Special Issue Paper

Framing Automation and Human Error in the Context of the Skill, Rule and Knowledge Taxonomy

Journal of Cognitive Engineering and Decision Making
2024, Vol. 18(4) 318–326

DOI: 10.1177/15553434241241892
journals.sagepub.com/home/edm


Marinus M. (René) van Paassen[1], Annemarie Landman[1,2], Clark Borst[1], and Max Mulder[1]

Abstract

Automation errors may result in human performance issues that are often difficult to grasp. Skraaning and Jamieson (2023) proposed a taxonomy for classifying automation errors into categories based on the visible symptoms of design problems, so as to benefit the design of training scenarios. In this paper, we propose a complementary classification that is based on the mechanisms of human-automation interaction guided by Rasmussen's Skill, Rule and Knowledge (SRK) taxonomy. We identified four main failure classes and expect that this classification can support automation designers.

Keywords

human error, automation failure, artificial intelligence, human-automation interaction, system safety

Introduction

Failure in safety-critical systems can seldom be assigned to an isolated, single initiating event. This inherently follows from the design constraints of such systems; redundancy is often used to reduce risk of catastrophic failure, operator training, maintenance and certification systems mitigate effects of component failure. However, every so often we are reminded of the fact that our capabilities to foresee failure scenarios are limited. In Boeing's design and implementation of the MCAS system (Republic of Indonesia, 2019), designers and test pilots expected that a fundamental failure of that system would be handled by pilots with the runaway trim procedure documented in the checklists. Instead, the instrument failures that triggered the MCAS system failure also generated a slew of alarms and instrument errors, forcing the pilots to be occupied with diagnosing a multitude of problems by the time the MCAS failures manifested themselves.

Human operators of automated systems, including pilots, are needed at least to handle those cases where designers fail to accurately foresee the context in which automation will have to operate. We are asking operators to handle situations in which equipment and automation fails in ways that are unforeseen, and it is therefore appropriate to study how people react under these circumstances, which training interventions can support them (Landman et al., 2017, 2021), and to develop simulation-based training programs with appropriate automation and

[1]Delft University of Technology, Netherlands
[2]TNO, Netherlands

Corresponding Author:
Marinus M. (René) van Paassen, Aerospace Engineering, Delft University of Technology, Kluverweg 1, Delft 2629 HS, Netherlands.
Email: M.M.vanPaassen@TUDelft.nl

equipment failures. Such training programs continue to be needed, even when advanced automation supported by AI applications are introduced into the workplace (Endsley, 2023).

Skraaning and Jamieson (2023) have proposed a taxonomy for automation error and argue its use for human-automation interaction research. As summarised, this taxonomy classifies automation-induced human challenges into three categories: (1) elementary automation failures, which involves isolated failures of automation (components); (2) systemic automation failures, which involves failures in the interplay among automation equipment, logic or functions, and (3) failures in human-automation interactions. A separate fourth category is described, for human and organisational slips and mishaps involving higher-order organisational issues.

Whereas the categories in the taxonomy by Skraaning and Jamieson (2023) are based on the visible symptoms of design problems, we propose a complementary classification that is based on the mechanism of human-automation interaction. Inspired by Reason's work (Reason, 1990), who used Rasmussen's Skill, Rule and Knowledge (SRK) taxonomy (Rasmussen, 1983) as a support in classifying and understanding human error, we take a new look at automation which involves a similar classification using the SRK taxonomy. We hope that this approach can not only be used in understanding and classifying automation failure, but that it will also give designers of automation a better understanding of the role of automation in a system with both human and automated agents, and can serve as a tool to critically inspect their designs.

In the following, we will discuss in what way a simple model inspired by the SRK taxonomy can be used to represent automation, and from there how failure mechanisms can be identified. Here, 'failure' will be both conventional device failure, such as introduced by faults in sensors, hardware or software, as well as the failure of functioning automation to provide its intended function.

