Natural, Multi-modal Interfaces for Unmanned Systems

Glenn Taylor(✉)

Soar Technology, Ann Arbor, MI, USA
glenn@soartech.com

Abstract. The prospect of using unmanned systems in dull, dirty, or dangerous jobs to save work or even lives has drawn increasing attention in the DoD. Unmanned ground vehicles are used in theatre to get views into buildings or to destroy suspected IEDs. Unmanned air vehicles are used to get views over the next hill or to deliver munitions on targets thousands of miles away. While the automation and sensing capabilities have increased, interaction with these systems is still fairly rudimentary. Deployed systems typically use tele-operation or waypoint control, in some cases requiring operators to carry heavy operator control units. These approaches place a high burden on the operator in terms of the added weight and the constant attention required to operate the systems. In fact, many of these systems require more than a single operator to control a single platform, which increases the cost and logistics of using them. This paper describes natural, multi-modal interfaces as an alternative to the current state of the practice in controlling unmanned systems, with the goal of leveraging how people already communicate with each other in order to reduce the physical and cognitive burdens of interacting with unmanned systems. We describe approaches and challenges in designing, building, and evaluating natural interfaces. We present our Smart Interaction Device (SID) as an example natural interface for interaction with unmanned systems, and highlight some use cases we have applied it to in the air and ground domains.

Keywords: Unmanned systems · Natural interaction · Multi-modal interface · Dialogue systems

1 Introduction

The prospect of using unmanned systems in dull, dirty, or dangerous jobs to save work or even lives has drawn increasing attention in the DoD. Unmanned ground vehicles (UGVs) are used in theatre to get views into dangerous buildings or to destroy suspected improvised explosive devices. Unmanned air vehicles (UAVs) are used to get views over the next hill or to deliver munitions on targets thousands of miles away. While the platform and autonomy capabilities have increased, interaction with unmanned systems is still fairly rudimentary. (Throughout this paper, we will abbreviate 'unmanned vehicle' as UxV to generalize across different domains, and will use 'unmanned vehicle' and 'unmanned system' interchangeably.) Deployed ground systems use tele-operation for control, making them essentially remotely controlled vehicles with cameras and

D. Harris (Ed.): EPCE 2017, Part II, LNAI 10276, pp. 145–158, 2017.
DOI: 10.1007/978-3-319-58475-1_11

actuators. Deployed UAVs typically include some autonomy for waypoint or route following, along with control loops to keep the aircraft aloft. Because the user's interaction with these systems is at a fairly low level (tele-operation or low-level commands to set waypoints), there is a high burden on the operator. In fact, many of these systems require more than a single operator to control a single platform, which increases the cost and logistics complexity of using the platforms. Much of this has to do with lack of autonomy – these deployed systems lack the capabilities to perceive or navigate their environment without running into problems very quickly. The result is that human operators are relied on to control much of UxV movement and sensing.

The DoD expects that advances in autonomy will allow unmanned ground vehicles to move along with a squad to carry their extra gear, or swarms of unmanned air vehicles to quickly search an area. Even further, the DoD is looking to have unmanned systems act as teammates rather than tools. Greater autonomy can help expand the uses of these systems, making the platforms more capable and reducing some burden on the operator, but there is still the need for an operator to communicate tasking or other information to the UxV in an efficient and effective manner. An operator who is performing other tasks will not be able to attend fully to one or more UxVs, either to drive or monitor them. To be more useful in operations, communicating with these systems needs to be as easy and as natural as communicating with a teammate or a subordinate.

This paper describes natural, multi-modal interfaces as an alternative to the current state of the practice in controlling unmanned systems, with the goal of leveraging how people already communicate with each other in order to reduce the physical and cognitive burdens of interacting with UxVs. We describe approaches and challenges in designing, building, and evaluating natural interfaces. We also describe our Smart Interaction Device (SID) as an example natural interface for interaction with unmanned systems, and highlight some use cases we have applied it to in the air and ground domains.

2 Natural Interaction

Instead of using devices such as joysticks, game controllers, or keyboards as ways to interact with robots, an alternative is to consider the ways in which people interact with each other. By allowing users to interact in ways that are familiar to them, without having to learn new devices, the expectation is that users should be able to learn how to interact with these systems more quickly, and that they would find the user interfaces more intuitive and familiar in general. In some cases, they may also free a user from having to carry additional equipment (such as an operator control unit or a laptop) to communicate with unmanned systems.

