Multiple Remotely Piloted Aircraft Control: Visualization and Control of Future Path

Gloria Calhoun[1], Heath Ruff[2], Chad Breeden[1], Joshua Hamell[3], Mark Draper[1], and Christopher Miller[3]

[1] Air Force Research Laboratory (AFRL), Wright-Patterson Air Force Base, Ohio
[2] Ball Aerospace and Technologies Corporation, Fairborn, Ohio
[3] Smart Information Flow Technologies, Minneapolis, MN
{gloria.calhoun,heath.ruff,chad.breeden, mark.draper}@wpafb.af.mil, {jhamell,cmiller}@sift.net

Abstract. Advances in automation technology are leading to development of operational concepts in which a single pilot is responsible for multiple remotely piloted aircraft (RPAs). This requires design and evaluation of pilot-RPA interfaces that support these new supervisory control requirements. This paper focuses on a method by which an RPA's near-term future flight path can be visualized and commanded using the stick and throttle. The design decisions driving its symbology and implementation are described as well as preliminary quantitative data and pilot feedback to date.

Keywords: remotely piloted aircraft, unmanned air systems, flight path, display symbology, RPA, UAS, flexible automation.

1 Introduction

Advances in automation technology are leading to development of operational concepts in which a single pilot is responsible for multiple remotely piloted aircraft (RPAs). With RPA flight highly automated in this vision, multi-RPA systems will necessarily involve supervisory control with requirements for the pilot to frequently shift attention between RPAs. Displays that facilitate rapid retrieval of each RPA's state and associated tasking are required. Moreover, new control methods will be necessary to enable the pilot, when time and attention are available, to quickly and precisely redirect any RPA's path and tailor the supporting automation.

To tackle this design challenge, the Air Force Research Laboratory "Flexible Levels of Execution – Interface Technologies" (FLEX-IT) effort developed a demonstration simulation illustrating four different RPA control modes: manual (pilot controls the RPA's flight with stick and throttle), noodle (enabling the pilot to visualize and command the RPA's future path), micro-plays (quick maneuvers initiated by verbal command), and plays (complex tasks initiated by the pilot, for

R. Shumaker (Ed.): VAMR/HCII 2013, Part II, LNCS 8022, pp. 231–240, 2013.
© Springer-Verlag Berlin Heidelberg 2013

example, command to monitor a specified target). The simulation (Fig. 1) also illustrates the symbology and controls to support seamless transition between any and all of the four control methods. Early and refined versions of the simulation are described elsewhere [1, 2, 3].

The FLEX-IT multi-RPA simulation was used to support operator-centered efforts to acquire feedback from operators, many of them USAF pilots with experience flying RPAs. The methodology and RPA operator feedback have been reported [3, 4]. Pilots indicated that a FLEX-IT approach is indeed promising in terms of providing intuitive multi-level control methods to interact with automation. A key to the utility of FLEX-IT is the symbology that provides feedback on the automation's processing, proposed plans, and state of execution.

Fig. 1. Flexible Levels of Execution – Interface Technologies (FLEX-IT) Simulation

Of the four control modes integrated, one was a mode enabling rapid command of a vehicle's immediate future path using the same flight controls (stick and throttle) as for normal flight. It also provided visualization that made use of the vehicle's flight model to show the result of these actions on the future path of the vehicle in the form of a flexible 'noodle' that extended from the front of the aircraft symbol and reacted, bending and shifting, as if the pilot were currently flying the commanded path. The noodle was particularly novel and viewed favorably by the pilots. In one evaluation, all six pilot participants rated this feature as a likely aid for future control of multiple RPAs. Specifically, they selected one of the top two favorable ratings on a 5-point scale [3]. Additionally, the pilots' comments were aligned with the ratings. Some described the noodle concept as "very intuitive" and providing a good quick visual representation of the vehicle's future flight path [3]. They indicated that the ability to input the future path of the vehicle with automated tools via the stick and throttle manipulations would be very useful, at least for the existing pilot community. For example, the tool provides a quick means of rerouting an RPA to avoid an area that suddenly became restricted. With the noodle, a complex maneuver can be quickly defined and commanded to the RPA (as constrained by the vehicle's real flight capabilities) ahead of time, allowing the pilot to turn attention to another vehicle/task while the first RPA flies its noodle directed path.