Skill-, Rule-, Knowledge-Based Behaviour for Automation

The Skill-, Rule- and Knowledge (SRK) taxonomy (Rasmussen, 1986) has been applied to classify and understand modes of human behaviour. In this paper, we consider whether this same structure can be used to describe processes in typical automation systems. The SRK taxonomy is used to classify behaviour, but it also forms the basis of models of human task execution, such as the well-known ladder model (Rasmussen, 1986). A simpler model is used here, depicted on the left-hand side in Figure 1. At the skill-based level, two blocks or functions can be distinguished. One of these is labelled feature formation (S1), and describes the process of converting perceived signals into 'features'. These features can be used as signs for triggering activities, or symbols as input to cognitive evaluation. The second block represents interaction with the external world (S2).

At the rule-based level, signs, in their form of recognition of familiar states of the world, from the feature formation process at the skill level (block S1), can be recognised as triggers for action (R1), matched against appropriate actions (R2), and these actions or tasks can then be initiated (R3), to be carried out at the skill-based level, as learned, trained, sensorimotor patterns (S2).

Input to the knowledge-based level relies on symbols, meaningful information received and parsed in the skill-based feature extraction process (S1), that must be interpreted using knowledge about the work domain (K1), resulting in an updated image of that work domain. Comparison with selected goal states may result in decisions to influence the work domain (K2), leading to action plans (K3), that are, once formulated, executed as rule-based tasks.

This simple, three-level taxonomy has been used in interface design approaches (Vicente, 1999), or in error analysis (Reason, 1990). Naturally, engineered automation and control systems have their respective engineering descriptions, such as block diagrams and various response diagrams in control engineering (Nise, 2019; Ogata, 1997), or UML diagrams in software engineering (Object Management Group, 2024). In this paper, we propose a model for automation shaped after the SRK model, not as a replacement of engineering design tools and representations, but as a means to understand how human performance challenges may arise from automation. We will investigate how representing automation in terms of the SRK taxonomy may guide a search for mechanisms that may create human performance challenges.

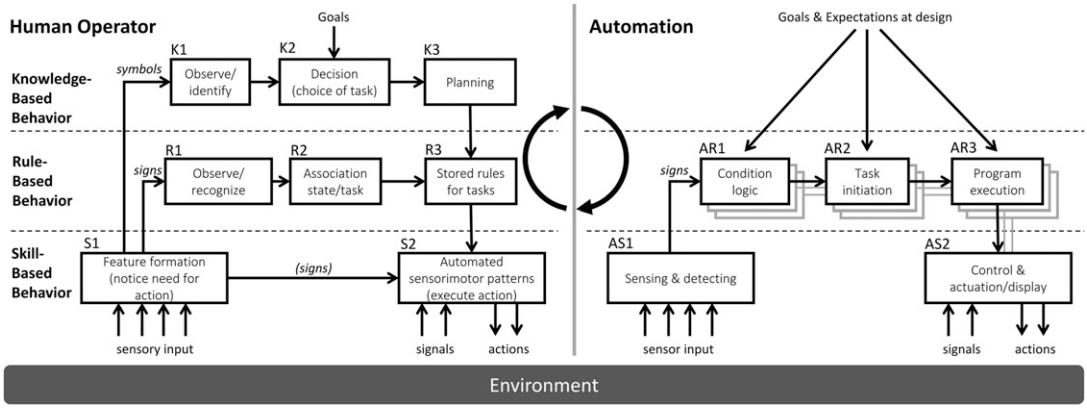

Figure 1. Rasmussen's SRK taxonomy for human cognitive control and automation behaviour.

Continuous Action: Control

The SRK-based taxonomy of automation (right-hand side of Figure 1) interacts with the work domain or external world at two locations; signals enter the 'feature formation' block (AS1), and signals and actions enter and exit the automated sensorimotor patterns block (AS2). In automation, such continuous interaction between the external world and the automation is characteristic for automated controllers, such as autopilots, and control augmentation systems, which intend to improve control properties of devices, as applied in military aircraft (e.g. F16 and F35), most Airbus aircraft and recent Boeing aircraft (B777 and B787). Analogous with applying the SRK taxonomy to human behaviour, continuous control by automation will be classified as a 'skill'.