People use many modes naturally. Speech is obviously a common mode of interaction, and this includes the words people speak as well as the prosodic elements such as pitch and intonation as ways to carry meaning. Gestures, too, are quite natural to people, and can include movements of the hands, arms, fingers, and other body parts, and these motions or held shapes are meant to convey specific meaning. In military domains especially, drawing maps and sketching *on* maps are common and natural ways to interact. Gaze, body language, and even the distance people stand from each other are ways in which people naturally communicate.

A person's task, and the environment in which that task is performed, often influences the modes of interaction. If the environment is noisy, or the task demands absolute quiet, then spoken communication may not be useful or desirable. If two participants cannot see each other, gesture is not effective. Resource limitations also play a role – if one participant has her hands full, then gesture will not be possible.

Having multiple modes to choose from can be useful in different ways. In some cases, the same information can be conveyed in different modes – e.g., saying goodbye or waving goodbye. This redundancy allows someone to choose how to communicate in the moment without loss of information. Redundancy also allows someone to use two modes simultaneously to make sure the message is received. On the other hand, not all modes are equally capable or as effective at conveying the same information. For example, giving a verbal description of an object may be less efficient than simply pointing at it. Often people will mix modes to communicate effectively – e.g., verbally saying a destination while also tracing a path on a map. This mixing of complementary modes can be more efficient than trying using just one [1].

Besides the mode of interaction (speech, gesture, etc.), another facet of natural human interaction is dialogue – communicating over time, with contributions from two or more participants. Dialogue can be used to ensure that the participants have a common ground for what's being communicated [2]. Dialogue can also be used to overcome failures in communication, where one person may ask for clarification if something is not clear, or if some information was missed because of a noisy environment. Dialogue also aids in efficient communication, where the dialogue itself serves as context for understanding references that someone might use in the conversation. A sign that dialogue is an aspect of natural interaction is that a participant can get frustrated when another participant fails to follow conversational rules [2, 3].

Naturalness in an interface is not defined solely in terms of the modes of interaction. A gesture-based interface that forces the user to place her arms in uncomfortable positions is not really natural. A speech-based interface for a robot that consists only of the verbal commands, *forward, turn-left, turn-right, and stop* might use a natural mode, but the language itself is possibly not very natural to the task. To make robot to do anything useful would take a great deal of effort and time. One goal of many human-system interfaces is *supervisory control*, which Sheridan describes as having three requirements: some level of *system autonomy*, user *situation awareness* of the system's behavior, and *high-level interaction* [4]. A robot that still requires low-level control will never let the user achieve supervisory control because she will be too busy driving it inch by inch.

3 Related Work

The last few years have seen a surge in natural interaction in commercial products. Speech-enabled assistants such as Amazon Echo®, Apple Siri®, Google Now®, and Microsoft Cortana® aim to make tasks such as playing music, finding directions, or searching the internet easier. While their speech recognition and language understanding capabilities are impressive, they are still fairly limited in the kinds of interactions they support, and typically don't give more than one-shot answers to questions or requests. Spoken interfaces to robotic systems have appeared in research systems for a long time. Fairly simple speech interfaces for robotic systems have also appeared in consumer-oriented robots, including the recent Cozmo® from Anki.

Researchers have investigated gesture recognition for some time, with impressive work being done in domains such as carrier deck hand-and-arm signals [5] and American Sign Language [6]. The recent introduction of relatively inexpensive sensor systems such as Microsoft Kinect®, Leap Motion®, and Thalmic Labs' Myo® arm-band have spawned a new wave of interest in gesture recognition as an input modality. Gesture-based interfaces have also begun to make their way into other commercial products. Some of the augmented reality headsets that are now available, such as Microsoft HoloLens® and Atheer Air®, include a handful of simple hand gestures as ways to select objects or invoke menus, in part of out necessity since these systems do not come with typical devices like keyboard and mouse.

Body posture, facial expressions, and gaze as user interface controls largely remain in the realm of research labs. Likewise, while mixing modes such as speech and gesture together is a common trait of human interaction, it is still largely a research endeavor. There are some exceptions, of course. For example, gaze-based control has found a place in user interfaces for physically disabled users, using gaze to for a wide variety of computer tasks [7]. Microsoft's HoloLens® allows a user to move a cursor with his head (a proxy for gaze), and, when the cursor is over a button, the user can say "select" to press the button.

Dialogue is another dimension of natural human interaction, and a number of dialogue systems have been researched, from personal assistants that help make reservations [8] to chat-bots for question-answering [9]. While dialogue systems exist in commercial products, most are quite limited. They take the form of customer support systems, such as automated phone systems for airlines or banks that essentially a user through a menu via speech, or in the form online chat-based help systems that help screen questions before a human operator takes over.

