2013

Validating a Model of Automation Supporting the Robotic Arm Controller

Christopher D. Wickens

Angelia Sebok

A. Marc Gacy

Huiyang Li

Nadine Sarter

VALIDATING A MODEL OF AUTOMATION SUPPORTING THE ROBOTIC ARM CONTROLLER

Christopher D. Wickens, Angelia Sebok, A. Marc Gacy
Alion Science and Technology, Boulder, Colorado
Huiyang Li and Nadine Sarter
University of Michigan, Ann Arbor, Michigan

A simulation of the space robotic arm navigation task is described. This simulation is used in both a human-in-the-loop simulation experiment to generate human performance data, and is coupled with a computational model of the human: MORIS, whose outputs are compared to the human operator data for both nominal conditions at three levels of operation, and for automation-failure conditions. Scan mediated model predictions of automation failure response are validated by the human performance data.

This paper describes an effort to model the astronaut controller of a space-based robotic arm, such as that found on the space shuttle or International Space Station. Such an arm is designed to both grasp objects within its “hand” (called the end-effector or EE) and transport them to 3D points in the environment by manipulating arm joints of the shoulder elbow and wrist, along multiple degrees of freedom. Two 2-axis controllers are typically employed; one controlling rotation of the wrist and EE, and the other controlling 3D translation of the EE.

While carrying out the 3D navigation, the human operator must continuously be aware of constraints on the shoulder, elbow and wrist rotations to avoid what are called “singularity lockups” that freeze the arm in place, following which a time consuming recovery process is required. Naturally the operator must also monitor the arm in the workspace to avoid collision of wrist and elbow with hard constraints (e.g., obstructions) in the space.

As shown in figure 1, the operator monitors and controls the 4D trajectory (XYZ and time) through any of 6 cameras (two depicted in the figure), viewable through 3 different “viewports” or monitors, selecting at any time, those cameras that provide the best spatial understanding of the arm and EE relative to hazards and target destinations. The operator can also monitor joint angles on a separate display to assess their proximity to singularities and other abnormal states. A typical arm mission can be described in 3 phases: initiation of the appropriate movement; movement itself, and a final alignment and grasping (or releasing) of the payload by (from) the EE. Figure 1 presents a schematic layout of the workstation & workspace.

Many aspects of this task are analogous to the aircraft pilot, flying a 3 phase trajectory (departure, cruise, approach) while both navigating, and also preserving stability, with information provided by multiple displays. The manual operation of both flying and robotic arm manipulation can impose extremely high levels of cognitive and motor workload. For the robotics operator, this can be moderated by slowing or pausing the operation. However for the aircraft pilot, this workload has been mitigated by several layers of automation (Ferris, Sarter & Wickens, 2011). In particular, relevant to the current project we consider automation of guidance, via displayed vectors (e.g., a recommended flight path, much like the highway in the sky (HITS) display [Prinzel & Wickens, 1999]); and automation of control, akin to the coupled autopilot in the FMS, where trajectories can be flown merely by specifying XYZ endpoints. In both cases, reductions in workload and flight path error have been achieved, although in the case of the autopilot, the reduced workload comes at a cost of reduced situation awareness.