There is a multitude of control system design tools available for designing and tuning automatic controllers. The advantage of these tools, and their accompanying concepts, is that description of performance criteria for the controller, and the conditions under which it should be functioning, do not need to be specified in terms of episodic knowledge, that is, time histories. Controllers can be designed to criteria such as bandwidth, robustness, disturbance rejection and the like. To a human user, a vehicle with control augmentation presents a new set of dynamic properties, in most cases better damped, less sensitive to external disturbances such as turbulence, or wind gusts, and easier to control. It can also protect a vehicle from unwanted inputs; traction control, anti-lock brakes and stability control in cars are examples of such control augmentation systems (SWOV, 2019).

Sensing: Feature Formation

One part of the human skill-based level is the feature formation. In addition to learning how to control systems, humans need to learn how to perceive and detect relevant data from the sensory input array. The equivalent process for automation would be the behaviour of sensors that convert physical quantities into signals, such as analogue voltages, or into discretized data, creating computer-transmittable information, with the associated signal conditioning, filtering and application of alarm thresholds. In the diagram on the right side of Figure 1, this is labelled as sensing and detecting (AS1). Newer automation, particularly in automotive, increasingly uses machine learning to process sensor information into signs and possibly symbols (Bachute & Subhedar, 2021).

Discrete Action

To describe how automation can be compared to rule-based behaviour, we will use the MCAS system developed for the Boeing 737 MAX as an example. This system was designed to compensate for stability deficiencies in this generation of Boeing 737 aeroplanes by initiating a pitch-down trim action when the aircraft is in or close to a stall. We can assume that this was the *intent* of its designers. For implementing the MCAS system automation, this intent was converted into a

number of actions and conditions. The signs for MCAS activation are threefold; first, the Angle of Attack (AoA) value exceeds a certain threshold; second, the autopilot is switched off; and third, the flaps are fully retracted. The original MCAS implementation would generate 8.5 s of nose-down trim in these conditions, and continue to repeat this while these conditions remain true. The updated implementation now adds the condition that there is no disagreement between the two AoA sensors (difference smaller than 5.5°), and that it has not previously activated during the same high AoA event, as Boeing's description suggests that the system resets and can only be re-activated once a high AoA alert has been cleared. The new implementation is thus more complex, and requires five signs for activation; high AoA, no AoA disagreement, no previous activation, flaps zero and autopilot off.

Analogous to human rule-level processes, the processes for automation (right, Figure 1) could be labelled as 'condition logic', 'task initiation' and 'program execution'. In the case of the MCAS, the condition logic would entail checking of the triggering conditions for MCAS activation (AR1), the task initiation would be activating the MCAS and initialising its timer (AR2) and program execution would provide a command for nose-down trim input for a given duration (AR3). Like the Boeing 737, operators of many advanced systems have to interact with multiple, more or less independent, automation systems. Even when implemented in a single hardware system, automation presents itself commonly as running all in parallel to human users. To illustrate that, the AR1 to AR3 blocks are depicted as a multitude of blocks. The resulting output of the rule level is sent to the control and actuation block (AS2), and consists of set-points and control modes for the automatic controllers.

Cognition and Reflection

One may wonder whether intelligent automation exists that should be modelled with the equivalent of (human) knowledge-based reasoning. This would involve maintaining a usable model for the state of the world and updating this model with incoming perceived data (Symbols). In a second step, it is checked whether this perceived state of the world is compatible with the set goals for the system or agent. If not, action is required, and based on (hypothesised) effects of possible actions on the world, and cost and compatibility of the actions with the user's goals, an action plan is then developed. Even descriptions of advanced self-driving systems do not indicate that this level of awareness and reasoning is implemented or achieved (Zhao et al., 2024).

As a hypothetical exercise, let's assume that a human had been observing the input signals to the MCAS system as it failed in the Lion Air and Ethiopian crashes. The AoA signal would have indicated a high AoA (note that this signal was *not* included in the pilot's instruments) from the start of the flight, and all while the flight progressed, the aircraft accelerated and climbed normally. Only when the autopilot is switched off, and the flaps are fully retracted, the condition for MCAS activation becomes true. However, in the context of the designer's goal, namely, correcting for stalls, and with knowledge of how an aircraft functions, having a continuous indication of stall during an otherwise successful take-off and climb out, would lead the human to conclude that there is a failure in the *sensor* rather than an actual stall. This knowledge-based reasoning component allows the human to reach this conclusion and prevent an MCAS action.

