



# A DISCRETE-EVENT SIMULATION MODEL FOR DRIVER PERFORMANCE ASSESSMENT: APPLICATION TO AUTONOMOUS VEHICLE COCKPIT DESIGN OPTIMIZATION

I. Iuskevich<sup>1,2,✉</sup>, A. M. Hein<sup>1</sup>, K. Amokrane-Ferka<sup>2</sup>, A. Doufene<sup>2</sup> and M. Jankovic<sup>1</sup>

<sup>1</sup>CentraleSupélec, France, <sup>2</sup>IRT SystemX, France

✉ [ilia.iuskevich@centralesupelec.fr](mailto:ilia.iuskevich@centralesupelec.fr)

## Abstract

The latest advances in the design of vehicles with the adaptive level of automation pose new challenges in the vehicle-driver interaction. Safety requirements underline the need to explore optimal cockpit architectures with regard to driver cognitive and perceptual workload, eyes-off-the-road time and situation awareness. We propose to integrate existing task analysis approaches into system architecture evaluation for the early-stage design optimization. We built the discrete-event simulation tool and applied it within the multi-sensory (sight, sound, touch) cockpit design industrial project.

*Keywords: model-based engineering, design optimisation, ergonomics, human-machine interface, autonomous vehicle*

## 1. Introduction

The accelerating technological progress rapidly changes the landscape of available design options in the automotive domain. Constant innovations in the design and functionality of an increasingly autonomous vehicle drive the evolution of the design of the vehicle cockpit. This happens because the increasing autonomy alters the status of the driver (authority of control is shared between driver and car). For example, in December 2017 Waymo LLC (the company is owned by Alphabet) has launched the commercial self-driving car service in Arizona. In their car most of the time driving tasks are performed by the car, though a trained driver is still required to take over in case of problems (Korosec, 2019).

In the manual mode the driver is included into the control loop. In other words, the driver has full control over the car's course (i.e. steering) and acceleration. Accordingly, HMI of such a car includes elements of direct control and stabilization – steering wheel, acceleration and brake pedal, tachometer, speedometer, etc.

At the highest level of autonomy the driver is fully excluded from the control loop, i.e. the driver becomes a passenger. Accordingly, the elements of control over the car's trajectory are eliminated. Between these two extremes there are countless interface design options for so-called semi-automated cars for which authority of control is shared between the human and machine (see spectrum of assistance and automation in (Flemisch et al., 2014)). Notable examples of such interfaces are the maneuver-based approach (Conduct-by-Wire), and haptic-multimodal approach (Horse-metaphor) (Flemisch et al., 2014).

Another practical semi-automated car design option is a vehicle with an adaptive level of automation. As [Scerbo \(2008\)](#) defined, in the automation is adaptive when “*the level of automation or the number of systems operating under automation can be modified in real time. In addition, changes in the state of automation can be initiated by either the human or the system*”.

Up to date, infrastructure and technologies are not mature enough to enable the operation of fully autonomous vehicles. Under these conditions, adaptive automation is a good solution compared to fixed task allocation (between the machine and a human), since it allows to avoid cognitive overloading of the driver, boost situation awareness, or reduce complacency ([Parasuraman and Wickens, 2008](#)) depending on the state of the driver or road conditions.

The major shortcoming of adaptive automation is the increased complexity of the user interface. Traditionally, a car has only interfaces for direct control. Future fully autonomous cars will have only interfaces that communicate the coordinates of the final destination to the machine. At the same time, an interface of a car with an adaptive level of automation maintains elements of direct control, includes some features of self-driving cars and, additionally, is augmented by elements of control over the level of automation (switchers, indicators of level of automation, take-over requests, etc.).

Accordingly, an interface for a car with an adaptive level of automation may increase workload and eyes-off-the-road time, especially at the moments of transition between automation levels. These negative effects can be mitigated by optimizing the HMI’s functional architecture.

This paper aims at integrating existing methods of workload and eyes-off-the-road time estimation into the system architecture evaluation process with the objective of optimizing the HMI architecture. The discrete-event simulation model was developed to perform a simulation and estimate HMI performances.

The remainder of the paper is structured as follows. In the next section we present the results of the literature review in the field of workload and eyes-off-the-road time modelling. Then, our proposed discrete-event simulation model is described. After that, we present the results of a case study. In the final section we discuss how the model can be extended and validated in future work.

