

# Semi-Autonomous Human-UAV Interfaces for Fixed-Wing Mini-UAVs

Morgan Quigley  
Brigham Young University  
Provo, UT, USA  
mquigley@byu.edu

Michael A. Goodrich  
Computer Science Department  
Brigham Young University  
Provo, UT, USA  
mike@cs.byu.edu

Randal W. Beard  
Elec. and Computer Eng. Dept.  
Brigham Young University  
Provo, UT, USA  
beard@ee.byu.edu

**Abstract**— We present several human-UAV interfaces that support real-time control of a small semi-autonomous UAV. These interfaces are designed for searching tasks and other missions that typically do not have a precise predetermined flight plan. We present a detailed analysis of a PDA-based interface and describe how our other interfaces relate to this analysis. We then offer quantitative and qualitative performance comparisons of the interfaces, as well as an analysis of their possible real-world applications.

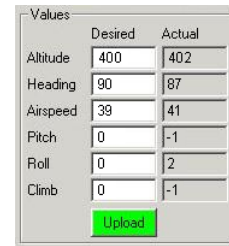
## I. INTRODUCTION

The semi-autonomous fixed-wing mini-UAV is an emerging class of vehicle that has many potential applications, including short-range military reconnaissance, rural search-and-rescue, law enforcement, and any other task that would benefit from a small, low-cost aerial vehicle [1] [2]. This paper is concerned with the human factors associated with small UAVs, focusing on human-UAV interaction issues and their implications to interface design.

Previous work has produced several prototype mini-UAVs which have significant autopilot capabilities [3]. The interfaces discussed in this paper operate the autopilot in a semi-autonomous control mode, where the user supplies a desired altitude or pitch angle, a velocity, and a heading or roll angle. The autopilot seeks to meet the desired flight characteristics of the user, who is able to modify these parameters in real-time via radio modem. Navigation decisions are made either by directly observing the airplane or through its on-board video camera. Although a semi-autonomous interface requires a higher user workload than a fully-autonomous waypoint-based interface, many mini-UAV applications require the operator to make strategic decisions on-the-fly, which reduces the effectiveness of predetermined flight plans.

An interface that requires the user to directly type flight parameters, as shown in Figure 1, is sufficient for some applications, but leaves much to be desired in many others. In many applications, such precision is neither required nor helpful, as users are required to have a solid understanding of appropriate values for each parameter, and the cognitive and physical requirements of typing parameters introduces unnecessary delays into user response time.

Our efforts present an alternative approach, one which trades precision for intuition and speed of response. The goal is an in-the-loop feel that supports efficient task



Values	Desired	Actual
Altitude	400	402
Heading	90	87
Airspeed	39	41
Pitch	0	-1
Roll	0	2
Climb	0	-1

Upload

Fig. 1. A numeric parameter-based UAV interface

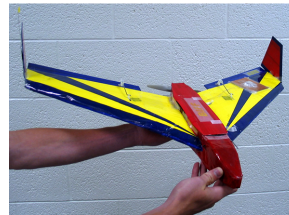


Fig. 2. An in-house fixed-wing semi-autonomous mini-UAV

completion. Although no interface can attempt to control the UAV more precisely than directly typing the desired flight parameters, we claim that such precise control is often unnecessary and burdensome for a mini-UAV. The user is often content with a lower level of precision, so long as the interface is simple, intuitive, and responsive.

We have created several alternative UAV interfaces in which users operate physical controllers or graphical direct-manipulation displays to generate the requisite numerical commands. These interfaces are built using PDAs, full-size computers, a voice-recognition system, a force-feedback attitude joystick, a force-sensing interface using an IBM TrackPoint™, and a novel “physical icon” interaction scheme.

## II. UAV AND AUTOPILOT TECHNOLOGIES

The airframe shown in Figure 2 has a 32” wingspan and is a flying wing design with a fiberglass fuselage and foam wings. The UAV is powered by an electric motor in a push propeller configuration and is hand launched and belly landed. It is actuated by two elevons, with fixed wing tips providing vertical stabilization.