Current automation is designed with the designers' interpretation of the system's goal in mind, and with an imagined set of conditions or scenario which the designer considers as the possible context in which the automation should work. It can be safely said that, currently, Artificial Intelligence is by no means really 'aware' of the context, environment and goals, and for current applications, we thus need to only consider skill-based and rule-based aspects of the automation.

A knowledge-based level of control over the automation has to be assumed by human users or operators of a system, or it takes place in a largely open-loop fashion, with designers setting the goals for the automation, and implementing it to achieve these goals, possibly with slower updates, much akin to software updates.

Meaningful Control

In his book, Vicente uses the concept of open and closed system (Vicente, 1999). An open system

has a variety of interaction with the external world, and in many cases this interaction cannot be easily defined or measured. In contrast, a more closed system has limited interaction with the external world, and its interaction can be defined and measured. An elevator in a building is an example of a relatively closed system. It has an elevator shaft that is kept free of other objects, and can only open its doors at a defined number of floors. Since their interaction with the external world can be well-defined and measured, closed systems are good candidates for automation, resulting in low-risk, reliable automated operation. On the other hand, participating in street traffic for surface vehicles (cars), for example, requires interaction with an ever-changing and often poorly defined external world, making full automation extremely difficult.

Another factor that determines whether automation is acceptable is the risk that a system poses to its environment or users. Automated parts manufacturing, whether with 3D printing or in milling/machining shops, is an example of such an activity. A failure typically results in only a failed part, or in rare cases damage to the equipment. A failure in autonomous vehicles can hurt or kill passengers and other road users.

'Meaningful human control' is a concept originating in the application of ethics to largely automated weapons systems, extended to consider the application of Artificial Intelligence to other safety-critical domains. Cavalcante Siebert et al. (2023) state that human users should be able to act on their responsibility (property 3). In other words, automation fails if it prevents operators from achieving their (assigned) goals. In the following, we will use enabling meaningful control as a condition for automation success.

Automation Function and Failure

In this section, we will use a pair of SR(K) taxonomy models to inspect in what ways automation can achieve its intended functions, and in what ways it can fail at the sharp end, creating automation-induced human performance challenges (Skraaning & Jamieson, 2023).

In analysing the combination of human operators and automation, we will use the criterion whether meaningful control can be achieved. In other words, do human users/operators have a real possibility to control the system outcome and achievement of their goals?

Associated with that, for those tasks that are fully assigned to the automation, are the stakes of those tasks compatible with meaningful human control? That means that either the stakes are not critical, or the section of the work domain controlled by the automation has a sufficiently closed nature, and the automation is sufficiently reliable, so that failure risks are acceptable. To clarify, an example of the first might be the operation of the entertainment system on board of an airliner; failures might disappoint the passengers, but not threaten life or limb. An example of the second case is the common application of FADEC; Fully Automated Digital Engine Control, a control system for aircraft engines that keeps the engine within operating parameters. The reliability of these systems is such that there is no meaningful way by which manual direct control by the pilots (in which also risks of overspeed or excessive temperature of the engine need to be avoided) might improve the safety of operation.

Basic Function and Failure

In its most basic form, failure by an automation system, for example, through wear, breakage, external causes or errors in maintenance or installation, is comparable to the concept of skill-based error in human operators. Skill-based errors occur when an activity is attempted, but not successfully completed. In automation, that means that there is a failure in either data acquisition, data transformation, data transmission or in actuation. The most common problems arise through failing sensors, cameras, and the like, and in actuation through failing actuators or motors.

Compared to this, actual failures in calculation are rare, although they do occur, for example, in Qantas flight 72 (Australian Transportation Safety Board, 2011). Here, some data points were sent with the wrong data label in the communication within the electronics, resulting in transient failures in interpreting AoA data that, through their transient nature, could bypass data validation. This essentially combined a failure in algorithm design with a basic error.

