, but the effort mentioned in the definition of adaptability cannot be described by a number. Effort could be documented by descriptions of the following:

- How difficult is it to make the changes and verify that they achieve the desired result?
- What data, compute, or other resources are needed for the changes?
- How difficult is it for users to tell when changes are needed?

For fundamental and applied research, the research and experiments may be conducted at a single site to evaluate performance and robustness. If the same site is used during different weather or seasons, some evaluation of the adaptability of the research might be made as well. At TRL 4 or greater, there may be systemic experimentation with a demonstration system in different operational environments to understand the adaptability of the system's behaviors. Depending on the research and on the behavior in which it is used, the following could be done to understand adaptability:

- Document what type of changes behaviors (or algorithms) need, including the process and how the researchers determine that the changes worked. This is not intended to include debugging, but only the modification of behaviors that are correctly implemented.

- Document how much data, compute, time, or other resources are needed for the changes. For example, the compute, data requirements, and training time needed to retrain object detection.
- Document how researchers know when changes are needed, and what, if anything, causes them to feel confident the changes will work.

While systemic experimentation to understand autonomous behavior of adaptability is not usually done at the TRL levels at which the DEVCOM Army Research Laboratory conducts research, we can provide an example in which ARL did perform such experimentation for a customer. In 2003, ARL undertook a series of assessments of autonomous mobility technology at the TRL 6 level (Bodt and Camden 2004) using the experimental unmanned vehicle (XUV) shown in Figure 9 to evaluate autonomous mobility across operational environments. Performance at autonomous mobility was measured by how far and at what average speed the vehicle drove autonomously. Adaptability was studied by selecting different operational environments, which varied between day and night, terrain type (wooded, arid, and urban), and the complexity of the course through which the robot had to navigate autonomously. Under each combination of conditions, the performance metrics were recorded and compared across domains. The effort it took to modify the system behavior between environments, the processes for doing so, and the methods of verifying that the correct changes were made were not recorded. These experiments were intended as an assessment of the prototype XUV in a relevant environment, as is required at TRL 6 (Office of the Executive Director for Systems Engineering and Architecture 2023).



Figure 9. Experimental unmanned vehicle (XUV).

7. Conclusion

The Army needs justifiable confidence that systems will successfully perform their autonomous functions when employed correctly, with appropriate human supervision, in their intended operational environment, and that the risks associated with those functions are correctly understood. Developing such confidence requires

- understanding the function and how it will be used,
- identifying the properties of an operational environment that affect performance and how to measure performance,
- understanding the robustness of performance in an operational environment,
- knowing how to adapt the system to perform in different operational environments,
- providing appropriate human supervision, and
- understanding the risks associated with the system's autonomous functions.

This report has provided examples of how low TRL experimentation can assess the value of research and contribute to developing justifiable confidence.

- Section 2 gave examples of research and applications selected to demonstrate it, with Section 3 identifying behaviors that could be observed to assess the application.
- Section 4 provided examples of how performance was measured.
- Section 5 described experiments for measuring robustness of performance.
- Section 6 provided an example of experiments evaluating adaptability.

We recognize human supervision and system risk assessment as also being important, but those topics are not within the scope of our proposed framework. Finally, for those wanting to use this framework, the Appendix contains questions that experimenters can ask when assessing research, which can be used as the basis for developing processes and procedures for assessing low TRL autonomy.

