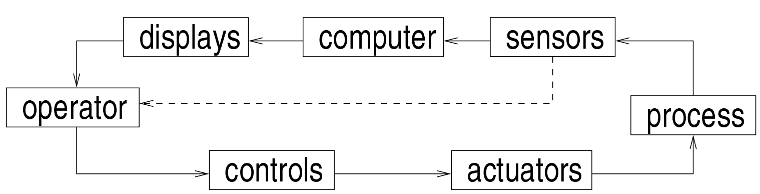
Human−Computer Interaction

[The designers] had no intention of ignoring the human factor ... But the technological questions became so overwhelming that they commanded the most attention.

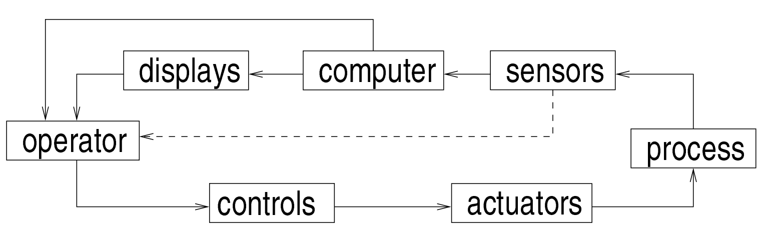
John Fuller
Death by Robot

Possible Roles for Computers in Control Loops

- Computer reads and interprets sensor data for operator

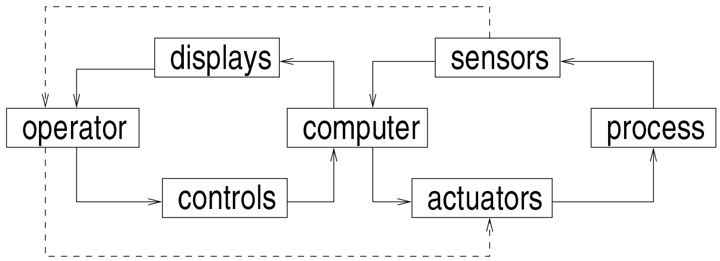


- Computer provides information and advice to operator



More Roles for Computers in Control Loops

- Computer interprets and displays data for operator and issues commands; operator makes varying levels of decisions.



- Computer assumes complete control with operator providing advice or high−level supervision or simply monitoring.

Role of Humans in Automated Systems

- The Human as Monitor
  - Task may be impossible
  - Dependent on information provided
  - State of information more indirect
  - Failures may be silent or masked
  - Little active behavior can lead to lower alertness and vigilance, complacency, and overreliance.

Role of Humans in Automated Systems (con't.)