Basic failure may also be introduced by failure of a support system, such as a failure of electrical, pneumatic or hydraulic power. From a technical point of view, it may be argued that sensors or actuators are not part of the automation proper. However, to operators, it is mainly the *functional* aspects of automation that matter, and any failure leading to loss of automation support or even erroneous behaviour is relevant and seen by them as 'failure of the automation'.

Design-Induced Failure

In a next step, the automation hardware may function as programmed and wired, but still fail to support the operators. This failure may be seen as analogous to rule-based error conditions, in which an automation action intended to support operation now counters the users' or even the designers' goals. One of the causes is a failure by designers to correctly foresee the context in which an automated system may operate. In the recent incident in which a door plug of a Boeing 737 MAX failed (Negroni, 2024), it also appeared that designers had installed a device to automatically open the flight deck access door in case of cabin pressure failure. As an unintended side effect, the door slammed into and blocked a lavatory door. The basic challenge is correctly foreseeing all possible situations under which automation may be active, as we tend to rely on narratives and envisioned scenarios. For open systems, that is a daunting challenge indeed.

Coordination Failure

Automation may also create undesired or dangerous situations when coordination with human operators or other automated agents is poor. A fundamental issue can arise when the information used by automation is not the same as information available to operators, for example, driving assistance based on radar information uses a different perspective than that of the human driver, potentially leading to poor understanding of automation actions.

Another issue may be that automation can be faster than human operators, leading it to operate in ways that are poorly observable by humans. This is acceptable, as long as no high stakes are involved, the automation is very reliable (by operating in a 'closed' part of the work domain), and is perfectly aligned with the human operator's goals. Phantom braking incidents in autonomous cars are an example of this issue. The basic failure in such incidents is often in the sensors or in the machine learning algorithm interpreting camera images. At the current level of autonomy (Level 2), the driver is assumed to intervene when automation is not successful (Myhre et al., 2019), although with phantom braking there is no quick way to override the car's inputs.

Poor observability of input signals and conditions to automation, and a complex logic to activation of automation, can also provide situations where human users can be surprised by automation actions. Mode confusion with autopilot and flight management systems is common (Albano et al., 2022), caused by the fact that many autopilot system modes are possible, and complex conditions guard the (often automated) transition between different modes.

Systemic Failure

In the previous sections, we discussed automation failures in isolation, and breakdown in the coordination between automation and human operators for single automation functions. However, in a typical case multiple automated functions vie for a human operator's attention, and may even interfere with each other. Older generation aircraft, such as the B737, are supported by notoriously fragmented automation. In a fatal accident with a Turkish Airlines flight (Onderzoeksraad voor de Veiligheid, 2010; Silva & Hansman, 2015), the auto throttle reduced power due to incorrect activation of the flare mode, while the autopilot system kept the aircraft on a vertical flight path, leading to a stall.

The two MCAS accident scenarios also have systemic failure effects. In these aircraft the angle sensed by the AoA sensor is used for the correction of aerodynamic pressure data, leading to a decreased altitude indication (by over 200 ft), and decreased speed indication (by 13 kts) on the captain's side. The speed indication on each side is also used to scale the stiffness in the pilot control manipulators through the feel system, and the differing speed inputs lead to a 'feel differential'

warning message. The indication from the AoA indicator on the captain's side also leads to a low airspeed warning, and to activation of the captain's stick shaker. Even though intended to be information displays to the pilot, one can also interpret this data processing as automation (Sheridan & Parasuraman, 2005). Feeding a wide range of automation and information functions with data from a single sensor introduces a common point of failure, leading to not only an erroneous MCAS activation, but to a challenging flight scenario in which different alarms and instrument discrepancies all compete for the pilots' attention, forcing them to prioritise between multiple checklists and procedures.

Each automation function, whether implemented separately in hardware, or implemented along with many others in a software program, must therefore be seen as an additional agent acting on the flight deck. Automation and information functions feed on thousands of input data points, and each piece of automation may combine multiple input data points as condition for activation. In general, pilots will not know the exact implementation of each automation function, and even for pilots with an engineering background and inquisitive minds, decoding surprising indications is more a debugging exercise than a result of finding the correct page in the manual (Huijbrechts & Van Paassen, 2023).