The UAV autopilot is implemented on a single circuit board with a 29 MHz Rabbit microcontroller daughter

board. Sensors include three-axis rate gyros, three-axis accelerometers, absolute and differential pressure sensors, and a standard GPS receiver. The autopilot package weighs 2.25 ounces including the GPS antenna, and the size of the autopilot is roughly 3.5 by 2 by 0.5 inches. Data communication between the airplane and the ground station is accomplished via a low-cost 900 MHz wireless modem, and video is transmitted through an off-the-shelf 2.4 GHz A/V transmitter.

### III. GENERAL INTERFACE CONSIDERATIONS

Human-UAV interfaces must deal with several constraints that are inherent to teleoperation of air vehicles:

- The unstable dynamics of a mini-UAV require the interface to support a significant level of autonomy for the UAV to be accessible to many users.
- Many users have little to no experience flying airplanes, and can be confused and disoriented by their many degrees of freedom.
- If the user loses control of the UAV, it may quickly result in significant damage or destruction of the UAV.
- Since the UAV can fly considerable distances away from its operator, depending on the accessibility and hostility of the environment, the UAV may not be recoverable in the event of a crash.

To deal with these challenges and to facilitate simple and intuitive operation of the UAV, we built upon several effective design principles demonstrated by previous work in human-robot interfaces [4]. The interfaces are designed to clearly present the state of the UAV, produce timely feedback, and provide a straightforward mapping between interface controls and the resultant actions of the UAV.

### IV. PDA INTERFACE DESIGN AND ANALYSIS

Although human-robot interfaces can be implemented on any computing platform, several advantages are offered by a handheld PDA implementation [5]. In particular, the small size and light weight of a PDA make it an excellent platform for the highly mobile environments often associated with mini-UAVs.

In our implementation, the PDA initiates a wireless 802.11b connection to a full-size laptop (the “base station”) which is connected to a radio modem capable of reaching the airplane within a range of several kilometers. The PDA operator is free to wander wherever he or she wishes while controlling the UAV, as long as the PDA stays within the range of 802.11b communication. Alternatively, for applications where it is undesirable to create a physical location for the “base station,” the required equipment (a laptop, 900 MHz radio transceiver, 2.4 GHz video receiver, and respective power sources) can easily be stuffed in a small backpack, with only the PDA in front of the user to coordinate the activities of the low-level equipment. Video monitoring can occur on a eyeglass display. A clean and simple PDA interface, then, offers much more than the initial “gee-whiz” reaction; it hides the complexity of the underlying system.

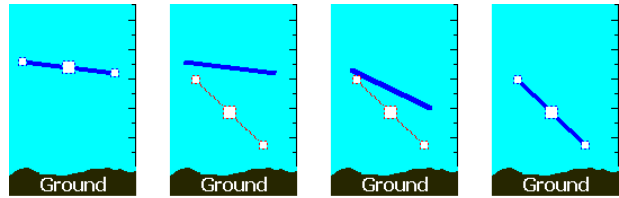


Fig. 3. Direct manipulation of the “wing-view” display. In the upper-left frame, the UAV is holding a relatively high altitude and a slight roll to the right. The upper-right frame shows the user-commanded target attitude. In the lower-left frame, the UAV is progressing to the target, and in the lower-right frame it has reached the new attitude, causing the control handles to return to the telemetry visualization.

Although we have not yet measured the relative levels of situational awareness (SA) provided by our UAV interface designs, the key features of the direct-manipulation PDA-UAV interface can be described in terms of the three-tiered definition of SA created by Endsley [6]: perception, comprehension, and projection. The following sections provide a brief analysis of the PDA interface using this framework.