Perhaps the closest to the work we describe here is that of the WITAS system [10], which includes mixed modes and simple dialogues for natural interaction with unmanned systems. Our work is also inspired by the QuickSet system [11, 12], in which users could engage in multi-modal dialogues to construct military simulation scenarios. Our work also squarely fits into the area of supervisory control [4], in which we aim to supplement the UxV's autonomy by raising the level of interaction with them, as well as helping the operator maintain awareness about the system's behavior.

4 Designing for Natural Interaction

As might be gleaned from the earlier discussion of natural interaction, how interaction happens in practice can be quite involved, and is affected by the task, the environment, the reliability of the communication channel, the skill of the participants, and their choices of how best to communicate. If we are to design natural user interfaces for people to interact with unmanned systems, we need to understand how natural interaction happens in context, and provide for the kind of flexibility and resiliency that is afforded by natural interaction in those contexts.

4.1 Discovering Interaction in Context

Designing for natural interaction typically starts as a discovery process. In many cases, the UxV use case involves using the system in a similar role that a person would play (for example, a large UGV playing the role of rear security in a dismounted squad). In these cases, we can use the person-to-person analog as the model for how participants interact. If we limit the scope of interaction with unmanned platforms to task-oriented communication (as opposed to, for example, small talk), we can start by studying how people communicate naturally when performing the task. Standard task analysis methods can provide a framework for understanding the task itself and even the cognitive elements of the task [13], but these methods do not typically focus on communicative aspects of a task. Instead, methods such as interaction analysis [14] focus on the interaction itself. From the interaction perspective, we need to understand a number of elements. What language(s) do participants use to communicate? What modalities do they use, and under what conditions? What do they talk about? Why do they communicate? What do they do when communication breaks down? How does the context change the communication? Consider some examples. If a squad needed to maintain quiet while moving to occupy a position, how would the squad leader tell the squad to halt? If a team were geographically dispersed, how would a leader communicate a new mission to a remote subordinate unit? Understanding these elements is the first step in building a user interface when an unmanned system is one of the teammates.

In other cases, a UxV may be used in a new task, or offer a new capability, for which there is no real analogue in that use context, or for which there are only typical "artificial" interface equivalents. For example, dismounted infantry units might have access to small UAVs, but the only interface to them may be through joystick or GUI control on a laptop. If we want to introduce natural modes of interaction, we must discover how these users might *want* to interact with these systems.

Wizard of Oz (WoZ) studies are a standard method to discover new types of interactions and user preferences [15]. For example, suppose a user didn't have access to a keyboard and mouse, but instead had a map, a pen and a microphone, how would that user *want* to interact an unmanned system for different tasks? In Wizard of Oz studies, user interface mockups are built but are not connected to a real system, and instead a "wizard behind the curtain" makes it seem like the user interface is having an effect. These studies let researchers design hypothetical interfaces to explore different ways of interacting without having to also implement the real system. One challenge in designing WoZ studies is getting users outside of their mentality of using a typical user interface, which might be a practiced way of interacting with the system. Additionally, standard graphical interfaces often provide visual hints for the kinds of inputs the system can accept, which helps the user along. Natural interfaces do not often provide these hints, and, in their absence, users can be at a loss for how to interact without some kind of prompting or training. Another challenge in Wizard of Oz studies is giving the "wizard" all the right tools and interfaces to implement the expected behavior of the system; without good supporting tools, the wizard's job can be quite frantic behind the curtain.

In these latter cases where there is no prescribed natural way of interacting with a UxV, it might also be useful to look for other analogs. For example, a typical UAV operator in a small infantry unit might be accustomed to flying the vehicle directly with a joystick in order to get the vehicle in position to see inspect an area. However, the unit leader likely gave a command to the UAV operator in some natural way. That operator's job includes translating the unit leader's commands into commands for the UAV using the joystick. To make the operator's interaction more natural, we might look to the leader-operator interactions as a way to understand how the operator might want to interact with the vehicle. This may only be a starting point, however; the UAV operator brings particular skills in translating the leader's commands into UAV commands, and there may be other types of inputs the operator would need to give to the system, or other types of feedback the operator would need from the system, for effective operation.

4.2 Designing for Failure

Accounting for the practical differences between truly natural interactions as humans perform them, and the limitations of today's sensors and recognition systems, is one of the challenges of actually building these kinds of systems. For example, speech recognizers, while getting very good, are still not wholly reliable. Often one compromise is to limit the interaction language to account for these deficiencies – for example, by limiting the lexicon or grammar to a particular sub-language. However, this can have the negative effect of putting increased burden on the users to "unlearn" what might be natural for them, and then learn the limited inputs the system can actually recognize, with regular mistakes when they fall back to their typical language.