The remainder of this paper will focus on this novel control mode. First the approach will be described in more detail. This will include examples of alternative designs entertained in the process of developing this automated tool. Next, the results of a study designed to collect data on how long it takes to employ this control method to designate a vehicle's future path will be presented. This quantitative data supplements the qualitative data reported earlier [3, 4].

2 RPA Candidate "Noodle" Control Mode

This control mode is termed "noodle" as its symbology includes a flexible line segment resembling a variable length bendable noodle emerging from the nose of an RPA symbol on the map display. Two noodle symbology sets were implemented and demonstrated. With the *Predicted Noodle,* a line (colored to match the respective RPA's symbol) is drawn to show the forecasted flight path the RPA will fly, given the current state of control inputs is maintained. By employing faster-than-real-time software simulations, the RPA's position is predicted as a series of points in the future based on current autopilot or manual joystick control. These points are then connected into a line that shows the expected flight path for the specified duration. Several alphanumeric readouts are presented along the noodle. The Predicted Noodle example shown in Figure 2 illustrates that at the end of the purple line, the RPA will have traveled for 119 seconds and be at an airspeed of 86 KIAS, an altitude of 14,000 feet MSL, and be oriented at a heading of 275 degrees. This prediction is perfect in the simulation, since the same model is used to predict aircraft path as is used to actually 'fly' it. In a real world deployment, winds and other factors will introduce error and will need to be compensated for by estimates and approximations. In the majority of control modes, the presentation of the Predicted Noodle is under pilot control in the simulation: either displayed or turned off.

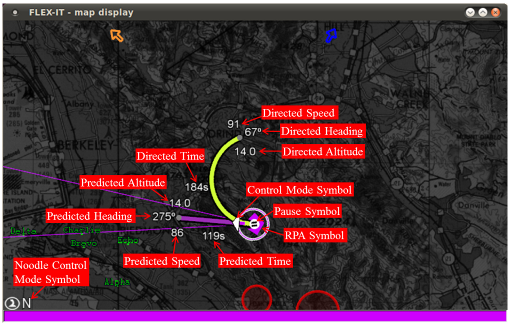

Fig. 2. Illustration of the Noodle Symbology in the FLEX-IT Simulation. (Purple line is Predicted Noodle, yellow curve is Directed Noodle. Note that red flags are annotations for this graphic and are not presented during use.)

In contrast to the Predicted Noodle that solely displays future path given current state, the *Directed Noodle* mode serves as both a display and control, enabling the pilot to specify the future path of a vehicle. For this mode, the functionality of the stick and throttle are remapped. Inputs on the right console control stick specify the RPA's destination change in heading and altitude at the end of the noodle. One throttle button controls noodle duration, while throttle movement controls desired airspeed. Exact noodle length is derived from noodle duration and airspeed setting. Once the pilot approves the setting (via a switch on the throttle), the RPA follows the noodle path.

When invoked, the Directed Noodle is visualized with a yellow "directed" path shown in addition to the Predicted Noodle (see Fig. 2). The yellow Directed Noodle is manipulated by the stick and throttle as a possible future path of the aircraft, but it is not carried-out unless the operator pushes a switch on the throttle to confirm the path. At this point, the Predicted Noodle reads that the control commands for the RPA have changed, and re-calculates accordingly, effectively taking the place of the previous Directed Noodle segment. This provides a very natural method, using the familiar flight controls, to input a future flight path that (because it is computed and shown using the same flight control algorithms used to control the aircraft itself) obeys all known constraints and capabilities of the RPA in its current state.

Figure 2 shows the effect of switching to noodle control after an automated play was enacted to monitor a target. As soon as the operator executes the noodle control mode, a pause symbol (two parallel lines) appears on the RPA icon to let the operator know that the play has been interrupted. The letter "N" next to the joystick symbol (lower left) shows that the joystick is in noodle control mode and the arrow icon on the RPA symbol shows that the aircraft is in a semi-autonomous state of control (in between auto-pilot/waypoint following mode and manual control). The purple-colored Predicted Noodle indicates the vehicle's current flight path. In Figure 2, the Directed Noodle is being maneuvered to reach an altitude of 14,000 feet, a targeted maneuver speed of 91 KIAS, and a heading of 67 degrees after 184 seconds of flight time.

