Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 91 (2020) 407–414

www.elsevier.com/locate/procedia

30th CIRP Design 2020 (CIRP Design 2020)

A Modular Research Platform – Proof-of-Concept of a Flexible Experiment Setup Developed for Rapid Testing of Simulators, UIs and Human Physiology Sensors

Leif Arne Hatlem[a], John Chen[a], Henrikke Dybvik[a]*, Martin Steinert[a]

[a]The Norwegian University of Science and Technology (NTNU), Richard Birkelands vei 2B, 7491 Trondheim, Norway

* Corresponding author. Tel.: +47 95 41 72 45. *E-mail address:* henrikke.dybvik@ntnu.no

---

Abstract

This work presents a modular research platform to design, test and run human-machine interaction (HMI) experiments. Traditionally, HMI experiments are time and resource consuming, particularly in the piloting phase. Furthermore, such experiment setups are often rigid and only fit to one particular hypothesis. Thus, significant time is needed to alter the setup to new hypotheses, if this is possible at all. The platform presented is a technical proof-of-concept of a highly flexible experiment setup, which can rapidly be adapted to alternative hypotheses. Examples of interchangeable modules include simulator software (context), user interface (independent variable) and human operator physiology sensors (dependent variable). An agile product development methodology, Wayfairing, was used to accomplish this.

Keywords: human-machine interaction, experiments, modularity, research platform, flexibility, UI, physiology sensors

---

1. Introduction

This paper presents development of a modular research platform intended for designing, testing and running human-machine interaction (HMI) experiments. The platform presented is a technical proof-of-concept of a highly flexible experiment setup. Examples of interchangeable modules include simulator software (context), user interface (independent variable) and human operator physiology sensors (dependent variable). The aim of this prototype is to be research-ready for HMI experiments, which means the researcher can rapidly investigate different hypotheses in a piloting phase [1] by changing abovementioned modules, prior to a full experimental run testing one or more selected hypotheses statistically.

Experiments and interaction studies are equipment intensive, time-consuming and labor intensive [2]. Availability of research platform, high cost and proprietary costs are additional challenges faced by academic researchers [3]. Furthermore, HMI experiments rarely run correct the first time [4], and although preparing all experimental parameters well in advance is ideal, this is not the case in practice. For novel interaction techniques in early stages of product development constructing a well-defined hypotheses if often very hard [5]. For example, multiple years were required to develop a humanoid research platform investigating human interaction [3]. Experimental setups like the car simulation of Ahn et al. [6] and autonomous car simulation of Gil et al. [7] are likely to be quicker to set up and pilot. However, they both feature a simple monitor, steering wheel and chair, thus lacking ecological validity. The lack of ecological validity is problematic as experimental findings are not generalizable to real-life settings.

HMI experiments such as the abovementioned investigate the human, how they interact, how they operate, how to and in what way include to the human user in the loop, etc. In such

10.1016/j.procir.2020.02.193

experiment setups there is an increasing interest in measuring the human's physiological response as a dependent variable. Examples of using physiology sensors in HMI research include assessing drivers' cognitive engagement under varying levels of automation in a driving simulator using fNIRS [8], investigating the effect of mental fatigue caused by sleep deprivation in driving using simultaneous fNIRS, EEG, and ECG [6], and investigating operators' mental state in ship navigation using ECG and GSR [9], [10]. When including physiology sensors in HMI research additional challenges, such as hardware integration, arise [11].

The fundamental nature of a classical experiment is to vary one or more independent variables, observing potential changes in one or more dependent variables [12]. HMI experiments has since the beginning embraced cognitive science and used psychology style experiments as a basis for usability testing [5]. Different designs or solutions are compared through controlled studies, often including baseline and comparisons tests in a within-subjects design (in which all subjects test all designs), measuring a range of variables [13]. Afterwards, testing for statistical differences in these variables are custom as there are no absolute values to compare with. As mentioned, the piloting phase is important in HMI since it increase the chance of experiment success [1], although it produces additional demands in terms of time, labor and cost simultaneously. Thus, time to develop an experiment and hypotheses can be greatly reduced by the possibility to quickly test different independent and dependent variables.