## 2. Background

### 2.1. Mental workload modeling

Since the mid-80s, a lot of research efforts have been devoted to the development of the human operator mental workload modeling tools. Most of these approaches are based on the idea of task analysis, or, more specifically, task network. A task network is a functional decomposition of a human operator’s activities down to elementary tasks ([Laughery et al., 2000](#)). Then these elementary tasks are annotated with a number of attributes (descriptors), e.g. required workload, task duration, triggering event, etc. The total cognitive workload is then calculated either with a matrix-based approach (W/INDEX) ([North and Riley, 1989](#)) or as the result of a simulation in a discrete-event environment ([Aldrich et al., 1989](#)). More recent works in this field propose to employ Petri nets to model human operator strategies, adaptive to the changing environment ([Kontogiannis, 2005](#)).

[Boy \(1998\)](#) generalized task network approaches under the notion of Cognitive Function Analysis (CFA). This approach responds to the emergence of highly automated and cooperative systems. In CFA, complex cognitive functions may be allocated not only to a human, but also to a machine, which in turn is interacting with other automated machines. Accordingly, CFA is more suitable for the modern context where authority of control is shared between humans and machines.

In the past, all these approaches were driven mainly by increasing automation in military, aerospace or power plants domains. Nowadays, research focus is shifting to the automotive systems due to the progress in the domain of car automation.

### 2.2. Eyes-off-the-road time estimation

Compared to mental workload modeling, eyes-off-the-road time prediction approaches received less attention in the literature. However, this metric is important from a safety perspective in the automotive domain. The driver’s distraction is the leading defined cause of road accidents according

to (Wang et al., 1996). Up to 20% of crashes due to driver distraction are caused by the interaction with interior equipment (car interfaces or cellphone) (Green, 2017). These facts underline the importance of efforts to minimize the potential visual distraction caused by elements of user interface during the design optimization.

Wittmann et al. (2006) demonstrated experimentally that the location of the onboard display greatly influences the safety of driving, e.g. perception of the information on the head-up display causes shorter distraction compared to instrument cluster or central panel.

The integration of the Keystroke Level Model and occlusion technique was presented in Pettitt et al., (2007). The goal of this approach is to predict eyes-off-the-road time having a list of driving tasks as an input. The validation of this approach has shown high accuracy with experimental results.

### 2.3. Cognitive simulation models

Approaches based on task analysis treat the driver's cognition as a black-box (tasks play the role of inputs and outputs). In contrast, cognitive simulation methods model internal mental processes of a human. Without the model of the human cognition system it is impossible to evaluate another important safety-related metric – situation awareness.

The most comprehensive model of the performance of a human driver up to date is COgnitive Simulation MOdel of the DRIVER (COSMODRIVE) (Bellet et al., 2011). COSMODRIVE is composed of three modules – perception, cognition and action. It models in details all main mental activities of a human driver. It can be connected to the SiVIC virtual road environment platform to provide very detailed input to a perceptual model of the virtual driver.

A workload prediction method based on a cognitive architecture for safety critical task simulation (CASCaS) (Feuerstack et al., 2007) was used to assess the driver's workload in real-time, in order to enable adaptive automation (Wortelen et al., 2016).

In contrast to cognitive simulation models, approaches based on task analysis are significantly less complex. Therefore, they can be rapidly developed, they require less computing power and their results can be easily interpreted. Applying a task analysis approach to a new type of a system (i.e. vehicles with adaptive levels of automation) has not yet been demonstrated in the literature.

## 3. Discrete-event simulation model architecture

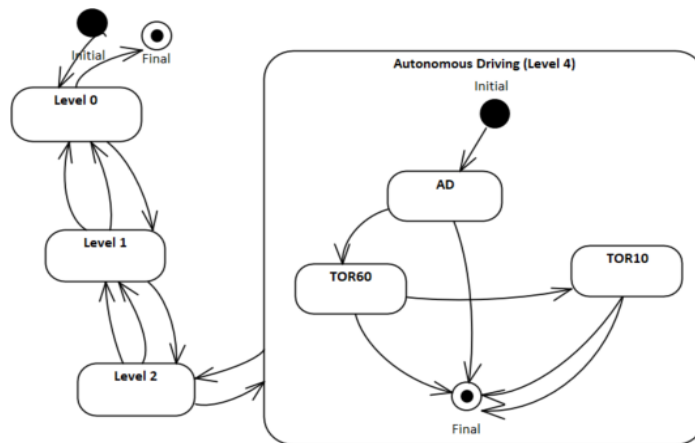
In this section we describe the architecture of our proposed discrete-event simulation model, task data structure and workload scales.