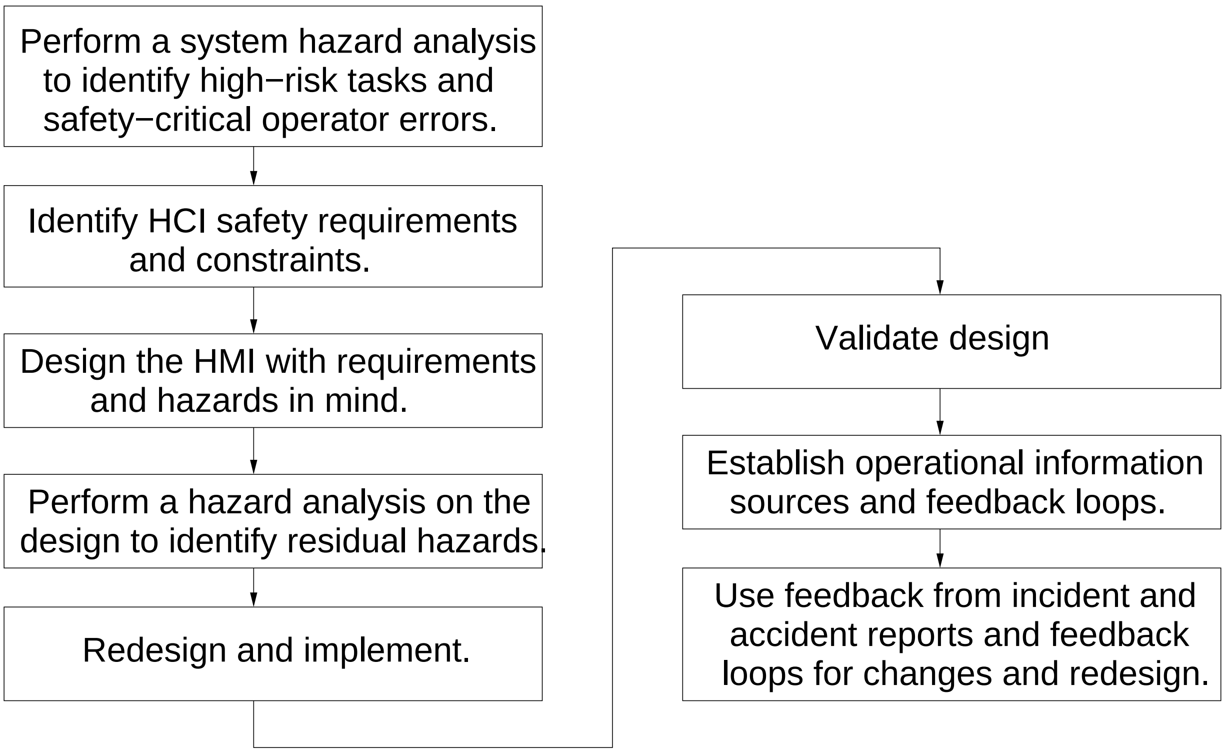
- The Human as Backup
  - May lead to lowered proficiency and increased reluctance to intervene
  - Fault intolerance may lead to even larger errors
  - May make crisis handling more difficult

- The Human as Partner
  - May be left with miscellaneous tasks
  - Tasks may be more complex and new tasks added
  - By taking away easy parts, may make difficult parts harder

HMI Design

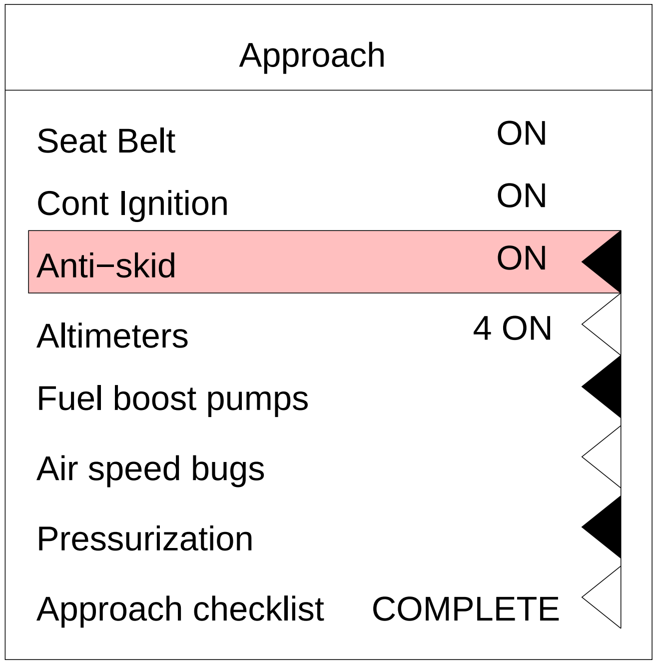
- Simple solution is to automate as much as possible, but is this the best solution?
- Different is not necessarily better.
- Need to consider conflicts between HMI design qualities.
- Norman: Appropriate design should:
  - Assume the existence of error.
  - Continually provide feedback.
  - Continually interact with operators in an effective manner.
  - Allow for the worst situation possible.