With these notions in mind the authors wanted to develop a modular research platform allowing for rapid changes of context, independent and dependent variables during piloting. The core feature would be the ability to easily and rapidly test different interface configurations, simulators and physiology sensors. The experiment setup should be flexible and produce an experiment with high ecological validity, reliability and reproducibility.

Thus, this paper presents a modular research platform. It describes design and development of the flexible experiment platform and demonstrate it by a proof-of concept experimental user test. This paper focuses on the physical aspect of the modular research platform, notably providing a prototype that is technically robust. A dedicated software was also developed as a part of the system, which is presented in brief, but an exhaustive description is not within the scope of the paper.

The paper is structured as follows: section 2 describes the method. Section 3 briefly describes the development process and proof-of-concept user test. Section 4 presents the modular research platform and describes how it can be adapted to new hypotheses. Section 5 is discussion and section 6 conclusion.

2. Method

Wayfaring is a flexible, physical product development methodology for early stage product development, which utilizes principles from agile software development [14], [15], [16]. Problem understanding is developed to such a degree that good concept choices and appropriate requirements can be made in an early phase thus preventing costly loop-backs later in the process. A special focus is placed on interactions between disciplines, leveraging diversity in teams to promote iterative learning cycles through rapid conceptual prototyping. Exploration is conducted through a journey of idea-probes, each probe consisting of designing, building and testing prototypes, aiming towards a vision of a problem solution. Critical functions of components should be tested in isolation. System integration occurs when these functions are fulfilled, to validate the continued fulfilment in the system context [14]. Integrating different disciplines can reveal inter-dependencies among disciplines, thus design changes in one domain can cause requirement adaptation in another domain. Unknown unknowns are uncovered early, while flexibility is high and cost of change low.

Wayfaring has been introduced as a development tool for human experiments in interaction design and engineering design science [17]. It is applicable in the early and ambiguous conceptualization and design of experiments, as well as cases where no obvious experiment precedes it and it must be built from scratch. Four main principles are particularly advocated: probing ideas, merging multidisciplinary, and retaining high speed and agility.

Prototypes are often used when developing products in engineering design and are important in fuzzy-front-end projects where wayfaring has been utilized, especially when developing products with a physical dimension [16], [18] [14], [17]. Prototypes are purposefully formed manifestations of design ideas built to traverse a design space. Such prototyping activity can create valuable knowledge of the final design [18]. In wayfaring, each prototype is built to test a specific idea and/or a system interaction [16].

Affective Engineering uses physiology sensors to capture and incorporate the human emotional dimension as a part of evaluating and identifying the better design of an interface, a process or product [19], [20]. Examples of physiology sensors are ECG, GSR, fMRI, fNIRS, EEG, PPG, EMG, pupil tracing devices, etc., [20].

3. Development

This section highlights certain aspects of the development process. It describes development of the physical infrastructure and user interfaces (independent variables), before pointing to rapid and relatively large changes between pilots. A description of a proof-of-concept experimental run follows.

3.1. Software development

A software, TrollSim, was developed using agile methodologies. Software requirements were mostly driven by needs and integration testing from the physical domain.

3.2. Developing experiment infrastructure

A small room was constructed using a timber-frame and MDF-sheets. This provided a stiff frame which internal structures could be attached to. Floor, walls and roof were easy to adapt and continue building upon. Flexibility for further development was continuously considered [16]. The internal infrastructure was designed and built to hold a classical experiment setup, aiming at a static, controlled physical environment. The simulator room was divided in two by a black curtain, with the majority dedicated to the participant. LED-lights were installed in the ceiling, as well as a curtain, limiting visual disturbances from the experimenter area and outside the box.

User interfaces was explored by developing Arduino based alarm systems and operator controllers. Initial development isolated the two systems until critical functions were stable, before they were integrated in one device. The alarm system first employed the modalities sound and smell, the first intended as a baseline and the second as a high novelty alternative. The sound alarm utilized a simple buzzer to play monotones and a 3D printed button to register reaction time. The container was laser-cut. The smell alarm utilized a servo motor to rotate a disk holding three different laser-cut cartridges containing variations of air fresheners (Little Trees/Wunderbaum), and a fan directing scent to the participant’s nose.

