Automated Online Determination of Pilot Activity Under Uncertainty by Using Evidential Reasoning

Fabian Honecker(✉) and Axel Schulte

Institute of Flight Systems (IFS), Bundeswehr University Munich,
Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany
{fabian.honecker,axel.schulte}@unibw.de

Abstract. The objective of a workload-adaptive associate system is to support human pilots in critical workload situations to avoid excessive demands on their mental capacity. Since workload strongly depends on the current activity of the pilots, i.e. the tasks the pilots are performing in a specific task situation, a key feature of an adaptive associate system is to determine the activity of the pilots online in real-time.

This contribution presents a method for determining the activity of helicopter pilots automatically in an uncertain and complex environment like military manned-unmanned teaming missions (MUM-T), where multiple unmanned aerial vehicles (UAVs) are commanded from the cockpit of a manned helicopter. We use a pilot observation system with different measurement sensors to collect evidences and apply an evidential reasoning algorithm to draw conclusions on the actual activity of the pilots. Our method is based on a simplified version of Dempster-Shafer theory and is capable of collecting and combining even contradictory evidences.

By providing a means of implicit deliberative communication, this method lays a foundation for improving human-machine team performance in complex task situations. The implementation of this method in a helicopter mission simulator is explained in detail.

Keywords: Activity determination · Adaptive associate systems · Dempster-Shafer theory · Mental workload · Pilot monitoring

1 Introduction and Problem Description

The domain of military helicopter missions is characterized by complexity and uncertainty, especially if the pilots command several unmanned aerial vehicles (UAVs) from the cockpit of a manned helicopter (manned-unmanned teaming MUM-T) [1, 2]. During their mission, the pilots must perform many cognitive tasks that require different amounts of mental resources. While processing those tasks, the workload of the pilots can widely change with different task situations. Especially in situations, in which the pilots are overloaded, decrements in human performance might occur. These decrements in human performance can have a negative impact on the total performance of a helicopter mission.

To counteract performance decrements in high workload situations, automation is often used to reduce mental workload (MWL) of the operator of a technical system. But

D. Harris (Ed.): EPCE 2017, Part II, LNAI 10276, pp. 231–250, 2017.
DOI: 10.1007/978-3-319-58475-1_18

the use of automation is not carefree and human factors problems play the central role. Among them are deskilling of the operator [3], boredom, complacency effects and clumsiness of the automation [4, 5], mode errors and mode confusion [6], loss of situation awareness [7], as well as design factors like complexity, brittleness, opacity and literalism [8].

To tackle those human factors problems in the domain of aviation, cognitive automation and cognitive associate systems have been developed over the years to support the human pilots [9–13]. In order to keep the pilots in the loop and to not disturb the work process, this support should adapt to the tasks the pilot is currently conducting. Furthermore, the goal of adaptive automation is to keep the operators' workload at an average level to maintain human performance in extreme workload situations [14, 15]. Workload-adaptive associate systems use adaptive automation to support the human operators.

According to [16], MWL is a multidimensional construct, which is determined by characteristics of the task, of the operator and, to a degree, the environmental context. Our current approach towards engineering a workload-adaptive associate system is given by [17, 18]. It is based on a context-rich description of MWL. A basic assumption in this approach is, that the MWL of the operators follows qualitatively the current task load induced by the task situation [19]. Therefore, a task-centered design has been proposed. With the application in an associate system in mind, in this concept the construct MWL is operationalized context-rich in the form of a plan (the tasks an operator has to do), the current activity, demands on mental resources associated with this activity and observable behavior patterns. The necessary knowledge is stored in a common task model. In order to derive mental workload, the mission has to be planned [20], the current activity determined, demands on mental resources estimated [13] and behavior patterns analyzed [21]. A requirement for determining MWL in a dynamic mission and specific task situation is a reliable determination of the current activity of a human operator.