A new Directed Noodle can be added at the end of the accepted noodle, and manipulated for additional flight path specification. Thus, multiple Directed Noodles legs can be "chained" together to command complex near-term flight maneuvers. Future segments can also be overridden by pulling back on a throttle switch. Each backward switch employment moves the currently manipulated Directed Noodle back one segment in the path. This allows for the pilot to redefine undesired noodle segments without discarding the entire sequence.

The intuitive use of manual inputs by the pilot, along with supporting automation, provides a means of quickly specifying a vehicle's future path with more precise control/granularity without employing detailed menus/procedures associated with complex route planning software. More importantly, this new method of flight path control may enable a pilot to devote more dedicated attention to other supervised RPAs.

3 Design of the RPA Noodle Control Mode

The noodle control mode described above represents the results of a series of design and evaluation cycles, each considering an alternative approach for presenting required information and supporting control inputs. Each cycle involved obtaining the expert opinions of RPA and manned aircraft pilots, while the methodology employed varied (ranging from PowerPoint illustrations to actual instantiations in simulation). Some design questions included: What should the color, style, and width of the line be to ensure visibility and yet minimize clutter on the map? What other information is needed to describe the future flight path? How should this information be presented (e.g., labels on the noodle) to minimize clutter and be visible in relation to the map? How should the pilot initiate the noodle control -- what input should control the functionality of the stick and throttle? How best can the operator be informed of the current mode of these controllers and the state of the path under noodle construction?

One question that received considerable attention was how best to indicate altitude variations that reflect the pilot's inputs during the construction of the Directed Noodle. One limiting factor in providing this feedback is that the map display is two-dimensional. Several methods of presenting altitude information directly on the map were considered (see Fig. 3). The first option, adding plus and minus readouts at the end of the noodle was determined to be inadequate in terms of saliency (Fig. 3a). However, symbology was needed to depict climb/descent for each noodle segment/chain, so "+" or "-" symbols were added in-line to reflect the individual segment altitude change (Fig. 3b). Symbology currently used by USAF pilots to denote climbing and descending turns was also considered (Fig. 3c) – both along the line (Fig. 3d), or in-line with the noodle (Fig. 3e). However, it would be difficult to depict this symbology for tight turns, so one design idea was to widen the noodle so as to completely surround the climb/descent symbology (Fig. 3f). Several coding methods involving the noodle line itself were considered, such as depicting the line segment in different shades (Fig. 3g) or colors (Fig. 3h). These methods were fairly visible on the map without adding clutter and helped portray whether the altitude was increasing or decreasing across the corresponding segment. However, color is a key coding method to differentiate symbology associated with one RPA versus another. All the aforementioned approaches were similar in that they only indicate whether the altitude increased or decreased in respect to some threshold. This binary information fails to provide an indication of the magnitude of the change.

A method commonly used to depict altitude changes on maps was next considered. Contour maps use circles to depict specific altitudes (e.g., every 100 ft of altitude change), with the proximity of each circle to other circles an indication of how rapidly altitude is changing. This convention can be applied to symbology overlaid on the noodle line, with the proximity of the symbols indicative of the severity or degree of altitude change. However, increases in altitude still need to be distinguished from decreases. Thus, it was decided to use two symbol elements to denote this difference. A caret symbol (^; similar to the notation suggested by pilots; Fig. 3c) is used in the current simulation to denote a vehicle's climb in altitude (Fig. 3i). A straight line perpendicularly intersecting the noodle represents a descent (Fig. 3i). These two symbols are also easier to code and draw on the noodle, although the width of the noodle had to be increased slightly to increase the visibility of the symbology. Additionally, explicit altitude readings are displayed at each segment barrier.

Figure 3i illustrates how both the direction and magnitude of altitude change are concurrently presented. The effect of chaining multiple Directed Noodles into an extended flight path is also shown. The RPA is approaching the start of two chained established (i.e., Predicted) noodle segments (blue-colored route) in which the predicted altitude will start at 15,000 feet. The RPA will travel over two noodle segments in 163 seconds of flight time. The first segment has the RPA descending, which is denoted by the hash-marks (| symbols) added in-line to the noodle symbology for every 50 feet of descent, to reach an altitude of 14,800 feet at the start of the next segment. This next noodle segment has the RPA climbing, which is denoted by the symbol (^) added in-line to the noodle at every 50 feet of climb, directing the aircraft to climb to ‘15.2’ thousand feet by the end of the noodle segment. At this point, the chained noodle segments have ended and the Directed Noodle is made available for further manipulation in order to continue chaining noodle segments into a flight path.