A two-piece joystick served as a baseline controller, which we sought to test along with a high-novelty alternative. This drove the development of a sensor glove, an Arduino-based controller with a finger-actuated flex sensor controlling throttle and a gyroscope allowing hand movement to control pitch, roll and yaw.

3.3. Experiment piloting

The final experiment procedure (Fig. 4) was refined through piloting to a high level of detail. The piloting phase can be characterized by relatively large changes and rapid learning between pilots. This work had a development time of two months and was conducted by two master students.

The experiment was piloted eight times. From pilot 1 to 2 questionnaires were created and implemented in the procedure. From pilot 2 to 3, the setup was changed from three factors to two factors by removing controllers as an independent variable. The alarm system was redesigned, changing from sound and smell modalities to combinations of sound, light, and haptic feedback modality. A physical alarm control panel was added. Between 3 and 4 the method of achieving low and high workload intensity was redesigned from using different pre-programmed flight school tutorials in X-Plane 11 [21], to the same one coupled with secondary tasks. This was controlled by a newly created audio control panel. The alarm system parameters were also tweaked to better approximate reality. Between 5 and 6, all task training was redesigned from manual instruction to automated and a second screen was added to the participant infrastructure. This marked a point where the experiment ran fairly smooth, except for minor bugs in different modules. Between 7 and 8, preliminary data analysis of questionnaire answers was implemented in Google Sheets.

3.4. Proof-of-concept user test

This section describes the proof-of concept experimental run conducted after piloting, using the proposed setup as described in section 4. An external researcher conducted the experiment. They received instructions, which included a walkthrough of the procedure, a procedure checklist and a manuscript. The researcher had some experience with running HMI-experiments with physiology sensors, having previously completed one study. This researcher conducted one pilot with one of the developers (pilot nr 5), and two user tests after pilot 8. There were minor issues with the first test, whereas the second ran had no issues and marked the proof-of-concept experimental run. In total 10 runs ensured technical robustness and repeatability.

4. Proposed system – A descriptive of the modular research platform

This section describes a proposed experiment setup. This is the setup used during the proof-of-concept user test. Then, we give examples of changes that are easy to implement in the modular research platform to build a new experiment and investigate new hypotheses.

4.1. Software

A dedicated software, TrollSim, was developed. In general, TrollSim can collect data from a multitude of data sources, manipulate the data, log it and transmit it. TrollSim can gather, synchronize and log physiology data with data from scenario events, controller input and the simulator. A fronted allows the researcher to influence data through the UI, and a live plot with logdata visualization allows for a quick overview over a finished experiment. The software is intended to function as an API for its users, designed with the aim of exposing the user to minimal amount of dataflow logic, focusing on actual business logic. Further description of TrollSim is not within the scope of this paper.

4.2. Physical infrastructure

4.2.1. Proposed physical infrastructure

The following setup is designed around one experimenter, but can also be used by two experimenters. The physical infrastructure is shown in Fig. 1. It consists of (1) monitors , (2) laptop controlling TrollSim, (3) keyboard and mouse for main computer, (4) control panel for alarm device, (5) audio control panel for triggering secondary tasks, (6) experimenter check list, and (7) boxes for forms used in experiment (prepared for use are on the right, used and collected are on the left).

Fig. 1. Infrastructure for the experimenter

Fig. 2. Close-up of the experimenter screen setup

Fig. 2. shows a close-up of the experimenters' screen setup, which contains (1) the camera feeds capturing the participants' face and desk, (2) a video feed of the participants main screen, (3) live plots of physiology sensors, (4) pyCharm, showing which audio tracks are triggered and played for the participant and (5) participant screen, used to monitor flying tasks as well as initiate X-Plane scenarios.

The participant is separated from the experimenter by a curtain. They enter thought the rearmost part, mimicking entering a cockpit. Participant interface is shown in Fig. 3. and consists of (1) 2 web cameras capturing participant face and desk, (2) 2 screens, instrument zoom (left), main screen (right) (3) ear protection w/auxiliary input from PC for playback of audio alarms and secondary task instructions, (4) joystick: throttle (left), main stick (right), (5) keyboard (FN-buttons and numpad restricted), (6) pink note with participant number and experiment group, (7) alarm device, (8) secondary task sheet, (9) mouse, and (10) printout with names of cockpit instruments.