From a systems engineering point of view, the design of an autonomous vehicle cockpit is a challenging task even though the complexity of the physical architecture of this kind of a system is relatively low (compared to, for example, to aerospace systems). The complexity of the design of a cockpit rests on the ambiguity of the functional architecture which builds upon the ambiguity and complexity of the behavior of the human driver. The latter is complex, due to the fact that the number of possible human reactions is large and hardly formalizable, and ambiguous because there are a number of known unknowns such as level of experience, personal attitude, tiredness, mood, etc.

The objective is to compare functional architectures of a cockpit in terms of safety without the ambition to predict precise, absolute values of the key metrics. For example, if one variant of a cockpit constantly transmits a lot of unnecessary information to a driver, the critical information may be eventually missed. Another, less distracting variant of a cabin would be considered safer.

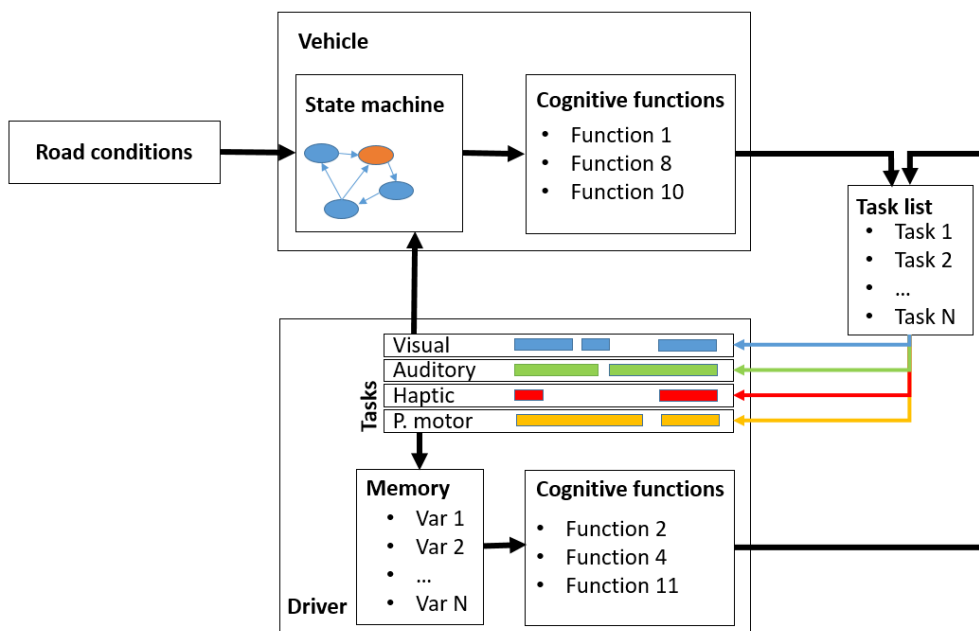
Our system of interest is a cabin of a car with adaptive levels of automation (Level 0-4). We will call Level 4 Automotive Driving (AD) mode in the following. A driver can switch to a higher level of automation at any moment if this level is available under specific road conditions (e.g. AD mode may not be available due to the absence of road markings). At any moment, the driver or the machine may switch from a higher to a lower level of automation if the higher level of automation is no longer available due to the road conditions.

A special mode called take-over request (TOR60) is activated 60 seconds before the moment when AD is no longer available. During this mode the interface sends signals to a driver to indicate to put hands on a steering wheel and switch to a lower level of automation. The state machine of the car of interest is shown on Figure 1.



**Figure 1. The state-machine of the car with adaptive automation**

The architecture of the human-machine interaction model is shown in Figure 2. It consists of 7 major elements: road conditions, the vehicle's state machine and cognitive functions, the driver's memory, task schedule and cognitive functions, and task list. Elements are communicating with each other by means of events. For the modeling we use Python discrete-event simulation library SimPy, due to its simple and detailed user manual.



**Figure 2. Architecture of the discrete-event simulation model of HMI**

An event is triggered in one element and influences the processes or state of another element. For example, changing road conditions may trigger the transition of the vehicle state (automation level).

Cognitive functions are modelled as random processes generating events in time. For example, “check the speed on average every 20 seconds” is a cognitive function. It triggers events normally distributed in time with a mean value of 20 seconds.

Tasks are modelled by complex objects, which are called processes in SimPy. A process is triggered by some event and activates within a limited interval of time. When a process is active it requires some defined amount of a limited resource. If the process is triggered but the resource is allocated to other processes, this process is put on the waiting list. Its position on the waiting list depends on its priority. Processes from the waiting list with a higher priority are executed first. After the process execution another event is generated, which triggers another processes, etc.