#### A. Level 1 SA: Perception

The first task of any interface is to help the user *perceive* the current relationship between the UAV and the world, while not overwhelming the user with unnecessary information. We have found that presenting a simplified “wing view” of the UAV successfully abstracts two key characteristics of flight: roll and altitude. This display is shown in Figure 3.

The notion of creating an aeronautical instrument using a miniature airplane icon dates back to the early days of aircraft instrumentation [7]. The classic “primary flight display” found in virtually all cockpits, however, has a moving horizon. This makes sense when the pilot is inside the aircraft and experiencing the maneuver first-hand. In contrast, the UAV operator is standing on the ground, and a display with a moving horizon may confuse users not trained as pilots.

The “wing-view” display is created from the vantage point of an observer behind the UAV who is looking through an abstracted cross-section of the UAV’s main wing in the direction the UAV is flying. Despite its simplistic appearance, the “wing-view” display presents a persuasive abstract visualization of the instantaneous relationship between the UAV and the world.

Heading and velocity are displayed as gauges already familiar to most users: a compass and a speedometer. A precise scale for the velocity gauge is unnecessary and is intentionally not supplied. This allows the interface to achieve platform independence across a variety of UAV airframes, each of which will have a different range of acceptable velocities; commands are scaled on the base station into a safe range before they are sent to the UAV.

#### B. Level 2 SA: Comprehension

The next level of situational awareness is obtained by combining perceptual (Level 1) data to *comprehend* how



Fig. 4. Complete PDA Screen

these data relate to the overall goal. For a mini-UAV performing a surveillance task, the overall goal is to provide a useful video image of an object of interest. A critical simultaneous task, however, is that the user must keep the UAV in the air. With this in mind, we drew some small mountains and the word “Ground” on the bottom of the “wing-view” display. Furthermore, the “min-max” constraints of the airspeed gauge also help keep the UAV airborne, as it is altogether impossible for the PDA operator to stall the UAV.

Because there are only three instrument displays on the PDA screen, synthesizing their information is a simple matter and does not require a high mental workload. Figure 4 shows the overall layout of the interface, which is intended to present the user with a clean and intuitive panel of data visualizers and controls.

### C. Level 3 SA: Projection

The highest level of Situational Awareness described by Endsley is *projection*: the ability to predict what will happen to the system in the near future. By using direct manipulation and visual overlays, the future behavior of the UAV is displayed on the PDA alongside the current state. On all three displays, “actual” parameters are plotted in blue, and “desired” parameters are plotted in red. The future behavior of the UAV is thus immediately discernible to the user: if any red is showing on the display, the UAV is seeking to match the red visualization. If all displays only have blue visualizations, then the UAV is simply maintaining its current flight parameters.

It has been shown in many complex environments that direct manipulation can be very effective in allowing the user to subjectively generate desired system parameters [8]. Applied to mini-UAV control, this form of interaction can be much more natural than supplying numerical values. The wing-view display draws three “control handles” on the abstract representation of the UAV, signified by dotted white boxes. The handle in the center of the wing gives the user control of the UAV’s target altitude. Using the PDA stylus, the user simply drags the center of the UAV to the desired target altitude. The handle on each wingtip allows the user to drag a wingtip to create a desired roll angle.

The heading and velocity gauges have the same functionality and color scheme as the “wing-view” display: the

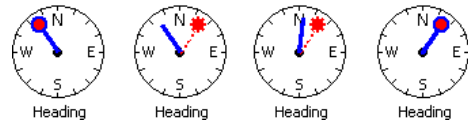


Fig. 5. Direct manipulation of the heading control.

user can drag the velocity or heading “indicator needle” to the desired position, and the airplane will then seek to match the user input, as shown in Figure 5.

### D. Laptop Implementation

We have ported our PDA interface to run on the standard Win32 platform. Although a full-size laptop offers significantly higher network and graphics performance, novice users actually found the laptop implementation to be more difficult to use when they were required to use a trackpad rather than a mouse. We suspect that this is due to the difficulty many users have in moving the cursor with a trackpad while holding down a trackpad button to drag a control, as is required by virtually all tasks in our direct-manipulation interface. We suspect that an interface that uses single- or double-clicks to select new parameter values, rather than continuous dragging, would be better suited to a laptop trackpad.