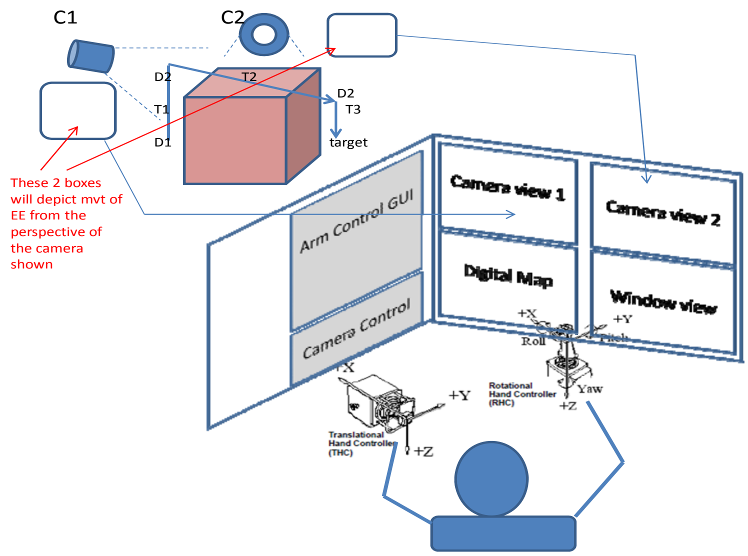

Figure 1. Schematic description of the robotic arm workspace (upper left) and displays. The figure depicts only two (rather than 6) camera views (C1 & C2) and a window view, and, within the workspace, a typical mission, to move the EE up, across a table and down to a target.

In marked contrast to the pilot, the robotic arm controller has not been well supported by corresponding layers of automation; and so our project has focused on development of automation support. This has been imposed in three ways. For **trajectory control**, we have created analogies to the two forms described for aviation: automation guidance, achieved by a 3D trajectory line through the workspace displayed on the camera view monitors, and full automation (autopilot) control. For hazard control, we have implemented an automatic collision warning system. For **camera control**, we have implemented intelligent guidance for the optimal camera view. Trajectory automation however will be the focus of the current paper.

We note that the two levels of trajectory automation (auto-guidance and auto-control) correspond closely to the levels of automation within the stages-levels taxonomy proposed by Parasuraman et al (2000) as an extension of the Sheridan and Verplank (1978) **levels of automation** scale. This distinction becomes of paramount importance, given the well-validated finding that the higher the level of automation, the better it works during normal operations and the lower the workload; but the **greater are the penalties when automation fails**. Onnasch, Wickens, Li and Manzey (submitted) carried out a meta-analysis of automation failure studies to validate this relationship, which they referred to as the "lumberjack analogy": like trees, *the higher they are, the harder they fall*. Thus one component of both our human-in-the-loop (HITL) simulation experiment, and our cognitive model predictions will examine this relationship between normal and failure performance.

Methods and Modeling

The overall content of this research endeavor included two parallel but interacting efforts: (1) At the University of Michigan, we developed a computer simulation of the robotic arm itself, modified from the original specifications of the system used to train astronauts at NASA, a simulation called BORIS. We employed this to gather HITL data of 36 well-trained subjects, operating the simulated arm under both nominal, and unexpected "failure" conditions, along a 3 segment trajectory that required multi-axis control and avoidance of a table hazard in the middle of the workspace (see figure 1). Details of these results are provided in Li et al (submitted). (2) At Alion Science, we developed a computer simulation model of the robotics operator using BORIS, a simulation which we called MORIS. The architecture, parameterization and validation of MORIS will be the focus of the current paper.

MORIS contains four linked sub-models, as shown in figure 2. At the left is a utility-based **decision model**, that decides, based upon maximum utility and pre-established rules, which modes to select, which trajectory to select and which cameras to choose for the two viewports. Input to the camera selection decision are outputs from the **FORT model** (Frame of Reference Transformation), which continuously computes the cognitive load of translating a given camera view into a control action (Wickens, Keller & Small, 2010). This model assigns penalties to the extent that a given view is closer to parallel to the line of sight into the display (McGreevy & Ellis, 1986; Wickens, Vincow & Yeh, 2005), and to the extent that the view provides EE motion information that is **incompatible** with the direction of control motion, or is hampered by poor visibility. In addition to deciding which camera to choose, the ubiquitous FORT model also influences the value or utility of each camera view to visual attention (via the SEEV model shown at the top of the figure, as discussed later), the fluency of control in the trajectory model, and provides an input to the perceived workload output of the model. The **trajectory model**, influences the fluency of actual control, and is heavily influenced by automation level (see below). Finally, a **visual attention model (SEEV**; Wickens, 2013; Wickens et al, 2003), predicts the pattern of eye movements across the 6 displays, based in particular on the **effort** to move attention, and the **expectancy** (bandwidth) and **value** of changes within each display (EEV within SEEV). The latter parameter is heavily determined by task relevance and the FORT based utility of each camera view of each display. As discussed below, expectancy and value are influenced by automation level