The SimPy object “resource” models the limited capacity of user cognition and perception. According to multiple-resource theory (Wickens, 2002), the human brain can perceive information from two sources efficiently if they require separate attentional resources. Seven separate attentional resources are used in our multi-sensory interface: visual (displays), visual peripheral (ambient lighting), auditory vocal and auditory non-vocal (speakers), haptic hands (steering wheel haptic actuator), haptic seat (seat haptic actuator) and psychomotor. We will use the simplest model of the resource conflict matrix, i.e. two tasks cannot use the same resource concurrently. The sum of the active tasks’ workloads in our model shall not exceed 10. This number was selected arbitrarily here, as the objective is to demonstrate the principle feasibility of the approach. However, we aim to replace it by an empirically validated workload limit in the future.

We describe each task as an excel row with fixed list of properties (Table 1).

**Table 1. Data structure of a task**

Name of property	Description
Name	Name of a task
Description	Textual description of a task
Location	Name of an interface element with which the driver should interact to accomplish a task (e.g. steering wheel, cluster, central console, etc), or, in other words, task-component allocation parameter
Cognitive workload descriptor	Textual description of cognitive task complexity
Perceptual workload descriptor	Textual description of perceptual task complexity
Perception type	Sensorial mode (Visual, Visual peripheral, Auditory Vocal, Auditory non-Vocal, Haptic hands, Haptic seat, Psychomotor)
Perceptual workload	Amount of perceptual workload required to accomplish the task (in a relative scale)
Cognitive workload	Amount of cognitive workload required to accomplish the task (in a relative scale)
Duration	Amount of time needed for task execution
Gaze Time	Time needed to change the visual focus from the road to the interface element (only for visual tasks)
Total time	Task duration plus gaze time multiplied by two
Cognitive function trigger	The name of CPF that triggers the task
Awareness parameter	The parameter in user memory that is updated after task execution
Triggers	The name of the task that shall be executed right after (if any)
Priority	Task relative importance (ordinal scale)

For example, the cognitive function “check the current speed approximately every 10 seconds” generates an event that triggers the task “check out the speedometer” with the “Inspect/Check (numerical)” perceptive descriptor and the “Evaluate single aspect” cognitive descriptor. This task requires 0.2 seconds for the gaze change from the road to the instrument and 1 second to accomplish the task. It takes 4.6 points of cognitive resources and 4.0 of visual perception resources. If there are

no other active visual tasks, the total cognitive workload is less than 10 and total perceptual workload less than 10, task is executed during 1.4 seconds. The accomplished task generates an event which triggers an update of the variable “current speed” in the object representing the driver’s memory.

In contrast to tasks initiated by the user (e.g. “check values in the display”, “change automation level”), tasks initiated by the machine (e.g. “send vocal message to the user”) cannot be put on the waiting list. If such a task is triggered and cognitive or perceptual resources of the driver are allocated, then the task is aborted immediately.

We are using the workload component scale presented in Aldrich et al. (1989), to define workload values depending on task complexity (Table 2).

**Table 2. Workload component scale derived from (Aldrich et al., 1989)**

*Cognitive*

Descriptor	Workload value
Simple association	1.0
Select alternative	1.2
Sign/signal recognition	3.7
Evaluate single aspect	4.6
Encoding/Decoding/Recall	5.3
Evaluate several aspects	6.8

*Visual*

Descriptor	Workload value
Detect simple signal	1.0
Discriminate (Sign)	3.7
Inspect/Check (numerical)	4.0
Read (text)	5.9
Scan/Search/Monitor	7.0

*Auditory*

Descriptor	Workload value
Non-vocal signal recognition	6.6
Vocal signal recognition	4.9

*Psychomotor*

Descriptor	Workload value
Push the button	2.2
Switch toggle	2.2
Continuous adjustive controller	2.6
Discrete adjustive controller	5.8

*Haptic*

Descriptor	Workload value
Detect simple signal	1.0

Situation awareness is modeled as a correspondence of the values in the driver’s memory to the actual state of the car and road conditions. They may not correspond to each other if the task, which is responsible for the driver’s memory update is delayed or interrupted due to the overload. This trivial model of situation awareness is far from the fidelity of modern comprehensive models such as COSMODRIVE but sufficient for our purposes.