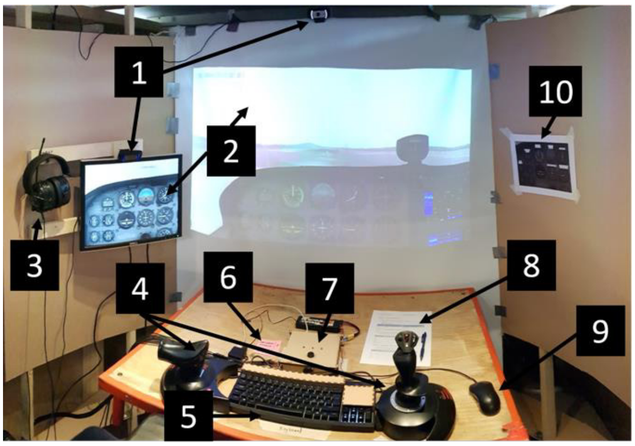

Fig. 3. Participant interface

4.2.2. Changing the physical infrastructure

The wooden structure allows for fastening objects anywhere in the experiment room. All physical objects can be moved to accommodate re-configurations. The main screen, a single, short-throw projector can for example be swapped for one or more monitors, different backdrop and projector. It could be interesting to use different monitor types as an independent variable, observing what effect different configurations impose on the participants. Arduino based devices or other devices can be swapped or added by simply connecting USB cables.

4.3. Procedure

4.3.1. Proposed procedure

Initial instructions were given by the experimenter after having greeted the participant, collecting consent form and attaching physiology sensors. Afterwards, instructions were automated to minimize variation. Exceptions were briefing and debriefing. Automatic instructions were made by combining iMotions [22] and Google Slides. iMotions also read physiology measurements and recorded video and audio.

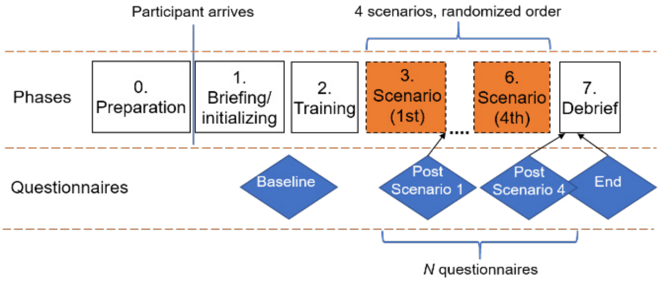

Fig. 4. Experiment procedure.

Primary task training was conducted through pre-programmed flight school tutorials in X-Plane 11 [21]. A Google Slides presentation provided secondary task training, allowing the participant to navigate back and forth at their own pace.

The primary task was operating a Cessna 172 airplane, the participants were tasked with completing a "Traffic Pattern" tutorial. Four different scenarios were created by combining two independent variables, alarm system and workload intensity:

- Traffic pattern. High-workload. Alarm system A.
- Traffic pattern. Low-workload. Alarm system B.
- Traffic pattern. High-workload. Alarm system B.
- Traffic pattern. Low-workload. Alarm system A.

To avoid learning effects, the order was randomized.

4.3.2. Changing procedure

The experiment procedure can be adapted to new setups as described above by altering the number and order of scenarios. Experiment instructions are easy to modify. The combination of iMotions with Google Slides in training allowed the participant to navigate selected instructions at their own pace, while the experimenter retains overall control.

4.4. Independent variables

4.4.1. Proposed independent variables

A two factor, two-level experiment design was achieved with the two independent variables: alarm system and workload intensity. Each has two levels: alarm system A and B, and high and low workload.

The alarm device triggers two different alarm systems with different combinations of the following modalities; visual (blinking LEDs), auditory (alarming sounds) and tactile (haptic force feedback). Alarm system A consist of light and sound. Alarm system B consists of light, sound and haptic vibration in the right-hand controller. Three levels of danger (high, moderate, low) approximated airplanes centralized warning system to achieve ecological validity. Alarms are triggered at random points in time, a pre-defined number of times. The participant is tasked with pressing a button when they register an alarm going off. Both the alarm going off and the button being pressed is logged, measuring response time.