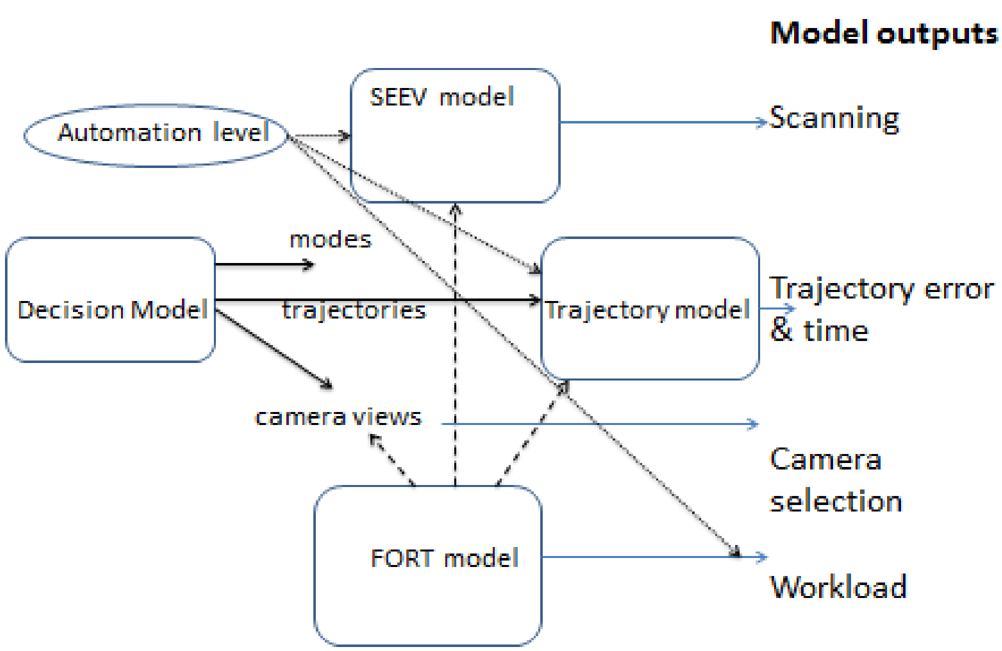

***Figure 2*:** Architecture of MORIS. Dashed lines represent FORT influences. Light blue lines show how model outputs are generated**.** The influences of automation level are dotted arrows.

Implicit in the model are three critical assumptions regarding automation and its linkage to SEEV parameters: (1) The highest level of automation (autocontrol or AC) directly diminishes the SEEV **value** of the camera monitors to the task of trajectory control (e.g., the relevance of those views to the navigational task). (2) Also in SEEV, the bandwidth or expectancy of the displays portraying EE motion is directly proportional to the tracking error observed from the empirical data at Michigan. Thus we assume that the higher error observed in the less automated conditions results from more frequent, and less stable corrections and requires more frequent sampling of the EE displays (Moray, 1986; Wickens, Goh et al., 2003). (3) We identified channel specific workload differences across the three trajectory automation conditions. Specifically, we predicted that *cognitive workload* in the autoguidance condition would be half the value in the manual condition. This is because in the manual condition, subjects needed to envision the correct trajectory between XYZ waypoints, whereas in the autoguidance condition these waypoints were directly displayed. Motor workload was little changed between the two conditions, as both required manual correction. In contrast, motor workload was assumed to be zero for the autocontrol condition, as neither response selection nor execution was required.

Results

Table 1 contains key aspects of the data from the Michigan simulation in the left of each column, that represent both the target values which we used to set parameters for the MORIS model as well as, for the two failure trials that occurred at the end of the experiment, targets for failure response validation, as discussed below. In the case of every dependent variable in the table, differences across the three levels were statistically significant ($p<.05$; see Li, Wickens et al., submitted). Presented in *italics* in the right side of the cells are the MORIS model predicted values, discussed below.