The HMI’s functional architecture optimization problem is formulated as follows:

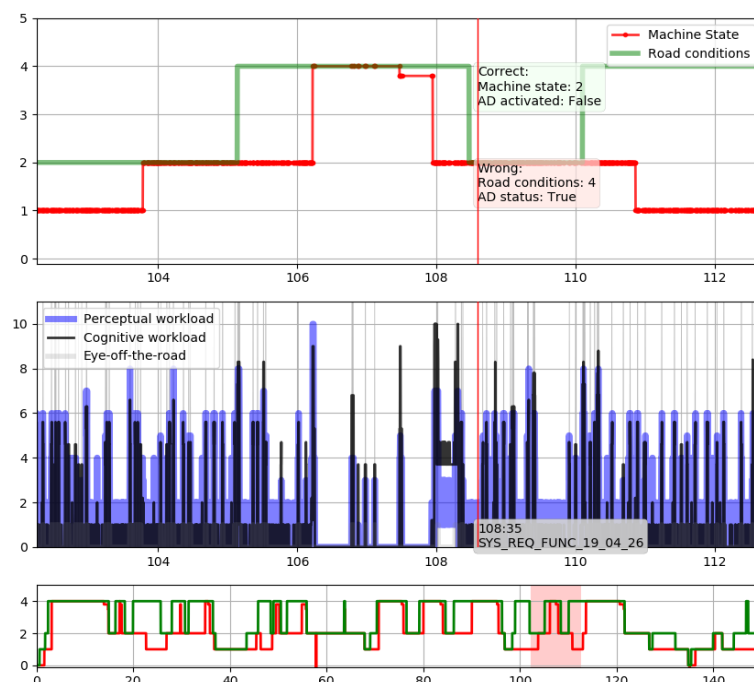
*Minimize the number of perceptual and cognitive overloads and the eyes-off-the road duration in a given interval of time, by optimizing the list of tasks and changing task-component allocation, wherein the situation awareness shall not go below a certain limit.*



When the parameters of the model have been set up, the model can be simulated in the discrete-event environment. Each run of the simulation is driven by several pseudo-random processes, namely by changing road conditions and by several cognitive functions maintained by human or machine (for example, “check automation level about every 20 seconds”). Though we are using pseudo-random sequences for the approximation of real processes, an alternative approach would be to use deterministic time series. For example, we can feed our model with records of some real road.

## 4. Results

The sample of the discrete-event simulation output is shown on Figure 3. The top chart shows the road conditions and machine state, the bottom – the full span of the simulation sample (the highlighted area marks the zoomed fragment). The chart in the middle represents cognitive and perceptual workloads of the executed tasks. The pointer between 108 and 110 seconds highlights the current task (19\_04\_26) and shows that at the moment the user has a wrong understanding of the road conditions state and autonomous driving mode availability.



**Figure 3. Timescale output of the discrete-event simulation**

In the following, we will compare two configurations of the functional architecture – the first is proposed by the company's internal experts in ergonomics (referred further to as base design option) and another one is a manually optimized version of the base design option. The latter means that we iteratively altered the list of tasks (functional architecture) to achieve improved overall characteristics – eyes-off-the-road time, perceptual and mental workload, and situation awareness. These values (i.e. the values of the multi-objective function) were calculated with our proposed tool.

The examples of the design decisions recommended for the functional architecture optimization are given in Table 3.

**Table 3. Examples of design decisions**

Change textual message “AD mode is available” on icon “AD” and reallocate it to Head-up display
Change textual message “L1 is activated” on icon “L1” on the Instrument cluster
Remove vocal message “Push on button to activate AD mode ”
Put vocal message “Drive Now” after haptic signal “TOR10” (avoid these signals to appear simultaneously)
Put non-vocal message “TOR10” after vocal message “Drive Now” (avoid these signals to appear simultaneously)

The discrete-event simulation environment uses pseudo-random generators to trigger events. Hence, to compare two functional architecture configurations we should execute a number of trials. The simulation time of each trial is 1000 minutes of virtual driving (computational time is around 30 seconds for each trial). Figures 4 and 5 show the statistics of the task execution in random trials for two architectures.