### V. VOICE CONTROLLER

We have implemented a voice controller that can recognize commands such as “turn left”, “climb”, “speed up”, “go north”, and the like, using a grammar of twenty commands. The speech recognition agent listens for these simple one to three word commands, and when a successful recognition is made, a speech synthesis agent offers instantaneous feedback by simply stating the command in the present progressive tense: “turning left,” “climbing,” “speeding up,” or “going north.” The process is fast and stable enough that the UAV can be reliably flown using these simple voice commands, either through direct line-of-sight visual contact or by using the UAV’s onboard video camera.

### VI. PHYSICAL CONTROLLERS

#### A. Attitude Joystick Controller

Most joysticks in traditional aeronautical control systems are rate joysticks – that is, a deflection from the resting position of the joystick produces a control surface deflection which under standard conditions will induce a proportional rate of change in the attitude of the aircraft. However, to promote semi-autonomous interaction and increase neglect tolerance, we chose to pursue a joystick mapping similar to what has been done in many “fly-by-wire” cockpits: the attitude joystick controller maps the deflection of the joystick to a deflection in the aircraft attitude from level flight. Our current parameters for this interface map the x-axis of the joystick to a fixed roll attitude ranging from -45 to +45 degrees, and y-axis deflections map to a fixed pitch attitude from -30 to +30 degrees. This style of interaction is particularly useful to users who are not trained as pilots,



Fig. 6. Single-handed operation of the Twiddler2 controller (left), and the physical icon controller (right).

as it is completely impossible to cause the mini-UAV to “barrel-roll” from over-deflecting the joystick in the x-axis, and inverting the aircraft by over-deflecting the y-axis is also impossible. In addition, the joystick throttle is mapped to the “min-max” airspeed range discussed previously to eliminate the potential for stalls.

### B. TrackPoint™ Controller

The Twiddler2™ controller [9], shown in Figure 6, is designed for single-hand operation, and features a built-in IBM TrackPoint device positioned under the thumb. This pointing device, found in many laptops, maps 2-dimensional forces to x- and y-axis mouse velocities. Our interface captures the mouse cursor and measures the mouse velocity in each axis, thus approximating the force placed on the TrackPoint by the user’s thumb. X-axis forces are mapped to the roll attitude of the UAV and y-axis forces are mapped to the pitch attitude of the UAV in the same fashion as the attitude joystick controller discussed in the previous section. Time-averaging the force estimates helps compensate for mouse acceleration and other unwanted operating system effects, and multiplying the deflections of each axis at every timestep by a decay scalar slightly less than one provides a smooth return to the origin when the user stops applying pressure.

Because event-driven operating systems simply stop sending messages when movement of the mouse cursor stops, the decay scalar allows the interface to handle the intermittent flow of data, treating the lack of data as an indication that forces are no longer being applied to the TrackPoint device. After experimentation, we settled on a time-averaging constant that allows the user to achieve full deflection on the interface with approximately one-half second of strong pressure on the TrackPoint. The decay scalar was set so that the interface returned to zero deflection approximately one second after the user completely removed pressure from the device. These constants help smooth the command stream being sent to the UAV while still capturing the intent of the user.