Table 1. Empirical HITL simulation data (left side of each cell) and MORIS model output (right side), as a function of trajectory automation level. Model data are not yet available for all measures.

Variable	Manual		Auto-guidance (AG)		Autocontrol (AC)	
Completion time (s)	440	*440*	401	*413*	215	*334*
Trajectory error	81		14		1.0	
Camera switches #	15.2		10.9		7.0	
Subjective workload	4.6	*4.6*	3.4	*3.8*	2.6	*3.0*
Trial 6 failure			**113**	***.40***	**144**	***.55***
Trial 7failure	**9/24 = .37**	***.37***	**8/24=.33**	***.40***	**20/24 =.83**	***.55***

(a) Parameterization

- Completion time is expressed as a value normalized to the maximum empirical value (manual condition). The correlation between obtained and observed completion time is 0.99
- The trajectory model simulated **tracking error** by adopting a threshold error, above which MORIS generated a linear closed loop correction to any path away from the target path. We assumed that this threshold was five times greater for the manual than the autoguidance condition, since in the manual case, the target path needed to be imagined or envisioned, whereas in the autoguided, it was directly visible. This is the basis of prediction of the two error measures for manual and AG. The autocontrol error was assumed to be close to 0, with a near perfect autopilot.
- The predicted number of camera switches was simulated based upon an algorithm in which, whenever computed FORT penalties for both camera views exceeded a key threshold, the currently unviewed camera with the lowest FORT penalty would be selected.
- Subjective workload was simulated by summing the predicted workload across (visual, cognitive, and motor) channels; based in large part upon the differing automation demands described above. However cognitive workload was also augmented by higher FORT values returned by the camera views across all three conditions, and by visual scanning. As with completion time, we normalized to the maximum (manual) value. The correlation between predicted and obtained workload measures was 0.99

(b) Validation: Automation Failure trials

- **Failure: trial 6**. On trial 6, a failure unique to the two automation conditions was imposed by depicting the guidance line (coupled, in the AC condition, with the actual flight path chosen by automation) along a path different from the correct direction to the final target on the third segment. Our measure of the ability to detect and correct this errant automation was the size of deviation from the correct path to the target.

- **Failure: trial 7**. On the final trial, for all three automation conditions, a proximity warning alert that had functioned correctly on all previous trials (including training) now failed, by remaining silent even as proximity limits were violated. This violation was guaranteed by directing the EE guidance too close to the table in the two automated conditions, and by providing XYZ coordinates for a corner turn that would yield a similar proximity violation trajectory in the manual condition. Our performance measure was a pooled measure combining the number of violations of proximity limits with the number of actual collisions with the table or wall hazard, both summed across subjects for this single trial.

We observe in the empirical data of both trials 6 and 7, a marked decrease in performance at the higher (AC) compared to the lower (AG) level of automation, consistent with the lumberjack analogy, and the better performance and reduced workload at the higher level, seen by measures in the upper rows of the table. We also note however that autoguidance automation does **not** induce poorer performance than manual automation on failure trial 7.

While the degree of model fit to the empirical data for the normal trials in the two upper rows for which model outputs were available was, to some extent expected, since we used those data to essentially "parameterize" the model, the same cannot be said for the failure trials. Here we made some basic assumptions grounded in eye movements and based upon "complacency theory" in human automation-interaction (Parasuraman & Metzger, 2007, Wickens, Dixon, Goh & Hammer, 2005). These assumptions allowed us to predict scanning behavior during normal trials on the basis of the SEEV model and use these to **infer** the manner in which automation-induced differences in scanning across the three conditions, would modulate fault detection ability. More specifically we assumed that (1) following the programmed failure deviation, ***a violation would occur if the trajectory was not manually corrected within 3 seconds***, and (2) ***complacency-induced scans*** away from the camera window where such deviation would be visually apparent, ***left that now-neglected area unattended***; hence this would create a human failure to notice the automation failure, if the eye did not return there before 3 seconds had elapsed. SEEV provided scan data, and in the SEEV model in the manual condition both expectancy and value were set to their maximum level, as described previously. In the autoguidance condition, expectancy was 1/3 maximum reflecting the large decrease in tracking error (see table 1) but value was retained at near its maximum level. In the auto control condition, both parameters were set to minimum. The SEEV scanning data and a noticing model (NSEEV) provided the probability of miss data shown in italics in the trial 7 failure trials; values that very closely approximated the obtained data, and a correlation of r=0.97 was obtained between model predicted and human generated data.