In this task-centered approach, the human operators are communicating with artificial team members. According to [22], anticipatory information sharing by implicit deliberative communication can improve team performance in complex task situations. Our task-centered method provides a means of implicit, non-verbal deliberative communication, since the human pilots inform the artificial crew member implicitly about their tasks and therefore their goals. Our method for activity determination lays a foundation for a better performance in mixed human and machine teams like the manned-unmanned teaming of a helicopter and UAVs. The following sections give deeper insights, how activity determination can be designed and implemented by using evidential reasoning.

2 Method

2.1 Requirements for Activity Determination

Activity determination can be regarded as a classification problem. If the current activity of a human operator is described by a set of tasks contained in a task model, the method must determine for every task, if the operator is currently conducting this task or not. For a successful and robust determination of the activity, different requirements must be met:

Activity determination must work in an environment, which is characterized by uncertainty and ignorance. Since there is no direct link into the human brain, measurement sensors like buttons, speech detectors or gaze trackers must be used to draw conclusions regarding the underlying cognitive processes. These sensors introduce uncertainty because each measurement has an individual measuring inaccuracy. A consequence is that generated hypotheses based on those measurements have limited reliability. Ignorance results from missing knowledge about model parameters and the environment. If a sensor is temporarily out of order, all measurements of this sensor are completely unreliably. This is for example the case for a gaze tracking system. Loss off gaze tracking can occur, if the test person is blinking or covering a camera. Not only the measurement equipment, but also the human behavior can be faulty. An easy example would be if a pilot forgets to set some system parameters like radio frequencies. Therefore, the algorithms must be able to deal with uncertainty and ignorance and weight the resulting hypotheses of the sensors according to their measurement accuracy. Since activity determination must work automatically in real-time as part of an associate system, these algorithms must also be fast enough. The cognitive processes inside the human brain run on a time scale of tens to hundreds of milliseconds [23]. Therefore, activity determination should also work on this time scale.

Methods that meet those requirements and which seem promising for activity determination are probability theory, Dempster-Shafer theory (DST) [24, 25] and certainty factors [26]. Because of its simplicity and mathematical foundation, probability theory, and especially Bayesian Networks [27], are suited for solving classification problems [28]. For example, Naïve Bayesian Classifiers can be used to filter junk email [29, 30]. Bayesian Networks model the reasoning process in the causal direction and use Bayes' Rule to solve diagnostic problems. The disadvantage of using probability theory is that all model parameters (a-priori and conditional probabilities) must be determined from statistical analysis prior using the model. Certainty factors are a heuristic approach, where the parameters are obtained from expert knowledge. Wittig has used certainty factors for determining pilot intent [31]. DST is also a theory based on a mathematical framework. From a certain point of view, it can be regarded as an extension of probability theory. In contrast to probability theory, DST distinguishes between uncertainty and ignorance. Considering ignorance enables an associate system to be aware of its own lack of knowledge, which might be useful when it comes to critical decisions. In those situations, it is usually better if the human operator decides what to do, rather than a system with uncomplete knowledge. DST is a method for evidential reasoning and extends probability theory by providing a rule of combination for diagnostic problems. The difficulty in DST is the interpretation of the parameters and how they are acquired. Beside certainty factors, DST has also be applied in the MYCIN expert system for computer based medical consultations [32].

Because of providing a rule how to combine evidences, the ability to consider both uncertainty and ignorance and its mathematical foundation, we suggest to take DST for pilot's activity determination. We are using a simplified version of DST based evidential reasoning because the drawback of the full theory is an exponential complexity in calculation and therefore difficult to implement in real-time [32].

2.2 Evidential Reasoning

Since the execution of tasks, especially cognitive tasks, cannot be directly observed, we are using an indirect method for classifying, whether a task is currently being executed by a pilot or not. The used evidential reasoning method is similar to the way a human observer would take (see Fig. 1). During the observation of the pilots, the technical system, and the environment, many observable facts are collected. Each of those facts proposes a hypothesis, whether an operator is currently executing a certain task or not. These single evidences are weighted according to their plausibility and combined into one single resulting hypothesis.