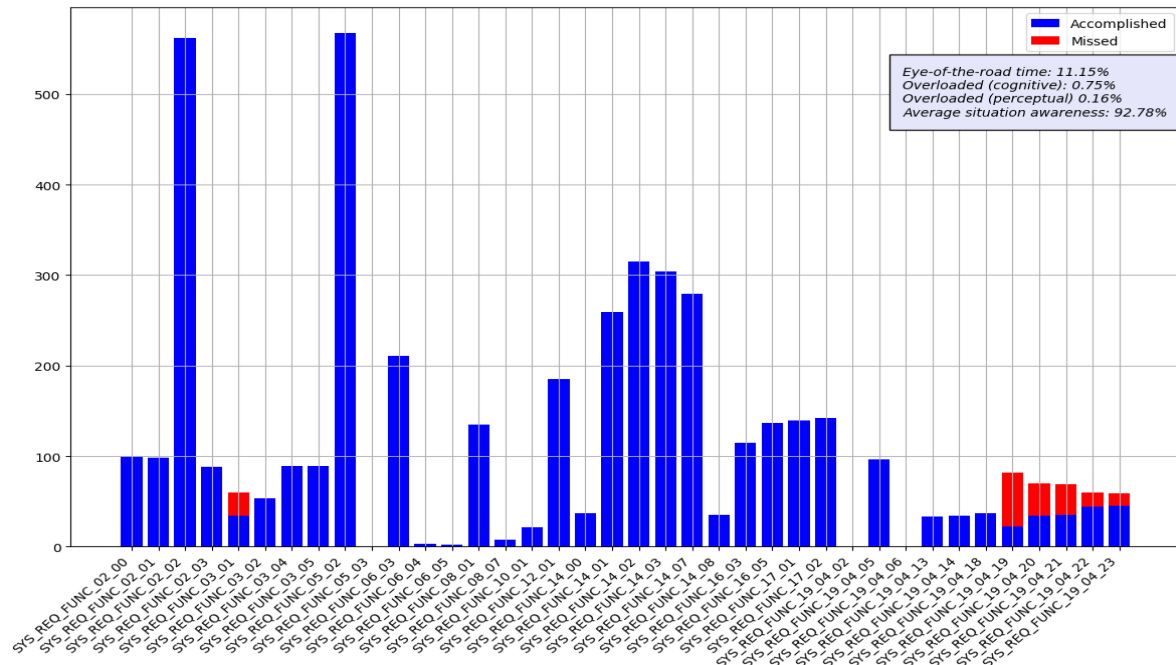


Figure 4. Base design option simulation statistics

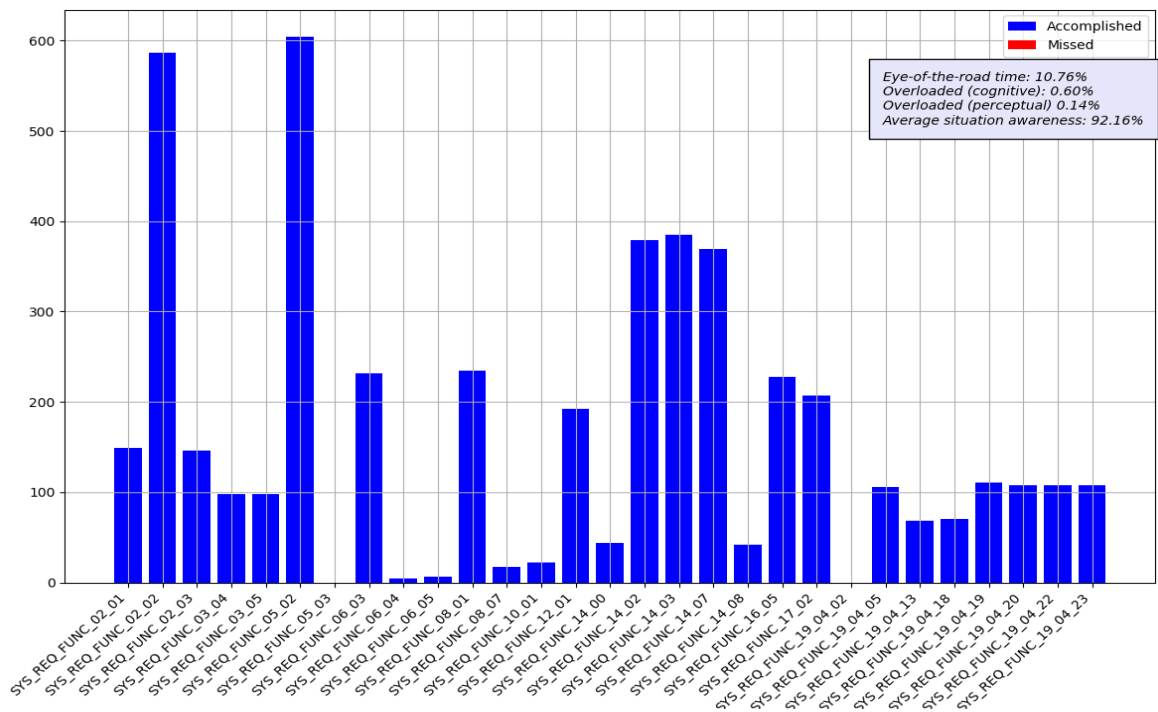


Figure 5. Optimized design option simulation statistics

Figure 6 shows the results of 40 simulations (20 for the base and 20 for the optimized configuration) with respect to eyes-off-the-road time and mental overload.