Another approach to this problem is to design for failure from the beginning. When two people communicate, one might fail to perceive an input, fail to recognize the input, or fail to understand the meaning of the input, for any number of reasons. Automatic recognition and understanding systems will also never be 100% reliable, and so the system must be designed in such a way as to accommodate these different types of failure.

Giving the user insight into what was recognized before the system acts on the input is one approach. Texting apps on smart phones often take this approach: when speaking instead of typing in a text window, the phone will often show the results of the speech recognizer before sending the text. This gives the user an opportunity to correct any errors manually before accepting the result. Of course, there is a cost – the message does not get sent out very quickly if the user reviews the message, and has to fix it, it beforehand.

Providing feedback to the user when recognition succeeds can also be helpful in maintaining the user's awareness about what the system is doing. If the direct result of the input is not obvious (for example, the unmanned system is not visible), or the effect will not happen for some period of time, the user may have no way of knowing what input was recognized or what command was issued. Making the user aware at least gives her the opportunity to make an adjustment. Whether the user corrects the recognition before the recognized input is acted on or after the user notices something went wrong in the system is akin to a *management by consent* versus *management by exception* approach to supervisory control [16].

This simple input-with-feedback exchange is an example of a type of dialogue with the user, with the intent of giving the user some awareness of what the system is doing. In addition to helping keep participants in sync, dialogue can help overcome failures in communication. Unlike the texting example above, there is no chance in regular spoken interaction for a person to see what another participant will recognize or understand. Dialogues might include asking the speaker to repeat when something was not heard, or asking for clarification when something is not understood. These different types of dialogue give the participants a chance to recover when something goes wrong. In building natural interfaces for unmanned systems, giving the system the ability to exercise these dialogue strategies can help keep the user aware of the system is doing as well as help overcome the inevitable failures in communication.

5 Smart Interaction Device

Over the last several years, we have been developing a system we call the Smart Interaction Device (SID), a natural multi-modal, dialogue-based interface to help users interact with unmanned systems [17, 18]. SID acts as a facilitator between a user and unmanned systems, allowing the user to communicate in familiar terms, and providing task-relevant information back to the user. SID allows for multiple input modes, including redundant modes and complementary modes, and communicates in multiple modes back to the user. SID frames all interaction with the user as a dialogue, working within familiar dialogue protocols to facilitate shared understanding and to overcome communication failures, and using the dialogue itself as context for understanding the user's inputs.

Figure 1 shows a high-level depiction of SID's architecture. SID Core contains three domain-independent modules that help facilitate the user's interaction with the unmanned platforms. The Dialogue Management component maintains the dialogue state and uses it as context to understand user inputs and provide feedback to the user. The Situation Awareness component maintains awareness of the unmanned platforms (progress and problems) and generates alerts and notifications in user terms rather than platform terms. The Planning and Execution module translates user-level commands into commands that the unmanned systems can understand. The processes within these modules can be tailored to specific tasks and domains.

Two implementation-specific modules are also part of the architecture. On the left side of SID Core is the Multi-Modal I/O component that fuses together multi-modal user inputs and also generates multi-modal outputs. Between the user and the system (and not depicted) is a range of input devices, sensors, and recognizers that take the raw inputs and generate semantic parses from them. Depending on the use case, these include recognizers for speech, gesture, and sketch, along with associated language parsers, the outputs of which are passed to the input fusion process, which generates

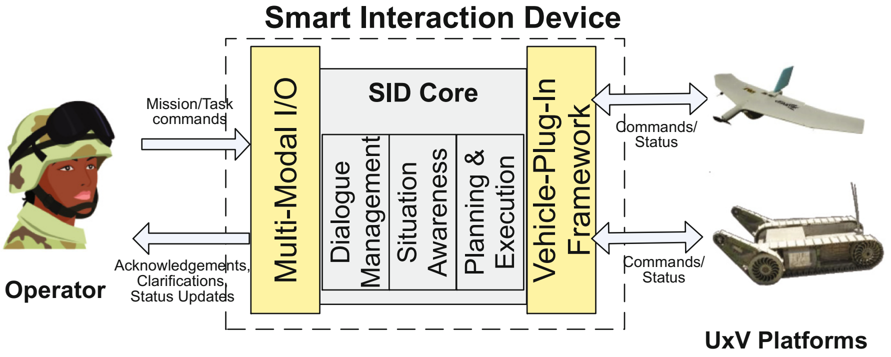


Fig. 1. High-level architecture of the Smart Interaction Device

hypotheses about the user's intent. Likewise, there are a variety of output devices and displays that portray information to the user: speech generation, video, text, graphics, and haptic devices. On the right side of SID Core is a plugin framework for interfacing directly with a range of unmanned systems.