HMI Design Process



Matching Tasks to Human Characteristics

- Tailor systems to human requirements instead of vice versa.
- Design to withstand normal, expected human behavior.
- Design to combat lack of alertness.
- Design for error tolerance:
  - Help operators monitor themselves and recover from errors.
  - Provide feedback about actions operators took and their effects.
  - Allow for recovery from erroneous actions.



Allocating Tasks

- Design considerations.
- Failure detection.
- Making allocation decisions.
- Emergency shutdown.

Reducing Human Errors

- Make safety enhancing actions easy, natural, and difficult to omit or do wrong.

  Stopping an unsafe action or leaving an unsafe state should require one keystroke.

- Make dangerous actions difficult or impossible.

  Potentially dangerous commands should require two or more unique actions.

- Provide references for making decisions.

Reducing Human Errors (2)

- Follow human stereotypes.
- Make sequences dissimilar if need to avoid confusion between them.
- Make errors physically impossible or obvious.
- Use physical interlocks (but be careful about this).

Providing Information and Feedback

- Analyze task to determine what information is needed.
- Provide feedback:
  - About effect of operator's actions
    - To detect human errors
  - About state of system
    - To update mental models
    - To detect system faults
- Provide for failure of computer displays (by alternate sources of information.

  Instrumentation to deal with malfunction must not be disabled by the malfunction.

Providing Information and Feedback (2)

- Inform operators of anomalies, actions taken, and current system state.
- Fail obviously or make graceful degradation obvious to operator.
- Making displays easily interpretable is not always best.

Feedforward assistance:

- Predictor displays
- Procedural checklists and guides (be careful)

Alarms

- Issues:
  - Overload
  - Incredulity Response
  - Relying on as primary rather than backup (management by exception)

- Guidelines:
  - Keep spurious alarms to a minimum.
  - Provide checks to distinguish correct from faulty instruments.
  - Provide checks on alarm system itself.
  - Distinguish between routine and critical alarms.
  - Indicate which condition is responsible for alarm
  - Provide temporal information about events and state changes.
  - Require corrective action when necessary.

Training and Maintaining Skills

- May need to be more extensive and deep.

  Required skill levels go up (not down) with automation.

- Teach how the software works.

- Teach about safety features and design rationale.

- Teach for general strategies rather than specific responses.

Mode Confusion

- General term for a class of situation−awareness errors
- High tech automation changing cognitive demands on operators

    Supervising rather than directly controlling

    More cognitively complex decision making

    Complicated, mode−rich systems

    Increased need for cooperation and communication

- Human−factors experts complaining about technology−centered automation

    Designers focus on technical issues, not on supporting operator tasks

    Leads to "clumsy" automation

- Errors are changing, e.g. errors of omission vs. commission

---

Mode Confusion (2)

- Early automated systems had fairly small number of modes.

    Provided passive background on which operator would act by entering target data and requesting system operations.

- Also had only one overall mode setting for each function performed.

    Indications of currently active mode and of transitions between modes could be dedicated to one location on display.

- Consequences of breakdown in mode awareness fairly small.

    Operators seemed able to detect and recover from erroneous actions relatively quickly.

Mode Confusion (3)

- Flexibility of advanced automation allows designers to develop more complicated, mode−rich systems.
- Result was numerous mode indications spread over multiple displays each containing just that portion of mode status data corresponding to a particular system or subsystem.
- Designs also allow for interactions across modes.
- Increased capabilities of automation create increased delays between user input and feedback about system behavior.
- These changes have led to:
  - Increased difficulty of error or failure detection and recovery
  - Challenges to human’s ability to maintain awareness of

    active modes
    armed modes
    interactions between environmental status and mode behavior
    interactions across modes

Mode Confusion Analysis

- Identify ‘‘predictable error forms’’

  accidents and incidents
  simulator studies

- Model blackbox software behavior
- Identify modeled software behavior likely to lead to operator error.
- Reduce probability of error occurring:

  Redesign the automation
  Design appropriate HCI
  Change operational procedures and training

Design Flaws

1. Interface interpretation errors
   - Software interprets input wrong
   - Multiple conditions mapped to same output

Mulhouse (A320):

Crew directed automated system to fly in TRACK/FLIGHT PATH mode, which is a combined mode related both to lateral (TRACK) and vertical (flight path angle) navigation. When they were given radar vectors by the air traffic controller, they may have switched from the TRACK to the HDG SEL mode to be able to enter the heading requested by the controller. However, pushing the button to change the lateral mode also automatically changes the vertical mode from FLIGHT PATH ANGLE to VERTICAL SPEED, i.e., the mode switch button affects both lateral and vertical navigation. When the pilots subsequently entered "33" to select the desired flight path angle of 3.3 degrees, the automation interpreted their input as a desired vertical speed of 3300 ft. Pilots were not aware of active "interface mode" and failed to detect the problem. As a consequence of too steep a descent, the airplane crashed into a mountain.

— Operating room medical device:

The device has two operating modes: warmup and normal. It starts in warmup mode whenever either of two particular settings are adjusted by the operator (anesthesiologist). The meaning of alarm messages and the effect of controls are different in these two modes, but neither the current device operating mode nor a change in mode are indicated to the operator. In addition, four distinct alarm−triggering conditions are mapped onto two alarm messages so that the same message has different meanings depending on the operating mode. In order to understand what internal condition triggered the message, the operator must infer which malfunction is being indicated by the alarm.

— Display modes: In some devices, user−entered target values interpreted differently depending on active display mode.

Design Flaws

2. Inconsistent behavior

- Harder for operator to learn how automation works
- Important because pilots changing scanning behavior

– In go−around below 100 feet, pilots failed to anticipate and realize autothrust system did not arm when they selected TOGA power because it did so under all other circumstances where TOGA power is applied (found in simulator study of A320).

– Cali

– Bangalore (A320): a protection function is provided in all automation configurations except the ALTITUDE ACQUISITION mode in which autopilot was operating.

Design Flaws

3. Indirect mode changes

- Automation changes mode without direct command
- Activating one mode can activate different modes depending on system status at time of manipulation.

Bangalore (A320):

Pilot put plane into OPEN DESCENT mode without realizing it. Resulted in aircraft speed being controlled by pitch rather than thrust, i.e., throttles went to idle. In that mode, automation ignores any preprogrammed altitude constraints. To maintain pilot−selected speed without power, automation had to use an excessive rate of descent, which led to crash short of runway.

How could this happen?

Three different ways to activate OPEN DESCENT mode:

1) Pull altitude knob after select lower altitude.
2) Pull speed knob when aircraft in EXPEDITE mode.
3) Select a lower altitude while in ALTITUDE ACQUISITION mode.

Pilot must not have been aware that aircraft was within 200 feet of previously entered target altitude (which puts into ALTITUDE ACQUISITION mode). Thus may not have expected selection of lower altitude at that time to result in mode transition. So may not have closely monitored his mode annunciations. Discovered what happened at 10 secs before impact −− too late to recover with engines at idle.

Design Flaws (2)

4. Operator authority limits

   - Prevents actions that would lead to hazardous state
   - May prohibit maneuvers needed in extreme situations

- Warsaw
- During one A320 approach, pilots disconnected the autopilot while leaving the flight director engaged. Under these conditions, the automation provides automatic speed protection by preventing aircraft from exceeding upper and lower airspeed limits. At some point during approach, after flaps 20 had been selected, the aircraft exceeded airspeed limit for that configuration by 2 kts. As a result, the automation intervened by pitching the airplane up to reduce airspeed back to 195 kts. The pilots, who were unaware that automatic speed protection was active, observed the uncommanded automation behavior. Concerned about the unexpected reduction in airspeed at this critical phase of flight, they rapidly increased thrust to counterbalance the automation. As a consequence of this sudden burst of power, the airplane pitched up to about 50 degrees, entered a sharp left bank, and went into a dive. The pilots eventually disengaged the autothrust system and its associated protection function and regained control of the aircraft.