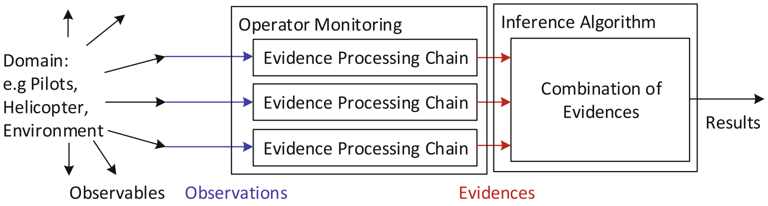


Fig. 1. Process of evidential reasoningin the domain of helicopter missions.

The underlying reasoning model is described in the Sects. 2.3 and 2.4. The processing chain from collecting observations to generate evidences is explained in Sect. 2.5. The inference algorithm for combining evidences to draw conclusions on the actual activity is explained in Sect. 2.6.

2.3 Representing Uncertainty and Ignorance

In probability theory, the strength of belief in a hypothesis X is represented by the scalar probability $P(X)$. $P(X)$ describes the belief in this hypothesis and $1 - P(X)$ the doubt. In contrast to probability theory, we represent the strength of belief in a hypothesis X by a belief triplet according to [33]:

$$\boldsymbol{Q}(X) = (p, q, r) \tag{1}$$

This triplet is a normalized distribution:

$$p + q + r = 1 \tag{2}$$

The belief quantities p, q and r can have continuous values in the range from 0 to 1. The quantity p is called belief and significates to which extent the hypothesis is supported, q the doubt, to which extent the hypothesis is rejected, and r the remaining ignorance. There are different interpretations of belief values. According to the interpretation of Dempster [24], the ignorance r describes uncertainty in quantifying probabilities, p is a lower probability value and $1 - q$ an upper probability value (plausibility), where the actual probability lies somewhere in between. Other interpretations do not try to relate belief values directly to probabilities. Shortliffe, for example, interprets belief values in certainty factor theory as an increase in information and gives the following example: “I don’t know what the probability is that all ravens are black, but I do know that every time you show me an additional black raven my belief is increased by X that all ravens are black” [26].

Similar to conditional probability tables, conditional belief distributions can be defined. These distributions are described in our method by matrixes:

$$\boldsymbol{M}_S(B|A) = \begin{pmatrix} p_t & p_f & 0 \\ q_t & q_f & 0 \\ r_t & r_f & 1 \end{pmatrix} \tag{3}$$

In this definition, the matrix components significate the following:

p_t: Belief, that $B = true$ if $A = true$.
q_t: Belief, that $B = false$ if $A = true$.
r_t: Belief, that $B = unknown$ if $A = true$.
p_f: Belief, that $B = true$ if $A = false$.
q_f: Belief, that $B = false$ if $A = false$.
r_f: Belief, that $B = unknown$ if $A = false$.

2.4 State Space Model

We are using a separate classification model for every possible task that can be conducted by a test person. The state space model for a single task, which classifies whether it is part of the current activity or not, is depicted in Fig. 2.

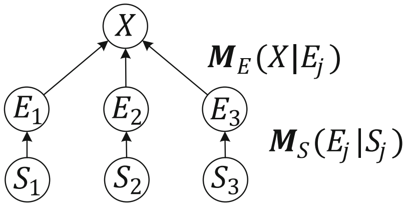


Fig. 2. State space model.

The state space model is a structured representation of the dependencies between a task, evidences and observations. It consists of nodes and edges and is similar to a Bayesian Network. The nodes symbolize state variables described by belief triplets (1). The root node is the state variable for the examined task X. Several evidence variables $E_1, \ldots, E_m$ are linked to the task X. Since each evidence is based on a hypothesis generated by a measurement sensor, each evidence variable E_j is connected to a sensor variable (observation) S_j. The edges in the state space model symbolize reasoning models described by conditional belief distributions (3). The arrows indicate the direction of belief propagation for diagnostic reasoning. There are two types of models: Sensor models and evidence models.

