

## Brent A. Terwilliger, Ph.D.

### ABOUT ME:

I have a passion for aviation and unmanned system development, fostered by my education at the Embry-Riddle Aeronautical University Daytona Beach (BS, Aerospace Studies, 2000) and Worldwide/Extended campuses (MS Aeronautical Science, 2005). I have a long-standing interest in the success of the University and its learners, which I continue to support through my interactions with faculty, staff, and students as an alumni, researcher, and advocate. I have spent the last decade working in defense contracting, simulation, and training system development, while continuing to expand my knowledge base through formal education (i.e., Ph.D. and MS degrees). In my spare time, I pursue research, actively read, travel, and serve as the Vice President of the Embry-Riddle Chapter of the Delta Chi Fraternity Alumni Board of Trustees (initiated as an undergraduate Fall 1996; see Figure 1).



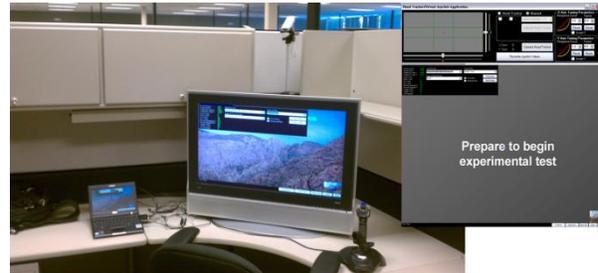
**Figure 1. U.S. Representative John L. Mica (left), Dr. Brent Terwilliger (center), and Frederick W. Leonhardt (right) at the Embry-Riddle Delta Chi Fraternity 40th Anniversary in Daytona Beach, FL**

I have recently begun writing a blog to actively stay engaged in advancements pertaining to unmanned systems, teleoperation, situational awareness, human-machine interfaces (HMI), simulation, and human-in-the-loop research. In support of my research and other academic pursuits, I have developed the following series of systems/applications using commercially-off-the-shelf (COTS) and custom developed components.

### Simulated Unmanned Operator Test Station

The simulated operator station was a custom developed system for experimental testing to capture the SA of each participant. The appearance and function of this system approximated the look, feel, and interaction of a generic GCS. The system hardware was composed of several components that facilitated test participant use of the experimental visual interaction methods (i.e., static eyepoint, analog joystick, head tracker,

uninterrupted hat/POV switch, and incremental hat/POV switch). The cost of the system hardware was under \$1000 USD.

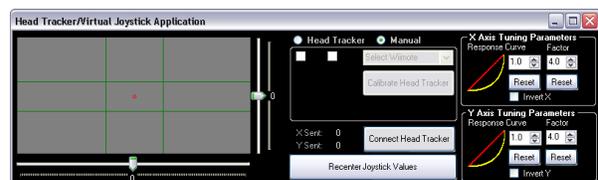


**Figure 2. Simulated Unmanned Operator Test Station**

The simulated operator station software was composed of several components that facilitated test participant use of the experimental visual interaction methods (i.e., static eyepoint, analog joystick, head tracker, uninterrupted hat/POV switch, and incremental hat/POV switch). The primary software components were the head tracker/virtual joystick application and the visual SA test application, described below.

### Head Tracker/Virtual Joystick Application

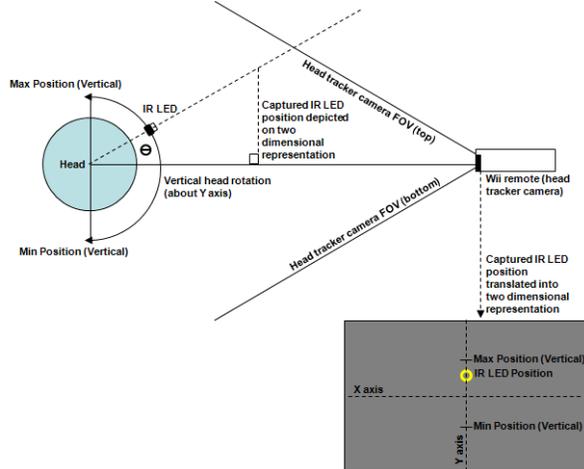
This application was designed to capture and process inputs from a custom created single-IR LED head tracker. The software captured the head tracker input values and converted them to values that correlated to the analog X and Y axes of a joystick. The converted values were then packaged and output as a virtual joystick, recognizable for use by external programs. To simulate a virtual joystick, this component interfaced to the PPJoy application, which had developmental virtual COM joystick interfacing capability. The head tracker/virtual joystick application was an interface between the head tracker and the visual SA test application. This component also featured several capabilities not applicable to the experimental testing: ability to display a heads up display (HUD) and an interface for manual control.