Design Flaws (2)

5. Unintended side effects

   - An action intended to have one effect has an additional one

Because approach is such a busy time and the automation requires so much heads down work, pilots often program the automation as soon as they are assigned a runway.

In an A320 simulator study, discovered that pilots were not aware that entering a runway change AFTER entering the data for the assigned approach results in the deletion of all previously entered altitude and speed constraints even though they may still apply.

Design Flaws (2)

6. Lack of appropriate feedback

   - Operator needs feedback to predict or anticipate mode changes
   - Independent information needed to detect computer errors

Bangalore (A320): PF had disengaged his flight director during approach and was assuming PNF would do the same. Result would have been a mode configuration in which airspeed is automatically controlled by the autothrottle (the SPEED mode), which is the recommended procedure for the approach phase. However, the PNF never turned off his flight director, and the OPEN DESCENT mode became active when a lower altitude was selected. This indirect mode change led to the hazardous state and eventually the accident. But a complicating factor was that each pilot only received an indication of the status of his own flight director and not all the information necessary to determine whether the desired mode would be engaged. The lack of feedback or knowledge of the complete system state contributed to the pilots not detecting the unsafe state in time to reverse it.

Example: Oops, It Didn't Arm (Everett Palmer, NASA Ames)

"Automation surprises" occur when the automation behaves in a manner that is different from what the operator is expecting. In the following case, a reasonable sequence of pilot actions performed in a high workload situation resulted in an unusual and undesirable result −− an automation surprise. In this case, the automation worked as designed.

Altitude deviations are the most common incident reported to the Aviation Safety Reporting System −− they are reported at the rate of about one per hour. In the following case, an automatic mode transition leads to a altitude deviation. This is such a common problem that it has been given a name, a "kill−the−capture" bust. Hundreds of similar altitude deviations have been reported to the ASRS. The only unique thing about this incident is that it occurred during a full−mission simulator study and was recorded for later analysis.

The incident took less than 20 seconds to play out. The crew had just made a missed approach and had climbed to and leveled at 2,100 feet. They received the clearance to "climb now and maintain 5000 feet " The Flight Mode Annunciator (FMA) showed:

	THRUST	ARM	ROLL	PITCH
A. Level at 2100 ft.	SPD 186		VOR CAP	ALT HLD

They received the clearance to :"... climb now and maintain 5000 feet ..." After some communication confusion on the cleared altitude (5,000 or 15,000) and which radial to hold on (0−0−6 or 0−6−0), the Captain set the MCP altitude window to 5,000 feet,

	THRUST	ARM	ROLL	PITCH
B. Enter 5000 in MCP	SPD 186	ALT	VOR CAP	ALT HLD

set the autopilot pitch mode to vertical speed with a value of approximately 2,000 feet per minute,

	THRUST	ARM	ROLL	PITCH
C Set VERT/SPD.	SPD 186	ALT	VOR CAP	VERT SPD

and set the autothrottle to SPD mode with a value of 255 knots.

	THRUST	ARM	ROLL	PITCH
D. Enter 255 in MCP speed window	SPD 255	ALT	VOR CAP	VERT SPD

Climbing through 3,500 feet, the Captain called for flaps up and at 4,000 feet he called for slats retract. As the aircraft climbed from 4,000 to 5,000 feet, the first officer was copying the holding clearance. Climbing through 4,000 feet, the FMA showed:

	THRUST	ARM	ROLL	PITCH
E. Approaching 4000 feet.	SPD 255	ALT	VOR TRK	VERT SPD

Passing through 4000 feet, the Captain pushed the IAS button on the MCP. The pitch mode became IAS and the autothrottles went to CLAMP mode.