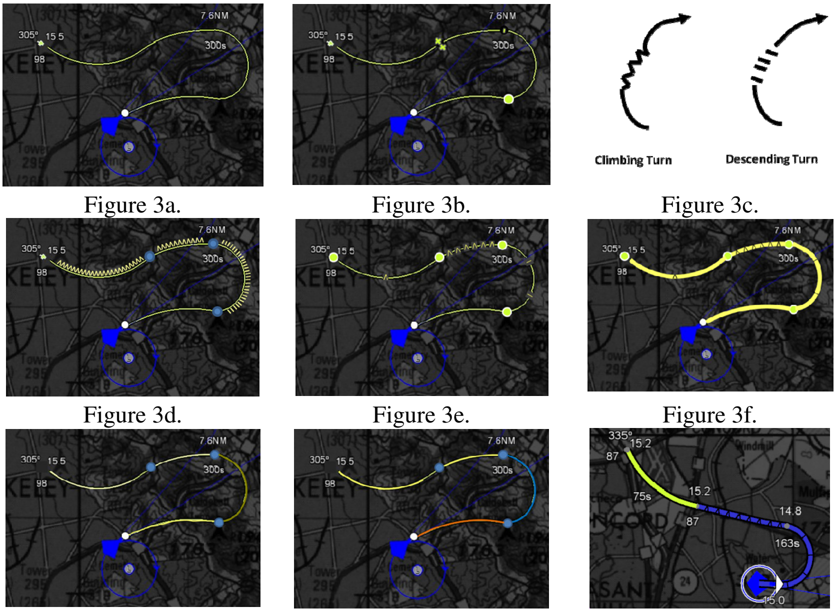

Figure 3a. Figure 3b. Figure 3c.

Figure 3d. Figure 3e. Figure 3f.

Figure 3g. Figure 3h. Figure 3i.

Fig. 3. Illustration of Different Methods Considered for Conveying Segment Altitude: 3a. “+” symbol. 3b. In-line “+” and “-” symbols. 3c. Pilot drawing symbology. 3d. Noodle augmented with pilot symbology. 3e. Pilot symbology in-line with noodle. 3f. Pilot symbology in-line with wider noodle enclosing pilot symbology. 3g. Shading. 3h. Color. 3i. Current implementation.

4 Evaluation of Noodle Construction

In evaluations conducted to date, feedback has been very favorable about the utility of the noodle tool, both from pilots [3, 4] and video gamer [4] participants. However, these data on the tool’s efficiency have only been qualitative. Thus, a pilot study was conducted to collect some preliminary quantitative data to characterize the effort required in employing this control method. In this study, only the noodle control was exercised; other FLEX-IT features were not employed. Specifically, data were collected on the time and accuracy in constructing noodles with varying complexities.

4.1 Noodle Construction Task

Six tracks (approximately 16.7 NM; 10.5 in centered on 22 in display) comprised of seven segments were designed and overlaid on the map of the simulator. Each track provided cues to the participants on what heading and altitude inputs were required during the construction of the noodle. Figure 4 illustrates a track that required both multiple heading and elevation changes. The change in the path's direction cued participants that a heading change would be required during noodle construction. Changes to the altitude were cued by the color of the path segment's borders (red: decrease altitude, white: maintain altitude, blue: increase altitude; all altitude changes were 500 ft). Changes in the heading and altitude were accomplished using the joystick. A throttle switch was used to change the length of the noodle and associated throttle switches were used to confirm or erase constructed noodle segments. Each track segment was designed to be approximately twice the distance of the default noodle length. Participants were not allowed to change the vehicle's airspeed setting.