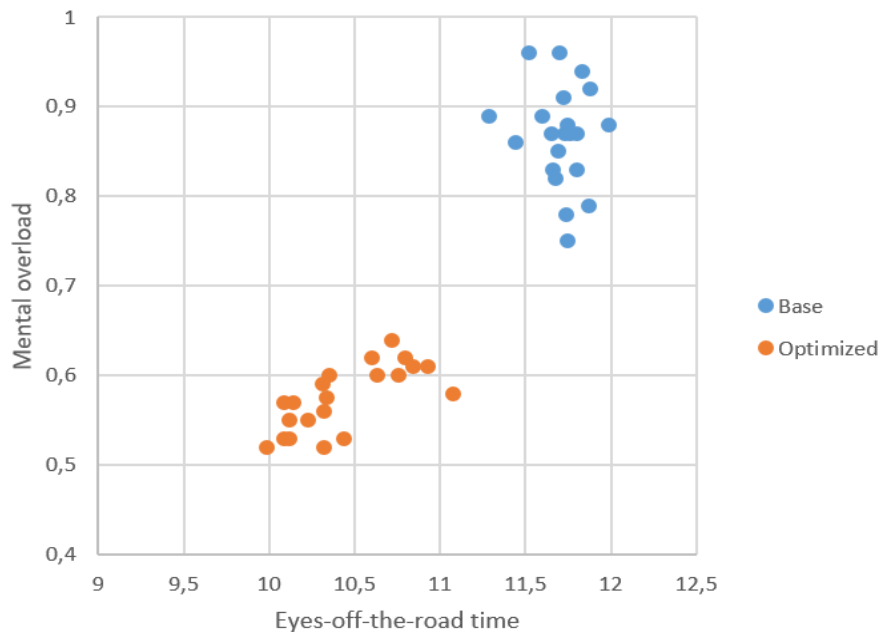


Figure 6. The results of discrete-event simulations of the base and optimized design option

Table 4. Averaged results of two design options

Indicator	Base	Optimized
Median eyes-off-the-road time, % of total time	11.7	10.3
Median cognitive overload, % of total time	0.87	0.57
Median perceptual overload, % of total time	0.19	0.13
Median situation awareness, average %	92.1	92.3

## 5. Conclusions and future work

In this paper, we have shown how existing task analysis approaches can be applied to the design of a modern vehicles with an adaptive level of automation. Our model enables a fast early-stage functional architecture optimization with respect to widely used safety metrics: cognitive and perceptual overload, eyes-off-the-road time and situation awareness.

We applied our model to a real industrial project and obtained a list of recommendations to improve the current conceptual design. Still, we can point out several directions for future work in order to improve the accuracy and validity of our model:

- Validation of the obtained workload and eyes-off-the-road results on the simulator with human subjects;
- Integration of a more sophisticated and accurate conflict matrix (measure of a conflict between pair of perceptual resources);
- Verification of the models of the human driver's cognitive functions (e.g. how often the driver checks the state of the machine?);
- Functional architecture optimization subject to different human profiles (experienced/novice driver, open-minded/conservative);
- Integration of models of human performance degradation (fatigue and drowsiness) and improvement (learning-curve) with time.

The major limitation of the model is the oversimplification of the cognitive processes of the human brain. In our setting, if the driver has glanced at some instrument, the perception of the indicated values on the instrument is without losses (under the condition of sufficient cognitive and perceptive resources in that moment). The situations of misinterpretation, low attention, effects of short-term memory and decision-making are not considered.