	THRUST	ARM	ROLL	PITCH
F. Push IAS	CLMP	ALT	VOR TRK	IAS

Altitude capture was still armed. Three seconds later, the autopilot automatically switched to altitude capture mode. The FMA arm window went blank and the pitch window showed ALT/CAP.

	THRUST	ARM	ROLL	PITCH
G. Automatic altitude capture	SPD 255		VOR TRK	ALT CAP

A tenth of a second later, the Captain adjusted the vertical speed wheel to a value of about 4000 feet a minute. This caused the pitch autopilot mode to switch from altitude capture to vertical speed.

	THRUST	ARM	ROLL	PITCH
H Adjust vertical speed	SPD 255		VOR TRK	VERT SPD

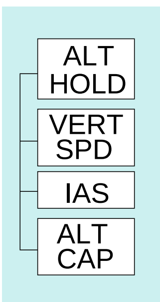
Climbing through 4500 feet, the approaching altitude light was on. As the altitude passed through 5,000 feet at a vertical velocity of about 4,000 feet per minute, the Captain remarked "Five thousand. Oops, it didn't arm." He pushed the MCP ALT/HLD button and switched off the autopilot. The aircraft continued to climb to about 5,500 feet and the "ALTITUDE − ALTITUDE" voice warning sounded repeatedly.

Exercise: What was the problem in the automation design that led to this incident?

Comments: As in many incidents involving automation, the error was first detected by the pilots not by using the autoflight displays such as the Flight Mode Annunciators that tell the state of the automation, but by the basic aircraft displays such as the alitmeter and the vertical speed indicator. The crew was apparently aware of the state of the aircraft but not aware of the state of the automation. The error was detected by observing the unexpected state of the basic aircraft displays, not the automation display. Woods has observed that most errors that result from the use of automation are detected by observing the system response and not the automation mode display.

In this incident, the automation display (the FMA) indicated what was actually happening; however the immediate response of the aircraft and the primary aircraft instruments were normal. The unusual and unexpected aircraft behavior occurred later. Although this is an error tolerant system, error detection was delayed beyond the point where that was possible. Why might this be the case? What makes it difficult to use the information in the FMA to verify the correct autoflight mode? A number of possible reasons. First, the FMA must be read and its meaning interpreted. Sometimes what must be "read" and interpreted is the absence of information. Second, the FMA's physical location away from the MCP requires that the pilot act in one place and check the outcome of the action in another place. Finally, the FMA does not provide a direct display of what the pilot needs to know to stay ahead of the aircraft, i.e., What trajectory have I set up the automation to fly the aircraft on?

PITCH MODE



→ VRT SPD

AND	OR	
AP in−mode On	T	T
Pilot adjusts V/SPD wheel	T	
Pilot pushes V/SPD button		T

RESULT:
Change Pitch annunciator to VRT SPD

→ ALT HOLD

AND	OR	
AP in−mode On	T	T
Pilot pushes HOLD	T	
Alt acquired		T
In−mode ALT CAP		T

RESULT:
Change Pitch annunciator to ALT HOLD

→ IAS

AND	
AP in−mode On	T
Pilot pushes IAS	T

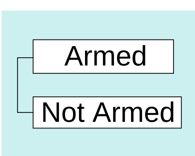
RESULT:
Change Pitch annunciator to IAS
Autothrottle goes to CLAMP mode

→ ALT CAP

AND	
AP in−mode On	T
Condition to start leveling off	T
Capture in−mode Armed	T

RESULT:
Start leveling off
Change Pitch annunciator to ALT CAP

CAPTURE MODE



→ Not Armed

AND	OR	
Capture in−mode Armed	T	T
Pilot pushes ALT	T	
Pitch in−mode ALT CAP		T

RESULT:
Change Arm annunciator to blank

→ Armed

AND	OR	
Capture in−mode Not Armed	T	T
Pilot sets new higher alt	T	
Pilot pulls ALT		T

RESULT:
Change Arm annunciator to ALT