An important aspect of SID, and other user interfaces we have built for unmanned systems, is providing transparency into the behavior of the UxV. This relies in part on the systems themselves providing information back about their progress and status, which SID then translates into user terms. In some cases, there may be standard reports that a user expects and which is built into the domain-specific application – e.g., the robot reached a destination or spotted a threat. SID also allows the user to query status on demand, such as by asking "Robot, what are you doing?" or "Where are you going?" This information may also be on a graphical display, but it may also be easier for the user to ask for status verbally rather than look at a screen. Providing natural ways of querying the system and conveying information back to the user in different ways is also important to keep the user informed throughout the varying situations.

The architecture depicted above describes a general framework that we have instantiated for several use cases. The core behavior of the system has largely remained intact, but we have extended it to apply to a particular domain and task. Likewise, the specific I/O devices and platforms typically vary per domain and task, so we have built particular adapters to for these devices and platforms. In this section, we describe two different applications of SID, which include different use cases, different I/O devices, and different interaction languages.

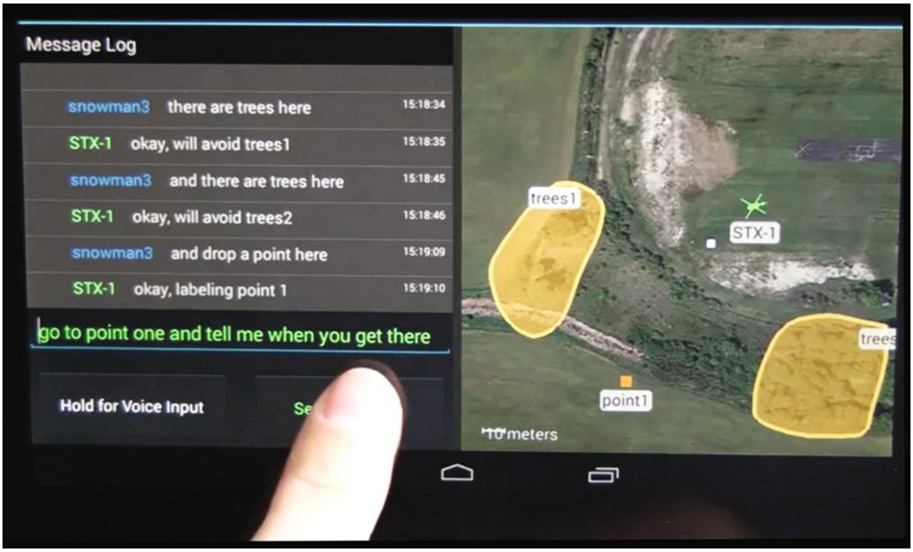
5.1 Remote Operations

There are many use cases in which the operator and the unmanned systems may not be co-located, and in fact may be tens or thousands of miles away from each other. Even many small UAV operations are beyond line of sight, where the operator needs to rely on some external means to get a sense of the vehicle location and current task. In these types of operations, a map is a standard tool that people use to coordinate behavior.

Where both participants have an identical map that each can see and refer to, they use the map for coordination and situation awareness. The map can provide visual landmarks that can be used as reference points for tasking ("Fly to the bend in the river") or for reporting location ("I just reached the bend in the river"). Map-based tasks are especially rich with mixed-mode interactions [19] where, for example, one participant can speak some information while drawing on the map to convey other information (e.g., saying, "Go here" while pointing to the map).

We have implemented in SID these kinds of interactions using a tablet-based map to show UxV positions and tasks, and to allow the user to create new reference objects (e.g., waypoints, routes) and tasks that refer to them (e.g., "Follow this route" while sketching the route). Map-based interactions such as drawing or pointing are through touch on the screen. Speech is through a microphone on the tablet or worn by the user. Once these reference objects are created and named, the user can refer to them later – e.g., "Follow route blue." A Wizard of Oz study early in the design process helped us identify the kinds of interactions users would want supported with this kind of system [18]. The study showed that pointing gestures as well simple sketches of points, routes and areas could account for over around 85% of the drawn inputs. This allowed us to build an effective interface using only a few simple drawing elements.

An example of one such implementation is shown in Fig. 2 below. On the upper left side is a log of the conversation between the user and the system, which gives the user a sense of what they have been conversing about. In this case, we also display what was recognized last, and the user must deliberately send the message after checking it, with the ability to manually change it if something was misrecognized. On the right side is a display that shows the position of the aircraft on the map (labeled as STX-1), along with a waypoint and two areas that have been constructed by the user.