To alter the perceived workload intensity under different flight scenarios, secondary tasks were implemented in the high workload scenario. These tasks were pre-recorded questions, triggered by the experimenter using an audio control panel at pre-defined times during the scenario. The participant would answer the question using information from X-Plane in writing on a dedicated secondary task sheet. To ensure ecologically validity the questions were developed in collaboration with a pilot. Three tasks resulted: estimation of remaining flight time based on remaining fuel, transponder code changing, and reading and logging of time and heading. The low workload scenario had no secondary tasks.

4.4.2. Changing independent variables

Interchangeability of peripheral programs such as X-Plane 11 and iMotions is a feature as they merely are examples of what TrollSim can communicate with. Switching X-Plane for a different simulator software, for example Ship Simulator Extremes [23] is relatively simple. One aspect to keep in mind, X-Plane provides an API for developers to communicate with and send commands through, which allows for e.g. reading and overwriting internal states. If the alternative software doesn't have an API these features are lost. Ship Simulator Extremes does not have an API, therefore reading or writing internal states is not possible, neither is sending commands. If data for controllers are still an important aspect, there are two possibilities here. Ship Simulator accepts input from keyboard, which we can emulate in two ways. The first is with physical microcontrollers such as Arduino. The second alternative is to create software running on the host side sending keyboard signals to the host OS. This shows that we can still create custom controllers, similar to the sensor glove prototype, where we get full access to data emitted. Full access to controller data means that we can log it as we would with X-Plane.

Sub-variables of the current alarm system are easy to change, e.g. the number of alarms, volume, alarm sound or haptic intensity.

A custom Arduino-based controller was developed (the sensor glove), but not included in the final experiment due to reducing from 3 to 2 independent variables. However, this means it is trivial to add custom controllers, as it is implemented in the same manner as the alarm device. Arduino devices are logged in TrollSim and the same protocols and dataref-control can be used for any Arduino-based controller. Possibilities for Aruduino based devices are only limited by available sensors and imagination.

Scenario nature can be altered by reducing or increasing the number, selecting different pre-programmed tutorials or flights, or custom-make flights in the software. Using pre-programmed tutorials in itself offers very little flexibility, aside from selecting an appropriate tutorial. Building highly customized scenarios in X-Plane is an option, which requires more time, but incudes many options and more freedom. Options include triggering hundreds of failure modes such as engine failure, rudder failure, fuel leakage etc., creating custom weather and plan routes for the participant to fly.

Scenario intensity can be changed by altering workload. Workload can be increased by triggering secondary tasks at more stress-full points in time, such as during a turn or landing. The three tasks could be made harder by increasing the arithmetic difficulty, creating a larger table to look up values in, and requiring the participant to log additional instrument readings. Furthermore, introducing more stimuli, such as white noise in headphones, makes it harder to interpret sounds and voice commands. Removing the second screen requires the participant to more actively zoom in/out to read instruments. For decreased scenario intensity analogue adaptations to the stimuli can be made. Generally, cognitive workload can be adjusted by adjusting primary and secondary tasks in combination with other stimuli.

4.5. Dependent variables

4.5.1. Proposed dependent variables

Two types of physiology data were collected, electrocardiography (ECG) and galvanic skin response (GSR). ECG measures the electrical potential difference across the heart, which can be translated to heart rate and heart rate variability (HRV) among other measures. Physical and mental states can be interpreted from these measures. The Shimmer3 ECG unit [24] was used with five electrodes and a sampling rate of 512 Hz. Galvanic skin response (GSR) measures skin conductance, which increase with physical activity and/or emotional arousal/alertness [20]. To detect GSR changes due to emotional stimuli the experiment must be conducted under very controlled physical conditions [25], which the setup exhibits. The Shimmer3 GSR+ Unit [26] was used with two electrodes and 128 Hz sampling rate.

Performance is measured by alarm response time, answers to secondary tasks, and flight performance data which is generated by X-Plane 11's 'report card'.

Subjective measures were collected by questionnaires in Google Forms. They included The Affect Grid [27], level of stress, NASA-RTLX [28], Overall Workload [29] and scenario specific questions, such as alarm preference.

4.5.2. Changing dependent variables

Alternative physiological sensors are possible through use of code wrappers and TrollSim. One example are open source physiology sensors in Arduino.