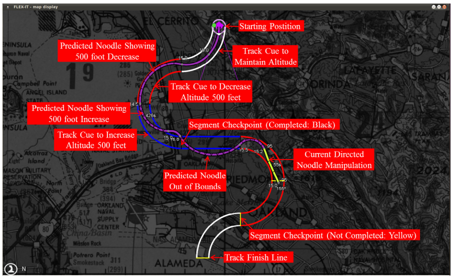

Fig. 4. Sample Segmented Track used during Experimental Trials

4.2 Experimental Design

Six different tracks were utilized varying in complexity in terms of the number of required changes in heading and/or altitude (see Table 1). Tracks that were mirror images of the six initially generated tracks were also used, to increase the variety of heading changes. Also, each of these 12 variations was presented both with the vehicle starting from the south and from the north (bottom and top of the display, respectively). These variations resulted in 24 track configurations (6 complexity levels X 2 mirror images X 2 directions) and each track configuration was considered an individual trial. The order of the 24 trials (i.e., 24 track configurations) was independently randomized for each participant in a within-subject design with the constraint that any track was not presented more than twice in a row and any other factors (e.g., direction flown) did not occur more than three times in a row.

Five volunteers (4 male, 1 female) with an average age of 34.8 years (SD=11.37) served as participants. All reported vision correctable to 20/20. None of the participants were rated pilots; none played more than 4 hours of video games per week.

Table 1. Number of heading and altitude changes commanded for each of the six track complexity levels as well as total absolute change in altitude across tracks

	TRACK COMPLEXITY					
	1	**2**	**3**	**4**	**5**	**6**
# Heading Changes	2	4	6	2	4	6
# Altitude Changes	0	0	0	3	4	5
# Altitude Increase	0	0	0	2	2	2
# Altitude Decrease	0	0	0	1	2	3
Total Absolute Altitude Change (ft)				1500	2000	2500

4.3 Procedures

Experimental sessions lasted 50-60 minutes with participants tested individually. After receiving training on the noodle construct, features of each controller, associated symbology, and test procedures, participants practiced constructing noodle paths for several vehicles using a “training track” with a variety of heading and altitude changes. This training took approximately 20 minutes to complete. Then, each subsequent trial was initiated by the experimenter’s input that caused a new track to be presented with the RPA symbol at the beginning of the track where it remained suspended for the duration of the trial. Recording of noodle construction time began when the participant activated a joystick switch that moved an indicator on the display to the “N” option. Participants were required to immediately start constructing a noodle that followed the track and overlaid each checkpoint (yellow lines dissecting the track) at the end of segments. (Once the accepted noodle passed a checkpoint, the yellow line turned black.) When a Directed Noodle segment was accepted the line became purple, denoting a Predicted Noodle. Upon completion of the new flight path (construction and acceptance past all checkpoints), the word “FINISHED” appeared in large text.

Participants were instructed that they could take as long as they wanted to construct the noodle, as well as employ any strategy they deemed most efficient (e.g., employ multiple short noodles versus fewer longer ones). However, they were also told that the time it takes for them to designate the future path of the RPA through the track using the noodle tool would be measured, as well as the extent to which the resulting path crossed outside the track’s borders. While the Directed Noodle (yellow line) could exit the track during construction, participants were instructed that the Predicted Noodle (purple; appears when Directed Noodle segment is committed with throttle switch selection) should be within the track’s borders. A questionnaire with 14 rating scales and 5 open-ended questions was administered after all trials were completed.

4.4 Results

Quantitative. Participants were very accurate in path construction: path was within borders on 97.5% of the trials. Mean time to construct the noodle was 62.84 s (range:

22.59-192.31). Results of a within-subject repeated measures 6 (complexity) X 2 (mirror images) X 2 (directions) analysis of variance (ANOVA) indicated a significant effect: mean noodle construction time significantly differed across complexity levels (Fig. 5, $F(5,20)=71.834$, $p=0$). Post-hoc tests indicated Track 6 with 6 heading and 5 altitude changes took significantly longer than others. The percentage of time savings realized by use of the noodle tool was estimated on each trial by subtracting construction time from the time it would take an RPA to hypothetically fly the designated route at the fixed airspeed (dividing that difference by the total flight time). An ANOVA of these data indicated that the time savings significantly decreased as the complexity of the track increased (Fig. 5; $F(5,20)=71.832$, $p=0$). Participants employed between 5 and 18 segments when constructing the noodle (mean=10.43, SD=3.69) and rarely used the erase feature (mean=0.267, SD=0.63).