8. References

- Allyn D. The U.S. Army robotic and autonomous systems strategy. Army Training and Doctrine Command (US); 2017 Mar.
- Bodt B, Camden R. Technology readiness level six and autonomous mobility. Proc. SPIE 5422, Unmanned Ground Vehicle Technology VI. 2004 Sep.
- Boularias A, Duvallet F, Oh J, Stentz A. Learning to ground spatial relations for outdoor robot navigation. Proc. ICRA; 2015.
- Childers et al. US Army Research Laboratory robotics collaborative technology alliance 2014 capstone experiment. Army Research Laboratory (US); 2016 July. Report No.: ARL-TR-7729.
- Headquarters, Department of the Army. Soldier's manual of common tasks. Warrior skills level 1. Headquarters, Department of the Army; 2009 June. Soldier Training Publication No.: STP-21-1-SMCT.
- Hicks K. DoD Directive 3000.09 autonomy in weapon systems. 2023 Jan. <https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodd/300009p.PDF?ver=e0YrG458bVDI3-oyAOJjOw%3d%3d>
- Huang H, editor. Autonomy levels for unmanned systems (ALFUS) framework, volume I: Terminology, version 2.0. National Institute of Standards and Technology; 2008 Oct. NIST Special Publication 1011-I-2.0.
- IEEE approved draft standard for robot task representation. IEEE P1872.1/D5, 2023 Oct 19. pp. 1–35. <https://ieeexplore.ieee.org/document/10289660>
- Lennon C et al. Performance evaluation of a semantic perception classifier. Army Research Laboratory (US); 2013 Sep. Report No.: ARL-TR-6653.
- Lennon C et al. An integrated assessment of progress in robotic perception and semantic navigation. Army Research Laboratory (US); 2015a. Report No.: ARL-TR-7443.
- Lennon C et al. Assessment of navigation using a hybrid cognitive/metric world model. Army Research Laboratory (US); 2015b Jan. Report No.: ARL-TR-7175.
- Office of the Deputy Assistant Secretary of Defense Systems Engineering. Department of Defense reliability and maintainability engineering management body of knowledge. Department of Defense (US); 2018 Aug. <https://www.cto.mil/wp-content/uploads/2023/06/RMBoK-2018.pdf>

- Office of the Deputy Director for Engineering. Systems engineering guidebook. Department of Defense (US); 2022 Feb. https://ac.cto.mil/wp-content/uploads/2022/02/Systems-Eng-Guidebook_Feb2022-Cleared-slp.pdf
- Office of the Executive Director for Systems Engineering and Architecture. Technology readiness assessment guidebook. Department of Defense (US); 2023 June. <https://www.cto.mil/wp-content/uploads/2023/07/TRA-Guide-Jun2023.pdf>
- Office of the Under Secretary of Defense for Research and Engineering. USD(R&E) technology vision for an era of competition [memorandum]. 2022 Feb 1. https://www.cto.mil/wp-content/uploads/2022/02/usdre_strategic_vision_critical_tech_areas.pdf
- Oh J et al. Integrated intelligence for human–robot teams. 2016 International Symposium on Experimental Robotics. Springer International Publishing; 2017.
- Rinehart DJ, Knight JC, Rowanhill J. Understanding what is means for assurance cases to “work.” National Aeronautics and Space Administration; 2017. Report No.: NASA/CR-2017219582.
- Space & Missile Systems Center. Systems engineering primer and handbook. 2nd edition. Air Force (US); 2004 Jan 15. p. 8.
- U.S. Army Publishing Directorate. Training and evaluation outline report. Army Publishing Directorate (US); 2024 Feb. Task No.: 17-PLT-4011. <https://rdl.train.army.mil/catalog-ws/view/100.ATSC/44443759-795A-4DDF-8985-0FA707B87873-1491925700311/report.pdf>
- Zhu M, Atanasov N, Pappas GJ, Daniilidis K. Active deformable part models inference. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T, editors. Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8695. Springer.

Appendix. Questions to Ask When Planning Autonomy Assessments

This appendix contains some questions which may elicit information that is helpful for planning an assessment. The research project that is the subject of the experiment is referred to as a “component” when the question refers to integrating it into a demonstration system. The term “scenario” refers to a general set of conditions which would be the basis for an experiment. For example, if a task is mapping a structure, the scenario might be to map the exterior of a building while avoiding being seen by an adversary at a given position. The general scenario has military relevance and could be translated into an experiment in different operational environments. The questions are categorized by the steps of the framework, with a final set of questions relating to the integration of the component into the demonstration system.