Discussion

In this paper we have presented the development of a computational computer simulation model of the human robotic arm controller. To our knowledge this is the first such effort. The model contains four submodels of spatial transformations, visual attention, decision making and trajectory control. For four outputs of the model, trajectory time and error, camera selection and workload, there was no a-priori basis for selecting parameters that would fit the experimental data from the HITL simulation of the arm controller. Hence agreement between predicted and obtained values was to be expected. However for one particular aspect, off-nominal automation failure response as a function of the level of automation in trajectory control, our effort produced something closer to a true (and successful) validation. We made a-priori assumptions of how level of automation would influence visual scanning (complacency) to critical areas where the automation failure would be noticeable, hence predicting failure response fluency. These model predictions were well validated. Additional empirical data will be sought to continue validation.

References

Ferris, T., Sarter, N., & Wickens, C. (2010). Cockpit automation: still struggling to keep up. In E. Salas and D. Maurino (Eds). Human Factors in Automation, 2nd ed. Amsterdam NL. Elsivier.

Li, H, Sarter, N., Wickens, C. & Sebok, A. (2013 submitted) Supporting Human-Automation Collaboration through Dynamic Function Allocation: The Case of Space Teleoperation. Submitted to Annual Proceedings Human Factors & Ergonomics Society.

Li, H., Wickens, C., Sarter, N., & Sebok, A. (submitted) Types and Levels of Automation in Support of Space Teleoperations. *Human Factors Journal*

Metzger, U., & Parasuraman, R. (2005). Automation in future air traffic management: Effects of decision aid reliability on controller performance and mental workload. Human Factors, 47, 35–49.

McGreevy, M. & Ellis, S., (1986)The effect of perspective geometry on judged direction in spatial information instruments. *Human Factors, 40*, 443-451.

Moray, N. (1986). Monitoring behavior and supervisory control. In L. K. K. R. Boff, & J. P. Thomas (Ed.), Handbook of perception and performance, Vol. II New York: Wiley & Sons.

Onasch, L., Wickens, C., Li, H. & Manzey, D. (submitted) Human Performance Consequences of Stages and Levels of Automation: An Integrated Meta-Analysis. Human Factors Journal.

Parasuraman, R., Sheridan, T. B., & Wickens, C.D. (2000). A model of types and levels of human interaction with automation. IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, 30, 286-297.

Prinzel, L. & Wickens, C.D. (2009) Preface: special issue on Synthetic Vision Systems. International Journal of Aviation Psychology, 19, 1-9

Sheridan, T. B., & Verplank, W. L. (1978). Human and computer control of undersea teleoperators. (Technical Report). Cambridge, MA: Man Machine Systems Laboratory, MIT.

Wickens, C.D (2013). Noticing events in the visual workplace: The SEEV and NSEEV models. In R. Hoffman & R. Parasuraman (Eds). Handbook of Applied Perception. Cambridge, U.K.: Cambridge University Press.

Wickens, C.D., Goh, J., Helleberg, J, Horrey, W., & Talleur, D.A. (2003). Attentional models of multi-task pilot performance using advanced display technology. Human Factors, 45(3), 360-380.

Wickens, C. D., Vincow, M., & Yeh, M. (2005). Design applications of visual spatial thinking: The importance of frame of reference. In A. Miyaki & P. Shah (Eds.), Handbook of visual spatial thinking. Oxford University Press.

Acknowledgments

This study was supported by a grant from the NASA Johnson Space Center Human Research Program (NNX09AM81G). Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of NASA. The authors wish to thank technical monitors Barbara J. Woolford and Douglas Wong for their support of and interest in this work.