**Figure 3. Head Tracker/Virtual Joystick application**

The head tracker/virtual joystick application was designed to run on a Windows XP system, using the DirectX 9.0c, Wiimote, and PPJoy support libraries to accept input from a Nintendo Wii remote device over a bluetooth connection established through the Toshiba bluetooth stack and USB bluetooth adapter. Once the Wii remote was configured for operation and interfaced to the Windows PC, it was able to track the position of an IR LED within the field of view (FOV) of the Wii remote device camera (i.e., head tracker camera). The captured IR LED position was mapped to a two dimensional area (X and Y axes) using the Wiimote library components and translated into a pitch (rotation about head's Y axis; see Figure 4) and yaw (rotation about head's Z axis) value. The application has several unique features:

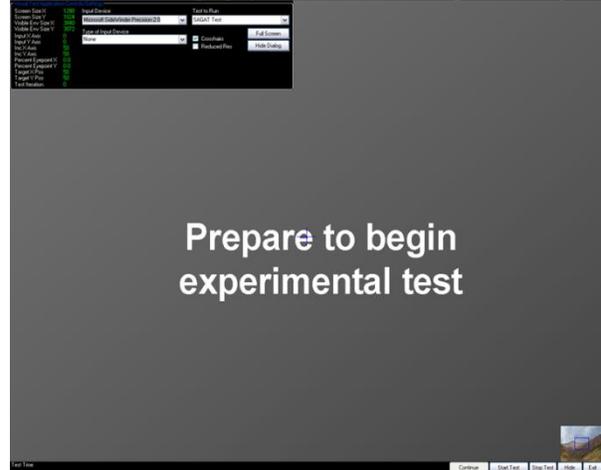
- two-dimensional rotational angle calculations (used Pythagorean geometry)
- tuning parameters (factor, response curve, and initial calibration)



**Figure 4. Head tracker angular position, as viewed from side depicting Y-axis rotation.**

### Visual Situational Awareness (SA) Test Application

This application was designed to capture and process user inputs for the control of the visual eyepoint and displayed the egocentric visual environment, dynamic depiction of geometric objects, and movement of the visible screen area within the visible environment area. The inputs included the head tracker and joystick. There were two primary components to the visual SA test application, the controls/settings window and the interaction controls (see Figure 5). Not all of the controls contained in the GUI were necessary for the experimental testing. Additional controls were necessary for debugging and component evaluation in the application development process.



**Figure 5. Visual SA Test application**

The visual SA test application was designed to run on a Windows XP system with a video resolution of 800 x 600, using the DirectX 9.0c support library to accept input from joystick devices or through user interaction with the application GUI. The software application used the DirectX 9.0c library to connect to and read the state of the joystick, including axes position and button depression. The software was designed to run the SA experimental test and the associated alignment and familiarization activities.

### Tactile Aural Simulated Kinetics (TASK) Unmanned Ground Vehicle (UGV) System

The predecessor of my Ph.D. research to explore the impact of dynamic eyepoint manipulation originated from development and refinement of an earlier concept to replicate the operating environment of a custom built remotely operated vehicle (ROV). This conceptual research was termed Tactile Aural Simulated Kinetics (TASK). Eventually the lessons learned from the development, implementation, and use of the TASK research formed the basis of the recent research, including:

- Mitigating the effects of spatial disorientation
- Identification of dynamic eyepoint interaction techniques
- Effect of visual interaction on operator SA

The purpose of TASK was to mitigate the factors that reduce situational awareness using simulated motion/force feedback and aural/sound cues, increasing an operator's performance and abilities. The incorporation of tactile and aural data was believed to provide an operator with additional means to interpret data through methods they are already familiar with and can react to faster. The TASK concept provided a means to effectively capture and provide the ROV

operator with “live” simulated tactile feedback and aural data.

**TIGER Vehicle** - The Tactile Information Gathering Environmental Remote (TIGER) vehicle (see Figure 6) was controlled by the ground control system (GCS) and relayed captured data (visual, aural, and motion) back.



Figure 6. TIGER vehicle

**TASK GCS** – The user interface that simulated the “Look, feel and sound” of the remote vehicle-operating environment. Provided control instructions and data presentation for the TIGER including visual orientation, aural playback (simulated sound) and vibration simulation (see Figure 7).



Figure 7. TASK GCS

**Teleoperation Development Kit (TDK)**

The TDK is a customizable/reconfigurable remote servo control system (hardware and software) that replaces the need to design and build the software/hardware of an unmanned vehicle control layer. TDK provides a control solution to unmanned vehicle developers or remote control (R/C) hobbyists

with a control radius requirement of 5 miles or less (see Figure 8).

