Automation as alien: challenges for human factors

by Professor Thomas B. Sheridan

In 1951, in an article about designing a better air traffic control system, psychologist Paul Fitts explicitly laid out what "men are batter at" and what "machines are better at", which came to be called the MABA-MABA list. That list is now well out of date, as modern sensors and computers have now clearly exceeded human capabilities in many of the attributes Fitts awarded to humans. And over these sixty plus years automation and decision support tools have become standard fare for aiding and abetting human operators in aircraft navigation and landing, collision avoidance, weather prediction and avoidance, and other complex tasks.

However, while automation is touted by its hard core engineering designers as a friend or even saviour to controllers and pilots, it has made the task of human factors professionals responsible for making it work with real people ever more challenging. In a 1980 article in MIT Technology Review titled "Computer Control and Human Alienation" I pointed to a number of ways computer automation has alienated its users, who often do not understand how it functions, why it is doing what is doing, and in essence do not quite trust it. They admit it can do marvellous things, but sometimes expect it to know more than it really knows, and consequently develop unrealistic expectations that can get them in trouble, especially in off-nominal situations. "Father of cybernetics Norbert Wiener" made the point in his prize-winning book God and Golem Inc, the theme of which is that the computer, like the Golem monster of Hebraic tradition, does what it is programmed to do, not necessarily what its human users want and expect.

By pressing this button I will get the attention of everybody on the board... which is pretty awesome... since that is as far as our responsibilities go...

Tom Sheridan is Professor of Engineering and Applied Psychology Emeritus in the Department of Aeronautics and Astronautics at the Massachusetts Institute of Technology. After retirement he worked at the Volpe National Transportation Systems Center on human factors for aviation. He is former president of the Human Factors and Ergonomic Society. He is a private pilot, has an ScD from MIT, an honorary doctorate from Delft University in the Netherlands, and is a member of the US National Academy of Engineering.

The 2013 crash of Asiana 214 in SFO provides an example. According to the accident report: "In an attempt to

A SCALE OF "LEVELS OF AUTOMATION"

1. **Computer offers no assistance: human must do it all**
2. **Computer suggests many alternative ways to do the task**
3. **Computer prioritizes alternative ways to do the task**
4. **Computer recommends one way to do the task**
5. **Computer executes that recommendation when and if the human approves**
6. **Computer allows a restricted time for human veto prior to automatic execution**
7. **Computer chooses a method, executes and necessarily informs the human**
8. **Computer chooses a method, executes and informs the human only if requested**
9. **Computer chooses a method, executes and ignores the human**

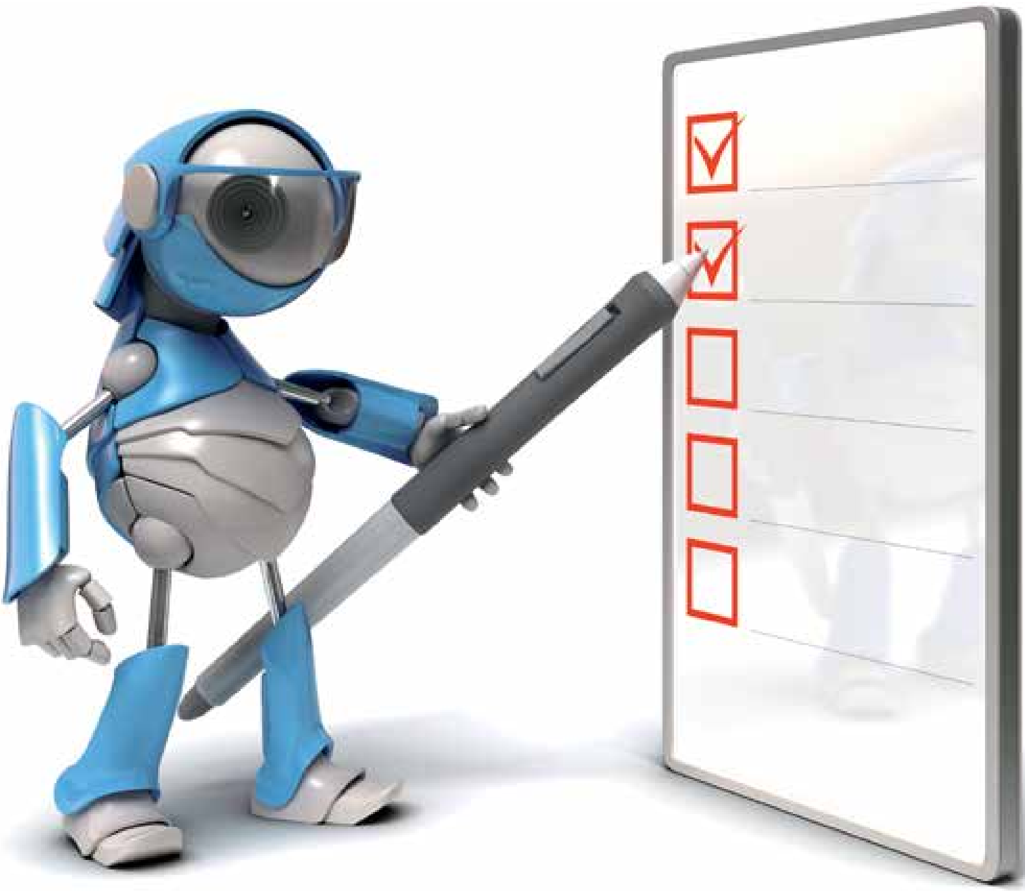
increase the airplane's descent rate and capture the desired glidepath, the pilot flying selected an autopilot mode (flight level change speed) that instead resulted in the autoflight system initiating a climb because the airplane was below the selected altitude. The pilot disconnected the autopilot and moved the thrust levers to idle, which caused the autothrottle to change to the HOLD mode, a mode in which the autothrottle does not control airspeed. The pilot then pitched the airplane down and increased the descent rate. Neither the pilot flying, the pilot monitoring, nor the observer noted the change in A/T mode to HOLD." Altitude, and then airspeed decreased, and at 100 feet an effort to initiate a go-around failed and the main landing gear and aft fuselage struck the SFO seawall.