### C. Physical Icon Interface and Mixed Reality

We created a novel Human-UAV interaction scheme by using a small model airplane as a “physical icon” of the real UAV. Inserting an autopilot, radio modem, and power source into the physical icon allows it to continuously track



Fig. 7. Mixed-Reality Physical Icon interface. The actual telemetry and desired attitude are plotted as aircraft in transparent blue and red colors, respectively. The user has requested a climb and a right roll, and the UAV has nearly matched these commands. The video image has been rolled so as to level the horizon.

its own attitude in three-dimensional space. These measurements are sent to a nearby computer, which converts the attitude of the physical icon into roll and pitch commands for uploading to the real UAV. The UAV is surprisingly easy to control using this interaction scheme, as the user is able to maintain high situational awareness because he or she is literally holding a physical representation of the UAV. In steady-state conditions, the physical icon serves as a model of the UAV state, assisting the user in the comprehension (Level 2) phase of situational awareness discussed previously. Immediately after the user changes the attitude of the physical icon, it becomes part of the user’s projection of the near-future behavior of the UAV, corresponding to the third level of situational awareness.

To assist the user in differentiating between these two roles of the physical icon, we have created a graphical interface similar to the PDA interface discussed previously. Telemetry from the UAV is used to orient a blue-colored OpenGL 3D airplane model. The attitude of the physical icon is used to orient an identical red-colored model. After a command has been received and matched by the UAV, the telemetry of the UAV matches the orientation of the physical icon and the two models merge to one. Immediately after the user alters the attitude of the physical icon, the red model serves as the target while the blue model tracks the progress of the real UAV in matching the desired orientation.

To further increase the functionality of this interface, we introduced an element of mixed reality [10] by superimposing the two OpenGL airplane models onto a horizon-stabilized video feed produced by digitizing and rotating the UAV’s analog video feed by the inverse roll angle of the airframe (Figure 7). Because the camera is rigidly mounted to the airframe, this simple transformation helps stabilize the video feed and allows the virtual-reality models to appear as if they belong in the camera frame itself. The user has the impression that the UAV’s onboard camera is instead a “chase camera.” Because the user need not attempt to perform the horizon-stabilizing rotation though mental effort, the user is able to focus more energy on analyzing the content and information in the video stream.

## VII. INTERFACE EVALUATIONS AND OBSERVATIONS

Physical UAVs have been flown extensively with each interface to gather anecdotal evidence, and simulations were used to carefully quantify and reinforce the anecdotal evidence. Our UAV fleet includes a variety of flying-wing and V-tail UAVs ranging in wingspan from 2 to 5 feet, and in weight from 1 to 5 pounds. By placing all airframe-specific calculations on the base station and the on-board autopilot, the only difference the high-level user notices when switching between airframes is that larger and heavier UAVs take more time to change altitude and velocity.

To provide a quantitative comparison of the relative difficulty of issuing commands with our interfaces, we created a testing scheme in which users flew virtual UAVs in a simulator which was capable of generating a real-time perspective identical to that produced by a mini-UAV's onboard camera[11]. The simulator was programmed to superimpose simple commands in large letters on the simulated camera feed, such as "Turn Left!" or "Climb!" Log files were generated that recorded the time required for the user to make each requested change in the UAV state.

Users were thus required to focus their attention on the simulated camera image, extract a desired action from the camera view, operate the interface, and return focus to the simulated camera view for confirmation of the success of the command. This interaction cycle is conceptually similar to the cycle performed by users performing a tracking or searching task with a real UAV. Median user response times function as estimates of the relative difficulty of issuing a command via a particular interface. Four novice users with no piloting experience participated in this study, and the response times were very similar for all users. The following table summarizes these results.

TABLE I  
MEDIAN TIME REQUIRED FOR ISSUANCE OF COMMANDS

Interface	Median Time (s)	Std. Dev.
Attitude Joystick	1.04	0.54
Physical Icon	1.24	0.41
Attitude TrackPoint™	1.31	0.46
Laptop Direct-Manip. (Mouse)	1.43	0.53
PDA-Based Direct-Manipulation	1.80	0.68
Laptop Direct-Manip. (Trackpad)	2.19	0.98
Voice Recognition	2.37	1.11
Numerical Parameter Entry	2.73	1.17

The interfaces can be grouped into four categories which both describe their functionality and account for their median response times.