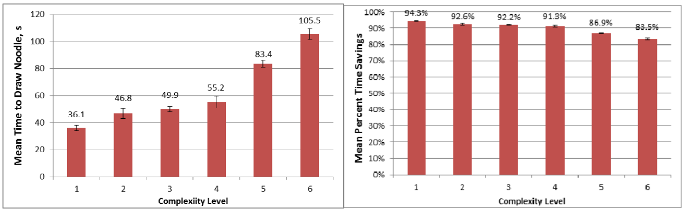

Fig. 5. Mean noodle construction time (left) and percentage of time savings for each track complexity level (right). Error bars are standard error of the mean.

Qualitative. The first questionnaire item asked for an overall rating on the participants' experience in constructing noodles (scale: very difficult, moderately difficult, moderately easy, very easy, or no opinion). Four of the five participants rated noodle construction as moderately or very easy. One participant selected 'moderately difficult' for this as well as 5 of 12 specific steps in creating the noodle, including changing heading, changing altitude, and heading south (as opposed to north). One other participant also indicated heading south was difficult, commenting that the southerly direction made it more likely to make erroneous left/right joystick inputs to change heading. Otherwise, the participants generally rated specific construction steps as either 'moderately easy' or 'very easy.' Four of the five participants indicated very little hand fatigue, while one selected 'somewhat.' Tracks that required a change in both altitude and heading for a segment were slightly more time consuming. There were also specific comments on some of the controls (e.g., too easy to activate the switch that erased a noodle segment). Participants' strategy in terms of the length of segments used in noodle construction varied.

5 Conclusions/Future Directions

The quantitative data collected in this effort compliments the qualitative data collected earlier indicating that this control mode for visualizing and commanding near-term future path is a definite candidate for future multi-RPA control. For example, the data demonstrates how the future flight path can be constructed very quickly (typically less than a minute). Comparing these data with an estimate of time to hypothetically fly the paths suggests a 90% attentional time savings for typical operations (albeit, the operator would still periodically check status of all vehicles even with the noodle method and would complete cross checks during some manual flight phases). Moreover, it is anticipated that pilots would construct noodles more quickly and realize additional time savings, compared to this study's non-pilot participants. Nevertheless, the ability to easily set the complex flight path of one vehicle should allow more time to be focused on other vehicles, enhancing supervisory control of multiple RPAs.

To apply this noodle concept operationally, however, there are several challenges still to be addressed. The predicted flight path is not likely to precisely match the actual path in the real world due to limitations in prediction models, effects of external disturbances such as significant winds, and inaccurate, delayed, or interrupted telemetry data. Near term applications will probably be more successful with high performance flight systems that are less impacted by external disturbances and have more sophisticated flight models and auto pilot systems. Another design issue to consider is how to represent the path's uncertainty to the operator. If the magnitudes of inaccuracies are known, then the predicted flight path could include augmented symbology that depicts the nature and source of the uncertainty, with an indication of how the degree of uncertainty increases with distance and time in the future.

Acknowledgments. Research was supported by AF/A2Q and managed by AFRL. Support from Ball Aerospace & Technologies Corp. and Smart Information Flow Technologies personnel was provided under Air Force Contract FA8650-08-D-6801.

References

1. Miller, C., Hamell, J., Barry, R., Ruff, H., Draper, M., Calhoun, G.: Adaptable Operator-Automation Interface for Future Unmanned Aerial Systems Control: Development of a Highly Flexible Delegation Concept Demonstration. In: Infotech@Aerospace 2012 Conference AIAA-2012, pp. 25–29 (2012)
2. Draper, M.H., Miller, C.A., Calhoun, G.L., Ruff, H., Hamell, J., Benton, J., Barry, T.: Multi-unmanned Aerial Vehicle Systems Control via Flexible Levels of Interaction: An Adaptable Operator-automation Interface Concept Demonstration. In: Infotech@Aerospace 2013 Conference (2013)
3. Calhoun, G.L., Draper, M., Ruff, H., Miller, C., Hamell, J.: Future Unmanned Aerial Systems Control: Feedback on Highly Flexible Operator-automation Delegation Interface Concept. In: Infotech@Aerospace 2012 Conference, AIAA-2012, pp. 25–49 (2012)
4. Calhoun, G., Draper, M., Miller, C., Ruff, H., Breeden, C., Hamell, J.: Adaptable Operator-automation Interface for Multi-unmanned Aerial Systems Control: Preliminary Usability Evaluation. In: Proc. Human Factors & Ergonomics Society (September/October 2013)