A sensor model describes the accuracy or reliability of a measurement sensor. Beside the general sensor model $\boldsymbol{M}_S(E_j|S_j)$ according to (3), we define some special types of a sensor model:

$$\boldsymbol{M}_S(E_j|S_j) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \text{Perfect Sensor} \tag{4}$$

$$\boldsymbol{M}_S(E_j|S_j) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \qquad \text{Unknown sensor} \tag{5}$$

$$\boldsymbol{M}_S(E_j|S_j) = \begin{pmatrix} P_t & P_f & 0 \\ 1 - P_t & 1 - P_f & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \text{Probabilistic sensor} \tag{6}$$

$$\boldsymbol{M}_S(E_j|S_j) = \begin{pmatrix} Z & 0 & 0 \\ 0 & Z & 0 \\ 1 - Z & 1 - Z & 1 \end{pmatrix} \qquad \text{Sensor with scalar reliability} \tag{7}$$

For a perfect sensor, both, uncertainty and ignorance are 0. An unknown sensor model is the contrary of a perfect sensor model and will always result in ignorance 1 for any input. For a Bayesian definition of the sensor model with no ignorance, the probabilistic model is used and for representing reliability and the amount of knowledge, a simple sensor model with a single scalar value Z might be used.

The evidence model describes the strength of an evidence for task execution under the assumption, that there is no measurement error by the sensor (perfect sensor model). In contrast to the sensor model, the evidence model contains human factors like uncertainty in human behavior. A general model of an evidence can be written in matrix form $\boldsymbol{M}_E(X|E_j)$ according to Eq. (3). Beside a perfect, unknown and probabilistic evidence (compare with sensor models), we consider two special types of evidence models:

$$\boldsymbol{M}_E(X|E_j) = \begin{pmatrix} p & 0 & 0 \\ 0 & 0 & 0 \\ 1 - p & 1 & 1 \end{pmatrix} \qquad \text{Supporting evidence (belief)} \tag{8}$$

$$\boldsymbol{M}_E(X|E_j) = \begin{pmatrix} 0 & 0 & 0 \\ q & 0 & 0 \\ 1 - q & 1 & 1 \end{pmatrix} \qquad \text{Rejecting evidence (doubt)} \tag{9}$$

A supporting or belief evidence, increases the strength of belief that the operator is currently performing a task, whereas a rejecting or doubt evidence increases the strength of doubt in that hypothesis.

2.5 Evidence Processing Chain

For each evidence contained in the state space model, a processing chain is used to calculate its belief values and therefore evidential strength (Fig. 3).

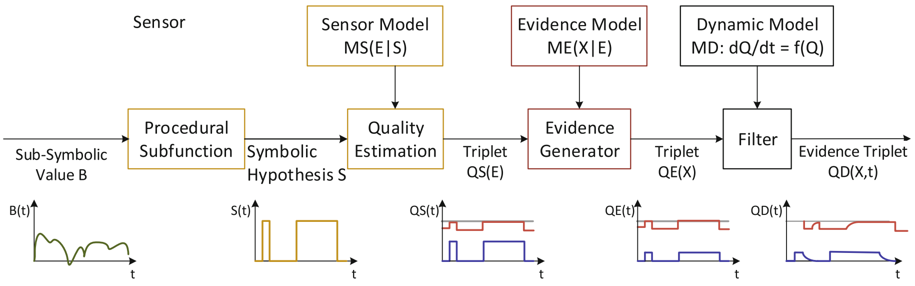


Fig. 3. Processing chain for evidences.

The raw information about the test persons and their environment is given in an inhomogeneous and sub-symbolic form. In the first step (procedural sub-functions), symbolic hypotheses *S* (observations) are derived from sub-symbolic values *B*. To be compatible with classical logic and Bayesian models, each observation is described by a state variable (see Sect. 2.4), which can be either *true*, *false*, or *unknown* if the sensor is broken or temporarily out of order. This value is then transformed into a belief triplet:

$$\boldsymbol{Q}_S(S_j) = \begin{cases} (1,0,0)^T & if \quad S_j = true \\ (0,1,0)^T & if \quad S_j = false \\ (0,0,1)^T & if \quad S_j = unknown \end{cases} \tag{10}$$