The current alarm system has one pushbutton recording the participants response time. To measure e.g. alarm recognition instead of reaction time is simple. Three additional buttons corresponding to the three different alarms can be added. This requires rewiring a few electronic circuits, slightly altering an Arduino code by adding variables, 3D printing extra buttons and a simple laser cut (this was done in an earlier prototype). As explained, any Arduino based sensor or device integrates with the setup.

Constructs other than stress, affective state and workload, can be teste by changing the questionnaires accordingly.

4.6. Example hypotheses

The proposed setup enable investigation of e.g. the following hypothesis:

- Reaction time decrease with alarm system B, both during high and low workload scenario.

By changing one or more of the independent variables, dependent variables or experiment procedure a new hypothesis can be investigated. One example of an alternative setup is as follows. A custom flight is made in while several accidents or near accidents occur, which trigger several different types of failure modes. The operator's action in response to the failure can be categorized as correct, intermediate and wrong reaction (based on domain knowledge). Physiology sensors are changed and EEG/fNIRS are used to measure e.g. cognitive workload. The alarm systems stay the same. This new setup enables investigation of e.g. the following hypotheses:

- Alarm system B significantly increase the number of correct failure reactions, and
- significantly decrease reaction time, and
- the operator's cognitive workload significantly decreases.

Here, the time to alter the setup to new hypotheses is significantly less than in traditional HMI experiments setups. Due to our platforms focus onto flexibility, researchers can thus rapidly iterate and test example alternative senarios, interface configurations or sensors which in turn enables them to rapidly investigate multiple new hypotheses.

5. Discussion

5.1. System development

Free software was prioritized to minimize system cost and to reduce entry barriers to testing the system. Using python and Arduino, both open source gave maximum flexibility. iMotions [22], a paid closed source software was used for experiment execution and initial sensor connection due to not finding an equivalent open source software. G-Suite products (Forms, Sheets, Slides) was used in combination with iMotions, which countered lacking flexibility in iMotions and provided an integral experiment execution. This illustrates that many issues can be solved without specifically written software, but instead hacking together existing solutions, provided integration does not lead to a greater total challenge. By basing many of the modules where flexibility was particularly important on inherently flexible, open source products and platforms we argue our system has inherited much of the same flexibility.

Throughout the development process, the system architecture considered TrollSim as a central hub, which was built to effectively handle data from a broad range of sources. Modularity was achieved by creating a standard code wrapper. This allowed development of custom modules without the need to extensively consider how they would affect overall system architecture and how it should be integrated. The Arduino based alarm device is one example of such a custom module. It was prototyped and tested on a module level first, until it functioned as intended by itself. The code wrapper was then added, identification defined and integrated in the code. We believe this strategy allowed for more testing since each prototype was built to test only one specific idea and/or system interaction, removing anything superfluous, which is consistent with Leifer and Steinert [16]. This simplified module-code made both debugging and modifications of code easier at the early stages of development. Furthermore, this modularity allowed a non-developer (mechanical engineer) to develop custom modules with relatively little code, which was wrapped and then functioned with the overall system. Through all pilot studies and two subsequent user tests, we have had no crashes or lost logs in TrollSim, which we argue is a showcase of its robustness.

The strategy of adding on modules transferred to the physical domain. The alarm control panel and the audio control panel are both examples of parts of the experiment system added when the need emerged, without requiring any noteworthy environment adaptation for system integration.

Throughout the development process, high flexibility was the main consideration driving development choices. This is illustrated by the choice of a modular system structure, versatile logging functions in TrollSim, and Arduino. Arduino is an opensource hardware platform for mechatronics development. Having developed and integrated one such device made all subsequent integration of Arduino based devices trivial. The choice to not solder any connections when prototyping mechatronics greatly sped up and simplified all iterations of these subsystems. Except for dynamically stressed connections, only breadboard or screw connectors were used.

Flexibility during development was further aided by how the physical infrastructure was constructed. The timber and MDF construction created a room in which all walls, floors and the roof could be modified and built upon. This gave great design freedom. A wall-to-wall scrum board inside the room and strategically placed storage for equipment contributed to adding many minor features, which overall contributed substantially to the end result. There were no penalties and low barriers to prototyping different setups and configurations, consistent with findings on key factors of spaces that facilitate change [16].