In older systems, or newer systems like the Boeing 737 Max that try to maintain compatibility with older generations, all these functions are independent, forcing a high workload on operators or pilots when a common failure occurs. In systems that have integrated automation, workload may be reduced because redundancy of input ensures that they are immune to single sensor failure that can carry over into multiple alarms, but the increased degree of integration also makes it more difficult to understand the automation's actions. For instance, in modern aircraft, typically the 'electronic checklist' will use pre-programmed rules to prioritise certain items, and the checklists will use a mix of 'sensed' items, that will automatically be updated in response to changes in input data points, and non-sensed items. Pilots will then need to understand both the behaviour of the aircraft, and integrated automation, but also of the actions of the automated checklists.

Conclusion

This paper presents a classification for automation error that can be seen as complementary to the work of Skraaning and Jamieson (2023). Using the Skill, Rule and Knowledge taxonomy (Rasmussen, 1983) as a template, automation processes can be identified analogous to human task performance. Four failure classes can be identified; 1. basic failures, that is, failures in the hardware implementation of the automation; 2. design failures, either because the automation was designed to fulfil a different goal than the user's goal, or due to the designer (team) limitations in predicting the future environment; 3. coordination failures, with issues in observability, understandability and controllability of the automation; and 4. systemic failure, typically due to complex interplay between the environment, multiple automation systems or multiple parts of the same automation, and the operators. Skraaning and Jamieson (2023) proposed the uses of their automation failure classification for the design of training scenarios. We expect that this complementary classification of automation failures can also support automation designers.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD

Marinus M. (René) van Paassen https://orcid.org/0000-0003-4700-1222

References

Albano, L. M., Fregnani, J. A. T. G., & De Andrade, D. (2022). Analysis of automation mode confusion with Brazilian airline pilots. *Journal of Aerospace Technology and Management*, *14*, Article e2822. https://doi.org/10.1590/jatm.v14.1280

Australian Transportation Safety Board. (2011). *In-flight upset 154 km west of learmonth, WA 7 October 2008 VH-QPA airbus a330-303 (ATSB Transport Safety*

Report No. AO-2008-070). Australian Transport Safety Bureau.

Bachute, M. R., & Subhedar, J. M. (2021). Autonomous driving architectures: Insights of machine learning and deep learning algorithms. *Machine Learning with Applications*, *6*, 100164. https://doi.org/10.1016/j.mlwa.2021.100164.

Cavalcante Siebert, L., Lupetti, M. L., Aizenberg, E., Beckers, N., Zgonnikov, A., Veluwenkamp, H., Abbink, D., Giaccardi, E., Houben, G.-J., Jonker, C. M., Van Den Hoven, J., Forster, D., & Lagendijk, R. L. (2023). Meaningful human control: Actionable properties for AI system development. *AI and Ethics*, *3*(1), 241–255. https://doi.org/10.1007/s43681-022-00167-3

Endsley, M. R. (2023). Ironies of artificial intelligence. *Ergonomics*, *66*(11), 1656–1668. https://doi.org/10.1080/00140139.2023.2243404

Huijbrechts, E.-J. A. M., & Van Paassen, M. M. (2023, June 1). *Flight Deck Procedures for a new Generation of Pilots*. 22nd International Symposium on Aviation Psychology, Rochester (NY).

Landman, A., Groen, E. L., van Paassen, M. M., Bronkhorst, A. W., & Mulder, M. (2017). Dealing with unexpected events on the flight deck: A conceptual model of startle and surprise. *Human Factors*, *59*(8), 1161–1172. https://doi.org/10.1177/0018720817723428

Landman, A., van den Hoed, A., Van Baelen, D., Stroosma, O., Van Paassen, M. M., Groen, E., & Mulder, M. (2021). *A procedure for inducing the leans illusion in a hexapod motion simulator* (pp. 1–7). American Institute of Aeronautics and Astronautics Inc (AIAA).