The reliability of this hypothesis is calculated as a matrix-vector multiplication of the sensor model with this belief triplet:

$$\boldsymbol{Q}_S(E_j) = \boldsymbol{M}_S(E_j|S_j)\boldsymbol{Q}_S(S_j) \tag{11}$$

Then the strength of a single evidence for a given point in time is derived from a forward propagation of the belief values similar to [34, 35]. For this calculation, the evidence model is used:

$$\boldsymbol{Q}_{Ej}(X) = \boldsymbol{M}_E(X|E_j)\boldsymbol{Q}_S(E_j) \tag{12}$$

In a time-dependent environment, it is not sufficient to describe evidences only for a given point in time. A dynamic model describes, how long an observed evidence is valid and is indispensable if short events like button presses are used as evidences. A general dynamic model can be defined as an initial value problem, i.e. a differential equation with initial conditions:

$$\begin{array}{rcl} \boldsymbol{M}_D\colon \ \frac{d}{dt}\boldsymbol{Q}_{Dj}(X,t) & = & \boldsymbol{f}\left(\boldsymbol{Q}_{Dj}(X,t)\right) \\ \boldsymbol{Q}_{Dj}(X,0) & = & \boldsymbol{Q}_{Ej}(X) \end{array} \tag{13}$$

We suggest a simple dynamic model, where the belief p and doubt q are decaying after the evidence has been observed. The rate of decay is assumed to be proportional to the current belief values:

$$\begin{array}{l} \frac{dp(t)}{dt} = \lambda p(t) \ \rightarrow \ dp = \lambda p\, dt \\ \frac{dq(t)}{dt} = \lambda q(t) \ \rightarrow \ dq = \lambda q\, dt \end{array} \tag{14}$$

Solving this differential equation leads to an exponential decay over time:

$$\begin{array}{rcl} p_{Dj}(X,t) & = & p_{Ej}(X)\ \exp(-\lambda t) \\ q_{Dj}(X,t) & = & q_{Ej}(X)\ \exp(-\lambda t) \\ r_{Dj}(X,t) & = & 1 - p_{Dj}(X,t) - q_{Dj}(X,t) \end{array} \tag{15}$$

This model imitates the intuitive observation, that knowledge is getting lost over time and ignorance increases. The decay parameter λ of this model can be expressed as a half-life value, which indicates the time, after which the belief is half of the originally observed one:

$$t_{\frac{1}{2}} = \frac{\ln(2)}{\lambda} \tag{16}$$

2.6 Combination of Evidences

After generating the evidences, in favor for or against the execution of a single task, all evidences are combined into a single resulting hypothesis. This is done by using a rule of combination, which is derived from Dempster's rule of combination for a binary frame of discernment. With this rule, two evidences are combined by:

$$\boldsymbol{Q}_1(X) \oplus \boldsymbol{Q}_2(X)\colon = \boldsymbol{Q}_{12}(X) = \begin{pmatrix} p_{12}(X) \\ q_{12}(X) \\ r_{12}(X) \end{pmatrix} \tag{17}$$

$$\begin{aligned} p_{12}(X) &= \frac{p_1p_2 + p_1r_2 + r_1p_2}{1-(p_1q_2 + q_1p_2)} \\ q_{12}(X) &= \frac{q_1q_2 + q_1r_2 + r_1q_2}{1-(p_1q_2 + q_1p_2)} \\ r_{12}(X) &= 1 - p_{12}(X) - q_{12}(X) \end{aligned}$$

All evidences resulting from the processing chain are combined by using this rule of combination iteratively:

$$\boldsymbol{Q}_{total}(X,t) = \boldsymbol{Q}_{D1}(X,t) \oplus \boldsymbol{Q}_{D2}(X,t) \oplus \ldots \oplus \boldsymbol{Q}_{Dm}(X,t) \quad (18)$$

The order of the evidences does not matter since the combination rule given above is both associative and commutative. Tasks, for which the resulting triplets $\boldsymbol{Q}_{total}$ have high belief and low doubt values, are considered as part of the current activity of a human operator.