### A. Physical Interfaces

The first three interfaces in the table have the fastest response times, and they all map a physical motion of the user to a desired attitude of the UAV. This mapping

is simple, straightforward, and easily understood by the user. In addition, these interfaces do not require a focal shift between the onboard camera feed of the UAV and the interface itself.

However, interaction time is certainly not the only factor by which to judge the efficacy of an interface. The very fact that these interfaces require physical components could be detrimental or even prohibitive, depending on the application. A UAV control station built into a large vehicle would have ample space and power for any of the physical interfaces, and an attitude joystick or physical icon interface would offer fast and easy control of the UAV. However, in a wilderness search-and-rescue application, the UAV interface might need to be transported on a small vehicle such as an ATV. In this context, the physical requirements of this class of interface may be a limiting factor.

### B. Direct Manipulation

A direct-manipulation interface offers a compromise between the response times of the physical interfaces and the numeric parameter-based interface. Perhaps its greatest strength is its neglect tolerance – the ability of the interaction scheme to function as user attention decreases[12]. A physical interface such as a physical icon or attitude joystick requires the user to literally hold the desired attitude command. The direct manipulation scheme, by contrast, allows the user to symbolically express the desired attitude. No further interaction with the interface is necessary until a change in attitude is desired. As a result, this type of interaction scheme is much better suited to applications where the user has other pressing concerns, such as navigating an off-road vehicle. In addition, this scheme is better suited to multi-agent teleoperation, as several direct-manipulation panels could simultaneously display state and accept commands for several UAVs.

### C. Voice Recognition

In situations where the UAV is flown solely using visual contact, the direct-manipulation interfaces do not seem to adequately address the problem of tracking the airplane, as the focal shift from standard reading distance to the several hundred yard (or greater) distance to the UAV is very difficult to perform reliably. Physical interfaces are also problematic, as the control mappings must be inverted when the UAV is flying directly at the user. We found the voice controller to be very effective in this application, as users can completely devote their visual attention to tracking the airplane and allow the headset microphone to pick up and transmit commands. Commands such as "go south" are environment-relative, as opposed to the UAV-relative commands produced by physical controllers, and thus need not be inverted when the UAV flies toward the operator.

Real-world tests with this interface have demonstrated that ambient wind noise and conversation can wreak havoc on the reliability of the voice-recognition system. A method of "muting" the microphone input is required, but even with

such a system in place, considerable difficulties arise in environments with strong winds or loud background noises. However, our experience has shown the voice interface to be very valuable, especially under favorable weather conditions.

#### D. Numeric Parameters

Of course, users can fly UAVs via traditional parameter-based interfaces, but they have several important drawbacks that are addressed by the high-level interfaces previously discussed. First and most obviously, parameter-based interfaces require users to type. In cold- or adverse-weather conditions, typing may be very difficult, if not impossible. Typing also requires the keyboard or keypad to be supported by either a stationary surface or a user's hand, and requires at least one hand for parameter entry. In highly mobile environments, such stability may be infeasible.

Second, users of parameter-based interfaces must perform additional steps of mental arithmetic, such as incrementing or decrementing parameter values. Informal tests with many users have demonstrated that many people have difficulty performing high-stress, time-critical arithmetic problems, no matter how trivial they may seem. This observation is confirmed by the fact that our test subjects unanimously found the numeric interface to be the slowest to operate.

In short, traditional numeric interfaces to semi-autonomous UAV control modes leave much to be desired. Although they are excellent for high-accuracy testing and evaluation of UAV hardware and software, they are not ideal for end users with little to no flying experience.

#### E. Summary

The observations of the preceding sections can be summarized by three factors that help determine the effectiveness and applicability of a human-UAV interface: *precision*, *mobility*, and *responsiveness* to user input. In the following table, we use the same categorization of interfaces used in the preceding sections: the joystick, physical icon, and TrackPoint interfaces are abbreviated as simply "Physical," the PDA- and laptop-based direct-manipulation interfaces are abbreviated as "DM," and the voice-recognition and numeric parameter interfaces are abbreviated as "Voice" and "Numeric," respectively.