## References

- Aldrich, T.B., Szabo, S.M. and Bierbaum, C.R. (1989), "The Development and Application of Models to Predict Operator Workload During System Design", In: McMillan, G.R., Beevis, D., Salas, E., Strub, M.H., Sutton, R. and Van Breda, L. (Eds.), *Applications of Human Performance Models to System Design*, Springer US, Boston, MA, pp. 65-80. [https://doi.org/10.1007/978-1-4757-9244-7\\_5](https://doi.org/10.1007/978-1-4757-9244-7_5)
- Bellet, T. et al. (2011), "Human Driver Modelling and Simulation into a Virtual Road Environment", In: Cacciabue, P.C., Hjalmdahl, M., Luedtke, A. and Riccioli, C. (Eds.), *Human Modelling in Assisted Transportation*, Springer Milan, Milano, pp. 251-262. [https://doi.org/10.1007/978-88-470-1821-1\\_27](https://doi.org/10.1007/978-88-470-1821-1_27)
- Boy, G.A. (1998), "Cognitive function analysis for human-centered automation of safety-critical systems", *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems – CHI '98, presented at the the SIGCHI conference*, ACM Press, Los Angeles, California, United States, pp. 265-272. <https://doi.org/10.1145/274644.274682>
- Feuerstack, S., Lüdtkke, A. and Osterloh, J.-P. (2007), "A Tool for Easing the Cognitive Analysis of Design Prototypes of Aircraft Cockpit Instruments: The Human Efficiency Evaluator", *Proceedings of the European Conference on Cognitive Ergonomics 2015*, ACM, New York, NY, USA, pp. 22:1-22:8. <http://doi.org/10.1145/2788412.2788434>
- Flemisch, F.O. et al. (2014), "Towards cooperative guidance and control of highly automated vehicles: H-Mode and Conduct-by-Wire", *Ergonomics*, Vol. 57 No. 3, pp. 343-360. <https://doi.org/10.1080/00140139.2013.869355>
- Green, P.A. (2017), "Motor Vehicle Driver Interfaces", In: *The Wiley Handbook of Human Computer Interaction*, John Wiley & Sons, Ltd, pp. 843-888. <https://doi.org/10.1002/9781118976005.ch40>
- Kontogiannis, T. (2005), "Integration of task networks and cognitive user models using coloured Petri nets and its application to job design for safety and productivity", *Cognition, Technology & Work*, Vol. 7 No. 4, pp. 241-261. <https://doi.org/10.1007/s10111-005-0010-z>
- Korosec, K. (2019). "Waymo to customers: 'Completely driverless Waymo cars are on the way'", *TechCrunch*, 10 October, Available at: <http://social.techcrunch.com/2019/10/09/waymo-to-customers-completely-driverless-waymo-cars-are-on-the-way/> (accessed 15.11.2019).
- Laughery, R. et al. (2000), "Task Network Modeling and the Micro Saint Family of Tools", *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 44 No. 6, pp. 721-724. <https://doi.org/10.1177/154193120004400649>
- North, R.A. and Riley, V.A. (1989), "W/INDEX: A Predictive Model of Operator Workload", In: McMillan, G.R., Beevis, D., Salas, E., Strub, M.H., Sutton, R. and Van Breda, L. (Eds.), *Applications of Human Performance Models to System Design*, Springer US, Boston, MA, pp. 81-89. [https://doi.org/10.1007/978-1-4757-9244-7\\_6](https://doi.org/10.1007/978-1-4757-9244-7_6)
- Parasuraman, R. and Wickens, C.D. (2008), "Humans: Still Vital After All These Years of Automation", *Human Factors*, Vol. 50 No. 3, pp. 511-520. [https://doi.org/10.1007/978-88-470-1821-1\\_27](https://doi.org/10.1007/978-88-470-1821-1_27)
- Pettitt, M., Burnett, G.E. and Stevens, A. (2007). "An extended keystroke level model (KLM) for predicting the visual demand of in-vehicle information systems", *CHI*, available at: <https://doi.org/10.1145/1240624.1240852>
- Scerbo, M. (2008), "Adaptive automation", In: Parasuraman, R. and Rizzo, M. (Eds.), *Neuroergonomics: The Brain at Work*, Vol. 3, Oxford University Press, p. 239.
- Wang, J.-S., Knipling, R.R. and Goodman, M.J. (1996), "The role of driver inattention in crashes: New statistics from the 1995 Crashworthiness Data System", *40th Annual Proceedings of the Association for the Advancement of Automotive Medicine*, Vol. 377, p. 392.
- Wickens, C.D. (2002), "Multiple resources and performance prediction", *Theoretical Issues in Ergonomics Science*, Vol. 3 No. 2, pp. 159-177. <https://doi.org/10.1080/14639220210123806>
- Wittmann, M. et al. (2006), "Effects of display position of a visual in-vehicle task on simulated driving", *Applied Ergonomics*, Vol. 37 No. 2, pp. 187-199. <https://doi.org/10.1016/j.apergo.2005.06.002>
- Wortelen, B., Unni, A., Rieger, J.W. and Lüdtkke, A. (2016), "Towards the integration and evaluation of online workload measures in a cognitive architecture", *2016 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, pp. 11-16. <https://doi.org/10.1109/CogInfoCom.2016.7804516>