5.2. System cost

The system is developed based on the lowest resolution possible (but not lower) in every area, keeping overall cost low. Overall costs can be split in two. A one-time cost of 4200 € for developing the system, and a yearly subscription cost of 2077 € for iMotions. Approximately 50% of the total one-time costs are general hardware a university or research institution might already have available, such as main computer, monitors, keyboards, mouse, cables etc. That aside the main costs are the projector, the Shimmer sensors and iMotions software. These were available and chosen over cheaper alternatives to secure medical-grade equipment and reliable data capture. If requirements were different and iMotions was not required it would be possible to replace the projector with a monitor at < 50% of the cost and exchange the Shimmer sensors for Arduino ECG and GSR sensors priced below 50 €. However, we assume the quality of these are different. The sum of these changes would reduce the total one-time cost of the setup by approximately 1/3 and remove the subscription cost.

It should be noted that iMotions has an initial cost, dependent on your setup and their proposal.

5.3. Limitations

The flexible experiment setup is a proof-of-concept of technical robustness. A full experimental run is out of the scope of this paper. Pilot experiments were conducted with student participants. The current TrollSim is a proof-of-concept and has not been tested extensively.

6. Conclusion

This paper presents a modular research platform to design, test and run HMI experiments. Traditional HMI experiments are time and resource consuming, often rigid and only fit to one particular hypotheses. Thus, significant time is needed to alter the setup to new hypotheses, if this is possible at all. The platform presented is a proof-of-concept of a highly flexible experiment setup, which can rapidly be adapted to alternative hypotheses by simply changing one or more independent and dependent variables. Examples of interchangeable modules include simulator software (context), user interface and human operator physiology sensors. The time to alter this setup to new hypotheses is significantly less than traditional HMI experiments. Wayfairing, an agile product development methodology, was used to accomplish this.

Acknowledgements

This research is supported by strategic funds from the department for Mechanical and Industrial Engineering (MTP) at NTNU.

References

[1] E. R. Van Teijlingen and V. Hundley, "The importance of pilot studies," 2001.

[2] K. Dautenhahn, "Methodology & Themes of Human-Robot Interaction: A Growing Research Field," *International Journal of Advanced Robotic Systems*, vol. 4, no. 1, p. 15, Mar. 2007, doi: 10.5772/5702.

[3] K. Nishiwaki, J. Kuffner, S. Kagami, M. Inaba, and H. Inoue, "The Experimental Humanoid Robot H7: A Research Platform for Autonomous Behaviour," *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1850, pp. 79–107, 2007.

[4] H. C. Purchase, *Experimental human-computer interaction: a practical guide with visual examples*. Cambridge University Press, 2012.

[5] J. Lazar, J. H. Feng, and H. Hochheiser, *Research methods in human-computer interaction*. Morgan Kaufmann, 2017.

[6] S. Ahn, T. Nguyen, H. Jang, J. G. Kim, and S. C. Jun, "Exploring Neuro-Physiological Correlates of Drivers' Mental Fatigue Caused by Sleep Deprivation Using Simultaneous EEG, ECG, and fNIRS Data," *Front. Hum. Neurosci.*, vol. 10, 2016, doi: 10.3389/fnhum.2016.00219.

[7] M. Gil, M. Albert, J. Fons, and V. Pelechano, "Designing human-in-the-loop autonomous Cyber-Physical Systems," *International Journal of Human-Computer Studies*, vol. 130, pp. 21–39, Oct. 2019, doi: 10.1016/j.ijhcs.2019.04.006.

[8] S. Sibi, S. Balters, B. K. Mok, M. Steinert, and W. Ju, "Assessing driver cortical activity under varying levels of automation with functional near infrared spectroscopy," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1509–1516, doi: 10.1109/IVS.2017.7995923.

[9] A. S. Wulvik, H. Dybvik, and M. Steinert, "Investigating the relationship between mental state (workload and affect) and physiology in a control room setting (ship bridge simulator)," *Cogn Tech Work*, Apr. 2019, doi: 10.1007/s10111-019-00553-8.