Message Log

snowman3 there are trees here 15:18:34

STX-1 okay, will avoid trees1 15:18:35

snowman3 and there are trees here 15:18:45

STX-1 okay, will avoid trees2 15:18:46

snowman3 and drop a point here 15:19:09

STX-1 okay, labeling point 1 15:19:10

go to point one and tell me when you get there

Fig. 2. A user interaction with a UAV with a speech-and-sketch map-based interface

Dialogue plays a role when the user gives ambiguous commands, refers to reference objects that don't exist, or when recognition fails for some reason. If the system failed to recognize the input, it will generically ask the user to repeat. If the user was ambiguous in the input, the system will ask for the specific information that helps complete the command (e.g., "Where do you want me to go?"). The user can reply with a brief answer to fill in the missing details, even using multiple modes in the reply (e.g., "Here" while pointing on the map).

5.2 Proximal Operations

Another use case is when the operator and the UxV are in close proximity such that they can see each other. An analogous use case in human operations would be a dismounted infantry squad on patrol. In this case, a large "robotic mule" UGV might travel with the squad to help carry extra equipment or to provide some surveillance capabilities. While a map-based interface such as described earlier could be used, one goal is to minimize the equipment that the operator has to carry and to keep his hands free for other tasks. Another goal is to help the warfighter keep his head up, looking at his surroundings instead of at a screen. Speech and gesture as modes help afford this kind of hands-free, head-up interaction, both among the members of the unit and between the operator and the UGV. In some cases, both speech and gesture are redundant modes: a command might be given using speech or gesture, such as in a "stop" command. The kinds of hand and arm signals in this domain are meant to be seen from a distance, so are fairly gross in detail and thus cannot convey a great deal of information (this is in contrast to, for example, American Sign Language, which is a complete language in and of itself, but is generally meant for face-to-face interaction).

We have implemented in SID these interactions using a few different technologies, most recently using a worn device as a means to capture both spoken inputs using a microphone and gesture inputs using the on-board inertial measurement unit (IMU) (see [20] for details). Redundant speech and gesture inputs cover basic tele-operation kinds of behaviors (forward, backward, left, right, stop), but also more autonomous behaviors such as following the user. Other speech commands that do not have gesture equivalents include taking control of the vehicle, releasing control, defining named rally points, and routes based on the position or movement of the UGV, maneuvering to those points or along those routes, and maneuvering in finer granularity such as turning a specific amount. Figure 3 shows some gesture interactions with a large UGV.

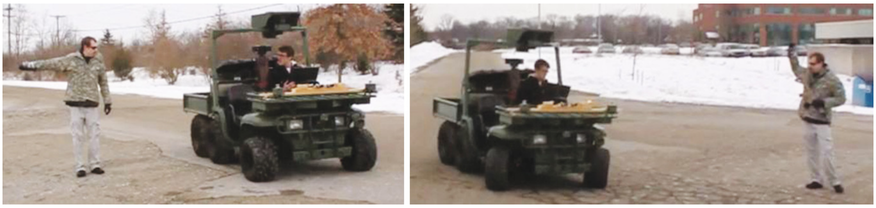


Fig. 3. Examples of gestures being given to control a large UGV (person on board the vehicle is a safety operator.)

6 Evaluation Approaches

There are many different approaches to user interface evaluations, most of which apply just as well to human-robot interfaces [21]. The inclusion of natural interaction as part of a user interface introduces new facets to the evaluation, and a few new challenges.

One factor that makes natural interfaces unique is that they can fail to recognize or understand the user. It is therefore important to measure the performance of input recognizers and related components in a wide range of conditions. Some of these conditions are environmental. For example, speech recognition is susceptible to surrounding noise, and even different types of noise may affect recognition differently (e.g., constant white noise, versus periodic, versus other people talking in the background). Different lighting conditions or other objects in the scene can affect vision-based gesture recognition. It is also important to evaluate the system with a wide range of users – ideally those who are good proxies for the target user. Speech recognizers may perform differently with different voices (e.g., from gender or age of the user, or accents), or with simultaneous user activity (e.g., running while speaking), or emotional state. Gesture recognizers have to account for different body shapes and sizes and different ways in which people might make a gesture. Each individual recognizer can be evaluated in isolation apart from other system-wide evaluation. While it does not tell the whole story about the user interface, it is an important first step to get a sense of how the system will perform and how natural the interface is.