Myhre, B., Hellandsvik, A., & Petersen, S. (2019). A responsibility-centered approach to defining levels of automation. *Journal of Physics: Conference Series*, *1357*(1), 012027. https://doi.org/10.1088/1742-6596/1357/1/012027

Negroni, C. (2024, January 13). *Not just the 737 max - cockpit doors will open on most airliners [Christine Negroni]*. https://christinenegroni.com/not-just-the-737-max-cockpit-doors-will-open-on-most-airliners/

Nise, N. S. (2019). *Control systems engineering* (8th ed.). Wiley.

Object Management Group. (2024). *Welcome to UML web site! [Object management group]*. https://www.uml.org/

Ogata, K. (1997). *Modern control engineering* (3rd ed.). Prentice Hall.

Onderzoeksraad voor de Veiligheid. (2010, May). *Crashed during approach, Boeing 737-800, near Amsterdam Schiphol airport - 25 February 2009 (M2009LV0225_01)*. The Dutch Safety Board.

Rasmussen, J. (1983). Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. *IEEE Transactions on Systems, Man, and Cybernetics*, *13*(3), 257–266. https://doi.org/10.1109/tsmc.1983.6313160

Rasmussen, J. (1986). *Information processing and human-machine interaction: An approach to cognitive engineering*. Elsevier Science Inc.

Reason, J. T. (1990). *Human error*. Cambridge University Press.

Republic of Indonesia. (2019). *PT. Lion Mentari Airlines Boeing 737-8(MAX); PK-LQP Tanjung Karawang, West Java* (Aircraft Accident Investigation Report KNKT.18.10.35.04; p. 322). https://www.aaiu.ie/sites/default/files/FRA/2018-035-PK-LQPFinalReport.pdf

Sheridan, T. B., & Parasuraman, R. (2005). Human-automation interaction. *Reviews of Human Factors and Ergonomics*, *1*(1), 89–129. https://doi.org/10.1518/155723405783703082

Silva, S. S., & Hansman, R. J. (2015). Divergence between flight crew mental model and aircraft system state in auto-throttle mode confusion accident and incident cases. *Journal of Cognitive Engineering and Decision Making*, *9*(4), 312–328. https://doi.org/10.1177/1555343415597344

Skraaning, G., & Jamieson, G. A. (2023). The failure to grasp automation failure. *Journal of Cognitive Engineering and Decision Making*, *11*. https://doi.org/10.1177/15553434231189375.

SWOV. (2019, April 24). *Intelligente transport- en rijhulpsystemen (ITS en ADAS) (SWOV-Factsheet, April 2019)*. SWOV.

Vicente, K. J. (1999). *Cognitive work analysis: Toward safe, productive, and healthy computer-based work*. Lawrence Erlbaum Associates.

Zhao, J., Zhao, W., Deng, B., Wang, Z., Zhang, F., Zheng, W., Cao, W., Nan, J., Lian, Y., & Burke, A. F. (2024). Autonomous driving system: A comprehensive survey. *Expert Systems with Applications*, *242*, 122836. https://doi.org/10.1016/j.eswa.2023.122836.

Marinus M. (René) van Paassen, PhD, is an associate professor in Aerospace Engineering at Delft University of Technology, working on human machine interaction and aircraft simulation. His work on human-machine interaction ranges from studies of perceptual processes, haptics and haptic interfaces and human manual control, to the design of and interaction with complex cognitive systems, applying cognitive systems engineering and ecological interface design for vehicle control.

Annemarie Landman, PhD works at TNO, the Netherlands, as a researcher of cognitive performance in aviation and military operations. She also works part-time as a teacher at the Control and Operations section of the Delft University of Technology, where she graduated on the topic of startle and surprise. Her topics of interest include complex cognition under pressure, spatial disorientation, and training.

Clark Borst received the M.Sc. (2004) and Ph.D. degrees (2009) in aerospace engineering from the Delft University of Technology, The Netherlands, where he is currently working as an Assistant Professor in aerospace human-machine systems. His work and research interests include developing human-centered aviation automation through the application and empirical evaluation of Cognitive Systems Engineering and Ecological Interface Design principles.

Prof. Max Mulder, PhD, is full professor 'aerospace human-machine systems' at the faculty of Aerospace Engineering, Delft University of Technology. He leads the section 'Control and Simulation' which investigates reaching higher levels of automation in aviation.