[10] H. Dybvik, A. Wulvik, and M. Steinert, "STEERING A SHIP - INVESTIGATING AFFECTIVE STATE AND WORKLOAD IN SHIP SIMULATIONS," presented at the 15th International Design Conference, 2018, pp. 2003–2014, doi: https://doi.org/10.21278/idc.2018.0459.

[11] A. Sears and J. A. Jacko, *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*. CRC press, 2007.

[12] L. T. Blessing and A. Chakrabarti, *DRM, a design research methodology*. Springer Science & Business Media, 2009.

[13] J. A. Jacko, *The human-computer interaction handbook : fundamentals, evolving technologies, and emerging applications*, 3rd ed. Boca Raton, Fla: CRC Press, 2012.

[14] A. Gerstenberg, H. Sjöman, T. Reime, P. Abrahamsson, and M. Steinert, "A Simultaneous, Multidisciplinary Development and Design Journey – Reflections on Prototyping," in *Entertainment Computing - ICEC 2015*, 2015, pp. 409–416, doi: 10.1007/978-3-319-24589-8_33.

[15] M. Steinert and L. J. Leifer, "'Finding One's Way': Re-Discovering a Hunter-Gatherer Model based on Wayfaring," *International Journal of Engineering Education*, vol. 28, no. 2, p. 251, 2012.

[16] L. J. Leifer and M. Steinert, "Dancing with ambiguity: Causality behavior, design thinking, and triple-loop-learning," *Information Knowledge Systems Management*, vol. 10, no. 1–4, pp. 151–173, Jan. 2011, doi: 10.3233/IKS-2012-0191.

[17] K. K. Leikanger, S. Balters, M. Steinert, and others, "Introducing the wayfaring approach for the development of human experiments in interaction design and engineering design science," in *DS 84: Proceedings of the DESIGN 2016 14th International Design Conference*, 2016, pp. 1751–1762.

[18] Y.-K. Lim, E. Stolterman, and J. Tenenberg, "The anatomy of prototypes: Prototypes as filters, prototypes as manifestations of design ideas," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 15, no. 2, p. 7, 2008.

[19] S. Balters and M. Steinert, "Decision-making in engineering-a call for affective engineering dimensions in applied engineering design and design sciences," presented at the Innovative Design and Manufacturing (ICIDM), Proceedings of the 2014 International Conference on, 2014, pp. 11–15.

[20] S. Balters and M. Steinert, "Capturing emotion reactivity through physiology measurement as a foundation for affective engineering in engineering design science and engineering practices," *Journal of Intelligent Manufacturing*, pp. 1–23, 2015.

[21] *X-Plane 11 Flight Simulator*. 2017.

[22] "iMotions: Biometric Research, Simplified," *iMotions*, 12-Jul-2017. [Online]. Available: https://imotions.com/. [Accessed: 07-Dec-2017].

[23] *Ship Simulator Extremes*. 2010.

[24] *Shimmer3 ECG/EMG Unit*. Dublin, Ireland: Shimmersense, 2017.
[25] M. van Dooren, J. J. G. (Gert-J. de Vries, and J. H. Janssen, “Emotional sweating across the body: Comparing 16 different skin conductance measurement locations,” *Physiology & Behavior*, vol. 106, no. 2, pp. 298–304, May 2012, doi: 10.1016/j.physbeh.2012.01.020.
[26] *Shimmer3 GSR+ Unit*. Dublin, Ireland: Shimmersense, 2017.
[27] J. A. Russel, A. Weiss, and G. A. Mendelsohn, “Affect grid: A single-item scale of pleasure and arousal,” *Journal of Personality and Social Psychology*, vol. 57, no. 3, pp. 493–502, 1989.
[28] S. G. Hart, “NASA-task load index (NASA-TLX); 20 years later,” presented at the Proceedings of the human factors and ergonomics society annual meeting, 2006, vol. 50, pp. 904–908.
[29] M. A. Vidulich and P. S. Tsang, “Absolute Magnitude Estimation and Relative Judgement Approaches to Subjective Workload Assessment,” *Proceedings of the Human Factors Society Annual Meeting*, vol. 31, no. 9, pp. 1057–1061, Sep. 1987, doi: 10.1177/154193128703100930.