	Precision	Mobility	Responsiveness
High	Numeric	DM	Physical
	Physical	Voice	DM
Low	DM	Numeric	Voice
	Voice	Joystick	Numeric

### VIII. FUTURE WORK

Our development efforts and field studies suggest several areas of future work. We intend to extend our work in video stabilization and processing by incorporating a live video image from the camera into the direct-manipulation interfaces. This will require some sort of a mixed-mode

display and perhaps the ability to switch between a full-screen camera view and a separate window that will contain a small version of the camera feed and all the controls necessary to fly the UAV. Waypoint-driven interfaces could also be incorporated into this mixed-mode display. We would also like to experiment with superimposing a transparent image of a compass on the image of a downward-facing UAV camera, and allowing heading commands to be generated simply by tapping or circling an object of interest with a stylus. We are also investigating various methods of collision avoidance using computer vision as well as various other methods of sensor processing.

### IX. CONCLUSIONS

Although parameter-based semi-autonomous UAV teleoperation interfaces offer great precision, they do so at the expense of high user workload and a steep learning curve. Searching of some form is required for many applications of mini-UAVs, and predetermined waypoint-based flight plans are difficult to prepare for such missions. Moreover, the flight plan may have to be heavily modified or forgotten altogether as the mission progresses. In situations such as these, neither parameter-based nor waypoint-based interfaces seem to offer the ideal control methodology. We have presented and evaluated several interaction schemes which serve to isolate the user from the technicalities and arithmetic requirements of the underlying parameter-based control system. We conclude that a high-level semi-autonomous interaction scheme drastically simplifies the typical commands required for searching tasks using a mini-UAV, especially when the interface design is tailored to the physical and cognitive requirements of the application.

### REFERENCES

- [1] T. Coffey and J. Montgomery, "The emergence of mini UAVs for military applications," in *Defense Horizons*, no. 22, December 2002.
- [2] O. of the Secretary of Defense, *UAV Roadmap*, 2002, pp. 127–133.
- [3] D. Kingston, R. Beard, T. McClain, M. Larsen, and W. Ren, "Autonomous vehicle technologies for small fixed-wing UAVs," in *AIAA 2nd Unmanned Unlimited Systems, Technologies, and Operations – Aerospace, Land, and Sea Conference and Workshop & Exhibit*, San Diego, CA, September 2003, paper no. AIAA-2003-6559.
- [4] T. W. Fong, C. Thorpe, and C. Baur, "Advanced interfaces for vehicle teleoperation: Collaborative control, sensor fusion displays, and web-based tools," in *IEEE International Conference on Robotics and Automation*. IEEE, April 2000.
- [5] T. W. Fong, C. Thorpe, and B. Glass, "Pdadriverr: A handheld system for remote driving," in *IEEE International Conference on Advanced Robotics 2003*. IEEE, July 2003.
- [6] M. R. Endsley, "Toward a theory of situation awareness in dynamic systems," in *Human Factors*, vol. 37, March 1995.
- [7] O. E. Patton, *Aircraft Instruments: Their Theory, Function and Use*. D. Van Nostrand Company, 1941, pp. 110–114.
- [8] B. Shneiderman, *Designing the User Interface, Third Edition*. Addison-Wesley, 1998, pp. 186–233.
- [9] [Http://www.handykey.com](http://www.handykey.com).
- [10] R. T. Azuma, "A survey of augmented reality," in *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 4, August 1997.
- [11] The open-source Slope Soaring Simulator, written by Danny Chapman, is available at <http://www.rowlhouse.co.uk/sss/>.
- [12] J. W. Crandall, C. Nielsen, and M. Goodrich, "Towards predicting robot team performance," in *2003 IEEE Conference on Systems, Man, and Cybernetics (SMC '03)*.