User interface evaluations in general are often comparative, assessing one against another, and natural interfaces are no different. In comparative evaluations, learnability is often an important metric: how quickly someone can become a proficient user. One motivation behind natural interfaces is that they are meant to be closer to how people communicate, so should be easier to learn. Such hypotheses need to be tested for particular interfaces. If the specific inputs are artificial constructs that use natural modes but where the input language is invented or limited in some way that requires training, then learnability may suffer. For example, speech interfaces often have a fixed input language that the user has to get just right, otherwise the system fails to recognize the inputs. (Frequent and common deviations from this accepted language can be an opportunity to learn how the language coverage needs to be expanded.)

Other typical comparative metrics include the amount of time it takes to perform a task, and the amount of work for each task, in each of the two systems. In terms of natural interfaces, where recognition can fail, it is important to capture the cost of that recognition failure, both in terms of total time and in the amount of work (and frustration) it causes the user. Standard measures like NASA TLX [22] can be used to measure perceived workload, and progress on secondary tasks can serve as a proxy for objective workload. As with human communication failures, this failure can happen at multiple levels – in recognizing the raw inputs or in understanding the user in context – and understanding how and why this failure happens is important. It may also be important to measure how aware the user is of system failures; some failures may not be recognized, which can cause more problems further down the line.

Dialogue as part of a natural interface is also an important element to consider in the evaluation, and multiple approaches have been used to evaluate the utility, impact, or even the naturalness of dialogue [23]. Where dialogue is used to help keep the user informed, it is important to measure that, such as by using standard situation awareness measures [24, 25]. Where dialogue is used to help recover from errors, it is important to measure the effect of the system on that recovery. Dialogue can also help make the interaction more efficient, leveraging the dialogue as context to help understand brief or otherwise ambiguous inputs. It can also be important to evaluate different dialogue strategies that might engage the user in different ways. In these cases, the study conditions would include isolating different strategies to measure their effect independently.

We will be exercising many of these aspects of evaluation in an upcoming study evaluating an interface for large robotic mules, with the help of the Army Research Laboratory's Human Research and Engineering Directorate (ARL/HRED) Field Element, and the participation of Soldiers at Fort Benning. There are several goals to the study. One is to look at user preferences between gesture and speech in cases where the input modes are redundant. We plan to put the interface through three trials on pre-defined courses: in one case, participants will use speech only; in another, gesture only; and in the third, they can use either mode interchangeably. We aim to understand not overall preference for a mode, but the contexts in which one is preferred over the other. Another goal is to assess the utility of some inputs that only have speech forms, such as defining routes and rally points, and tasking the vehicle using those control measures. In addition to quantitative measures of the effectiveness of different interactions, we also simply want to understand how Soldiers use the system in practice, based on their training and operational experience. This includes looking at the coverage of interactions we have so far and how they support Soldier tasks. This also includes understanding how the particular fit and form of the interface (a worn device) works with the way in which they operate. Putting the system in the hands of representative users and getting their feedback from an operational perspective is a critical step in understanding the interface's true value to its intended users.

7 Conclusions

Continuously improving recognition algorithms and the low cost of sensors has driven an increased interest in natural interfaces for computing systems, including for unmanned systems. In this paper, we have described an approach to designing and building natural interfaces for unmanned systems, including how to identify natural interaction in specific use contexts: how people interact today in these cases, or how they might want to interact with unmanned systems in the absence of traditional input devices. We have identified different dimensions of natural interaction, including the use of different modes (e.g., speech and gesture), ways in which use of these modes might be redundant or complementary, and ways in which interactions can become dialogues.

Natural interfaces can help improve the usability of unmanned platforms; however, this requires not just adding speech or sketch to the interface, but also taking into account the context of use. These modes have to be applied in ways that people find natural to a particular task, especially in ways in which they use those modes for the same or similar tasks. In designing theses systems, it is important to keep in mind how limitations in recognition technology can impact the design of the interaction languages themselves, and how these designs can subsequently affect the usability of the system. If the language understood by the system is too contrived or outside of what people use in their daily activity, the "naturalness" of the interface may be compromised and, in fact, make the system harder to use. It is also important to recognize that natural interfaces by – by the nature of natural inputs – will never be completely reliable, and to design the system to account for and recover from those failures. Framing interaction as a dialogue between the user and the system, and implementing dialogue strategies that are familiar to people, is one way to overcome these failures.

Natural interfaces also present some unique challenges for evaluation, for a number of reasons. Since one goal of natural interfaces is to be closer to how people communicate, it is important to design studies to measure this specifically. Because natural interfaces can fail to recognize or understand user input, it is also important to measure those failure rates and the impact they have on the task. Where additional mechanisms are used to mitigate those potential failures, such as shrinking the interaction language artificially or through the use of dialogue, the impact of those mechanisms must also be evaluated.