Identifying a behavior and task:

What behavior(s) does your research (component) enable on a platform?

What types of a task can be accomplished with this behavior?

What aspects and features should be included to make the scenario relevant for the intended customer?

Are there properties of your component that restrict the scenarios in which it can be used?

Is there subsequent research you plan based on this? If so, what is it, and what types of task and scenario would allow for comparison of your current research with that follow on research?

Measuring performance:

At the task suggested, what would constitute success or failure for your research (component) (e.g., plan produced in 5 s)?

What would constitute success or failure for the system using your component in the behavior (e.g., the robot navigated to the goal point without the user taking control)?

Can success be measured by something more refined than a simple pass/fail or yes/no (e.g., distance from the goal rather than reaching the goal)?

What data do you need to save in order to measure performance, or for other experiment-related purposes, and what data are you confident you will not need?

Is there a baseline level of performance at this task against which you can measure yourself?

If the above cannot be done, what alternative behaviors might yours be compared against? What does exist that you can compare your research to?

Measuring Robustness

What properties of the environment are you confident will (or will not) affect the behavior that your component enables?

For properties you are not confident in, which might most plausibly change the behavior?

By effects, we mean the effect of the property on a system using your component. Of the properties above, which do you already know the effects of, which are you uncertain of the effects of, and which are you most interested in learning the effects of?

Are there properties of the robot/system you intend to use for the experiment that make it more or less likely to fail separately from the performance of your component?

Evaluating adaptability

Have you tried using your research for this behavior in other environments? If so, how did it perform?

When you conduct demonstrations in different environments, do you need to make changes to the robot or the environment? If so, what changes?

How do you know when changes are needed, and how do you know if your changes worked?

Human supervision

Can a user intervene to make changes to the behavior while it is running, or view the proposed behavior and stop or modify it without restarting? If so, via what means?

Can the user stop the behavior for safety, and if so, can they intervene (e.g., take control) and then let the behavior resume afterwards?

Are there human subjects involved, and if so, have you followed the institutional review process?

Demonstration system

What platform or supporting technology is needed for your component to be used at the task you have chosen?

How do failures or errors of components providing input your component affect your component? Here we do not mean bugs, but errors that might occur even on a mature system, like small localization errors or classification errors, or loss of GPS signal.

How does the accuracy of input (e.g., GPS) affect your component?

What research-related failure modes might your component have, and how might they affect the system? Here we do not mean bugs, but failures inherent to any implementation of the research (e.g., class confusion).

How can failures of input to your component be controlled to show best possible performance?

Are there properties of your component that should be tested in simulation, e.g., for safety or because of the number of repetitions? If so, can selected results from the simulation be validated in reality?

Can simulation be used to ensure the component works properly?

If testing in simulation, can a realistic amount of input error be introduced to show realistic performance? By realistic input error, we mean error that might occur after integration into a system.

When you integrate the component onto a robot, identify tests you can do to ensure:

- a) Input flows to your component works properly.
- b) Your component produces good output given good input.
- c) You are saving all data you need.
- d) System components relying on yours use the output of your component correctly.

Can you come up with a short and simple test that shows that the physical system works and does everything mentioned in a–d?

List of Symbols, Abbreviations, and Acronyms

AI	artificial intelligence
ARL	Army Research Laboratory
DEVCOM	U.S. Army Combat Capabilities Development Command
DoD	Department of Defense
DPM	deformable parts model
GPS	global positioning system
IOU	intersection over union
RCTA	Robotics Collaborative Technology Alliance
TBS	tactical behavior specification
TRL	technology readiness level
XUV	experimental unmanned vehicle

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

1 DEVCOM ARL
(PDF) FCDD RLB CI
TECH LIB