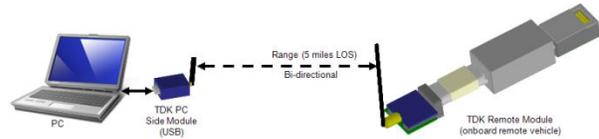


Figure 8. TDK overview

Example platforms include:

- Micro Unmanned Aerial Vehicle (MUAV)
- Small Unmanned Aerial Vehicle (SUAV)
- Unmanned Ground Vehicle (UGV)
- R/C aircraft
- R/C cars, trucks, and tanks
- R/C boats

The TDK replaces the need to design and build the software and hardware control layer (see Figure 9), providing a user with additional time and resources to focus on the development of the actual vehicle platform.



Figure 9. TDK application

Use of the TDK system would:

- allow individuals unfamiliar with software or hardware development to create an unmanned research, development, or test platform
- facilitate rapid development of complex vehicle controls that provide higher fidelity/increased capabilities
- provide flexibility through reconfigurable design (reuse for other platforms, activities, or requirements; see Figure 10)
- increase the accuracy of first person view (FPV) remote control systems
- increase the fidelity of simulate R/C aircraft controls (i.e., use of actual flight sticks/yolks, throttle controls, pedals, and custom developed user interfaces).

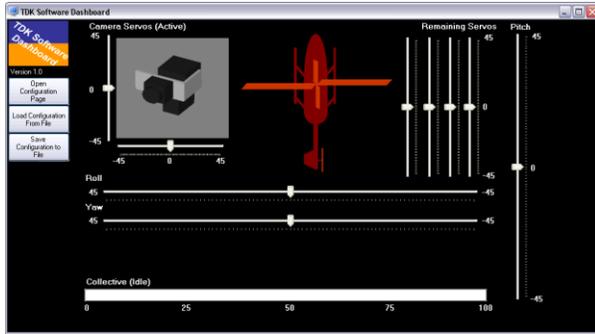


Figure 10. TDK helicopter configuration

### Generic Rotorcraft Flight Dynamics Model (FDM)

This application was developed to create a generic rotorcraft FDM for comparative analysis purposes. It uses a USB joystick or on screen (i.e., embedded) flight controls (cyclic, collective, pedals, and three stage throttle/engine control unit (ECU) as user inputs to the model (see Figure 11).



Figure 11. Generic Rotorcraft FDM

Available airframes include the AS350 Squirrel, SA330 Puma, AH-64/D, EH101 Merlin, and CH-47 Chinook. However, new configurations/ airframes can be added by editing a \*.xml data file and adding values for the number of rotors, main rotor diameter, tail rotor diameter, total engine horsepower, maximum operating altitude, and the airframe weight. In addition, sling loads can be added to the model and their characteristics (i.e., weight and cable length) modified during operation. The flight dynamics calculations of the FDM take ground effect region, weights, rotor sizing, engine output, and sling loading into consideration.

The following are the outputs of the FDM, available through a shared memory interface:

- Vertical Speed
- Altitude
- Speed
- Rotor Torque

- Airframe Acceleration
- Airframe Velocities
- Airframe Rotations
- Latitude/longitude position

### FlightGear UDP Generic Interface

This application (see Figure 12) was developed to interface with the FlightGear simulation software using user datagram protocol (UDP). It featured a shared memory interface that could link to the Generic Rotorcraft FDM application to pass through information to use the FlightGear application as an image generator (IG).

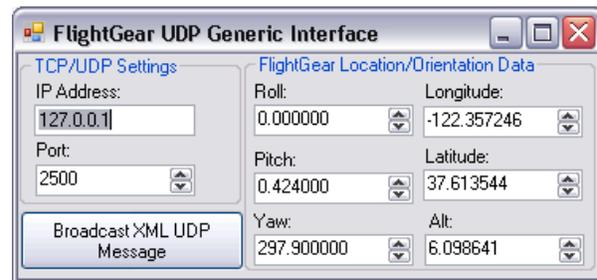


Figure 12. FlightGear UDP Generic Interface application

### Directory Reader/Writer

This simple application was developed to provide a search tool to locate specific files or file types in a folder/directory and write the information into a user specified log file (\*.txt format).

### On Screen Dialog (OSD)

This application (see Figure 13) was developed to display an OSD/heads-up-display (HUD) reticle, editable by the user. The edit features include (transparency level, color, and ON/OFF state). This application was originally developed to provide a reticle overlay for camera data captured from the operation of an unmanned vehicle (i.e., FPV camera).



Figure 13. OSD application

For more information, please feel free to visit my blog: <http://brentterwilliger.blogspot.com>

Brent A. Terwilliger, Ph.D.  
[brentterwilliger@yahoo.com](mailto:brentterwilliger@yahoo.com)  
 (607) 624-4275