To illustrate some of these aspects of natural interfaces, from design to evaluation, we have described an example system, the Smart Interaction Device (SID) that incorporates many of the natural interaction features described. In the development of SID, we have applied it to multiple different domains and tasks, which has included integration with many different input devices and recognizers, covering speech, sketch, and gesture. We have also integrated and demonstrated SID with over a dozen different unmanned platforms, both air and ground, including heterogeneous, multi-platform demonstrations.

References

1. Oviatt, S., Coulston, R., Lunsford, R.: When Do We interact multimodally? Cognitive load and multimodal communication patterns. In: ICMI 2004. ACM, State College (2004)
2. Clark, H.H.: Using Language. Cambridge University Press, Cambridge (1996)
3. Grice, P.: Logic and conversation. In: Morgan, J. (ed.) Syntax and Semantics. Academic Press, New York (1975)
4. Sheridan, T.B.: Telerobotics, Automation and Human Supervisory Control. MIT Press, Cambridge (1992)
5. Song, Y., Demirdjian, D., Davis, R.: Tracking body and hands for gesture recognition: NATOPS aircraft handling signals database. In: IEEE International Conference on Automatic Face and Gesture Recognition (FG 2011) (2011)
6. Pansare, J., Gawande, S., Ingle, M.: Real-time static hand gesture recognition for American sign language in complex background. J. Signal Inf. Process. **3**, 364–367 (2012)
7. Hutchinson, T.E., et al.: Human computer interaction using eye-gaze input. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **19**, 1527–1534 (1989)

8. Traum, D.R.: Conversational agency: the TRAINS-93 dialogue manager. In: Twente Workshop on Language Technology 11: Dialogue Management in Natural Language Systems (1996)
9. Quarteroni, S., Manandhar, S.: A chat-bot based interactive question answering system. In: Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, Rovereto, Italy (2007)
10. Lemon, O., et al.: A multi-modal dialogue system for human-robot conversation. In: NAACL (2001)
11. Cohen, P.R., et al.: Quickset: multimodal interaction for distributed applications. In: 5th ACM International Conference on Multi Media. ACM Press (1997)
12. Johnson, M., et al.: Unification-based multimodal integration. In: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics. ACL (1998)
13. Kirwan, B., Ainsworth, L.K. (eds.) A Guide to Task Analysis. Taylor & Francis, London (1992)
14. Jordan, B., Henderson, A.: Interaction analysis: foundations and practice. J. Learn. Sci. **4**(1), 39–103 (1995)
15. Kelley, J.F.: An iterative design methodology for user-friend natural-language office information applications. ACM Trans. Off. Inf. Syst. **2**, 26–41 (1984)
16. Ruff, H.A., Narayanan, S., Draper, M.H.: Human interaction with levels of automation and decision-aid fidelity in the supervisory control of multiple simulated unmanned air vehicles. Presence **11**, 335–351 (2002)
17. Taylor, G., et al.: A multi-modal intelligent user interface for supervisory control of unmanned platforms. In: Collaboration Technologies and Systems Collaborative Robots and Human Robot Interaction Workshop, Denver, CO (2012)
18. Taylor, G., et al.: Multi-modal interaction for UAS control. In: SPIE.DSS, Baltimore, MD, April 2015
19. Cohen, P., McGee, D., Clow, J.: The efficiency of multimodal interaction for a map-based task. In: Applied Natural Language Processing Conference (2000)
20. Taylor, G., et al.: Multi-modal interaction for robotic mules. In: SPIE Defense and Security: Unmanned Systems Technology XIX, Anaheim, CA (2017)
21. Yanco, H.A., Drury, J., Scholtz, J.: Beyond usability evaluation: analysis of human robot interaction at a major robotics competition. J. Hum. Comput. Interact. **19**(1), 117–149 (2004)
22. Hart, S.G., Staveland, L.E.: Development of NASA-TLX (Task Load Index): results of empirical and theoretical research. In: Hancock, P.A., Meshkati, N. (eds.) Human Mental Workload. North Holland, Amsterdam (1988)
23. Paek, T.: Empirical methods for evaluating dialog systems. In: Workshop on Evaluation for Language and Dialogue Systems. ACM (2001)
24. Taylor, R.M.: Situational awareness rating technique (SART): the development of a tool for aircrew systems design. In: Situational Awareness in Aerospace Operations (AGARD-CP-478). NATO-AGARD, Neuilly Sur Seine (1990)
25. Endsley, M.R., Garland, D.J. (eds.) Situation Awareness Analysis and Measurement, p. 383. Lawrence Erlbaum Associates, Mahwah (2000)