"Expectancy, workload, fatigue and automation reliance" were blamed in the report. Those factors are all inter-connected, and it is also well established experimentally that some people just take much longer than the average to acquire sufficient situation awareness, make correct decisions and act properly, especially under stress. One answer to the Asiana 214 accident is more training to understand autothrottle modes and system activation logic, and adhere better to standard operating procedure. But there are also serious automation design issues—whether there is a level of complexity that is just too much for busy operators to comprehend and confidently use when the need arises (which may be rare!). Too many automation modes to accommodate and too many contingencies makes operators' mode awareness more difficult to maintain.

Stay in the loop or not?

Controllers, pilots and human factors professionals for years have debated under what circumstances operators should stay "in the loop" and whether there are inherent perils in making the human a supervisor of automation, a "flight manager". But this is not a binary choice. It is really a debate concerning what "level of automation" to invoke (see Table).

While it has been many years since autopilots and flight management systems were first introduced to aircraft, that level of automation has not yet come to air traffic controller workstations. Decision support tools at the 2-5 levels however, are appearing.



Automation as alien: challenges for human factors (cont'd)

For example, while continuous monitoring and vectoring are proven techniques, there are pressures to move controllers to a higher level of responsibility in coordinating with flight plans and traffic flow, allowing automation to spot incipient collision potential and alert the controller to attend to the screen and impose remedial action when the need arises. But then will there be sufficient time for the controller to drop some unrelated task, observe and understand the situation, make a decision and effect communications with the aircraft involved?

Computer-adaptive automation versus automation adaptable by human

There has also been much discussion within human factors circles recently about adaptive automation versus adaptable automation. In adaptive automation, the automatic control or information processing/display works differently depending on aircraft speed, altitude, attitude, traffic density, deviation from course, or some other measured parameter, independent of the operator. On the other hand, in adaptable automation the parameters must be changed by the human. The pressure from the computer community is always to make automation "smarter" (in the adaptive direction), but much research in human factors has shown that removing the human from the decision loop can produce reduced situation awareness, complacency, over-reliance on the automation, and unbalanced workload. When operator workload is too high there might be a situation where it would be desirable for automation to automatically take over control, but one problem is: how to measure workload quickly and reliably. If there were a well defined time window during which the human must perform a certain function, that is a situation where automation had better take control, hopefully to at least "buy time" for the human to recover. But again, can such situations be well defined, and if so how long should the automation wait before seizing control?

Authority and responsibility

The implication from much research is that some intermediate level between full automation (what is possible) and full human control is best. However, as more automation creeps into aircraft and air traffic control systems, and complexity necessarily increases, what is the degree to which human pilots and controllers are responsible if events go awry? If the automation hardware fails that is usually detectable, and the automation (or its designer, installer, maintainer, etc.) can be blamed. More often the situation is murky: an unusual weather or traffic situation, software that may not have been designed for exactly what occurred, a slight misunderstanding by the humans involved as to what the automation knew, was doing, was capable of, or how to manage it. Making provision for just shutting off the automation and assuming direct manual control might seem like an easy solution, but it takes time for humans to figure out what has gone wrong and to recover control, in some cases much longer than the system designers expect. It seems to me that system developers need some automation policy with clear guidelines allocating authority and responsibility.

LESSONS

1. Knowledge of automation activation logic. Try your best to understand the logic of how the automation works. If it is not understandable at an operational level, or if documentation/training is missing or inadequate, complain.

2. How much to trust. Developing appropriate trust in an alien being like automation requires training, time and interaction. Be conscious of what you can reasonably expect from the automation and what its limits are.

3. Getting back into the loop. Use of the automation often requires your being "out of the loop." So be sure you know how, if the need arises, to reinsert yourself in the loop to re-establish direct manual control, and how much time is necessary to do this.

4. Adaptive vs adaptable control. Currently very little adaptive control (as described above) operates within aviation automation; it is essentially all adaptable, meaning the pilot/controller is responsible for setting what mode the automation is in. For multi-model automation be conscious of what mode has been set in.

5. Be kind to your friendly human factors colleague. These folks work at the intersection between the pilot/controller user and the technical automation engineer. They are advocates for the user, and are pleased to get your feedback. But they necessarily do so with awareness of the realistic limitations of automation capability, operator training and cost.