3 Implementation in a Helicopter Mission Simulator

3.1 System Overview

The described method for activity determination is implemented in the helicopter mission simulator of the Institute of Flight Systems at the Bundeswehr University in Munich. It is part of a situation and workload-adaptive pilot associate system for MUM-T helicopter missions. Figure 4 shows an overview of the system for activity determination.

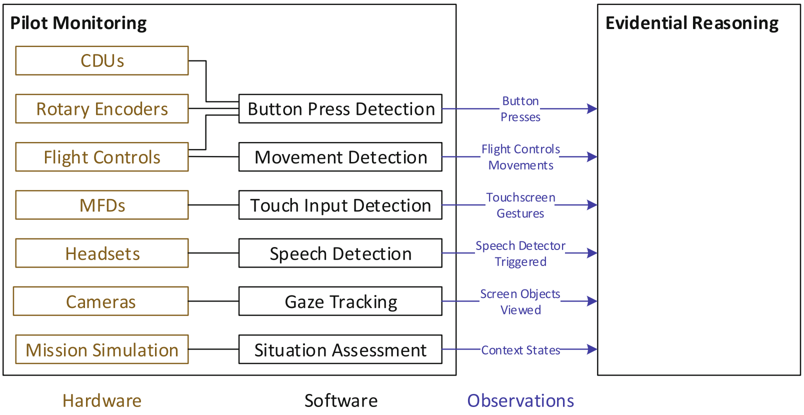
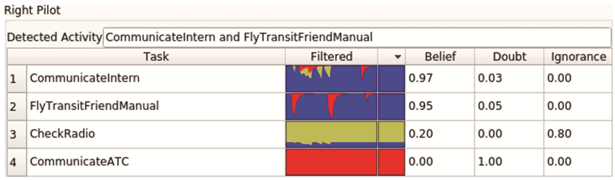


Fig. 4. System overview of the activity determination.

Firstly, observations are created during pilot monitoring, then the actual activity is inferred during evidential reasoning.

As hardware part of the sensors, different input devices are used. Input devices are the command and display units (CDUs), rotary encoders, flight controls, touch-sensitive multi-function displays (MFDs) [36], microphones in the headsets of the pilots and cameras for a gaze tracking system. Furthermore, states of the mission simulations are considered to capture the task context. For each type of these hardware sensors, we calculate observations consisting of symbolic detection values and quality measures to estimate sensor reliability in different software modules. These modules implement the low-level, procedural signal processing sub-functions of the processing chain described in Sect. 2.5. We explain the process of pilot monitoring for each different sensor type below (Sects. 3.2–3.8).

The inference algorithm is implemented in C++ computer code in the program PAD. The program PAD uses a separate thread for every pilot and does calculations in parallel. The observations from different sensors may have different signal propagation delays. Therefore, in the first step, those delays are corrected by using the actual times of the measurements. This is important to maintain causality and prevent reasoning problems, if later created signals arrive earlier at the activity determination. Latencies in the signals result from inter-process communications between the sensors and the PAD program. Then, the sensor model $\boldsymbol{M}_S(E_j|S_j)$ for a scalar reliability value Z (7) is used to derive the observation triplet for every evidence $\boldsymbol{Q}\left(E_j\right)$. The processing chain and the rule of combination are implemented as described in Sects. 2.5 and 2.6. Figure 5 shows an excerpt from the resulting belief distributions.



Right Pilot

Detected Activity: CommunicateIntern and FlyTransitFriendManual

	Task	Filtered	Belief	Doubt	Ignorance
1	CommunicateIntern		0.97	0.03	0.00
2	FlyTransitFriendManual		0.95	0.05	0.00
3	CheckRadio		0.20	0.00	0.80
4	CommunicateATC		0.00	1.00	0.00

Fig. 5. Example results for activity determination, screenshot of program PAD. Here, the pilot is flying a manual transit flight and communicating via the intercom at the same time.

The final step is to extract the actual activity of the pilots from those distributions. That means a classification decision must be made. Therefore, the belief triplet must be interpreted. We consider a task as part of the activity, if the belief is the greatest value:

$$p > q + r \tag{19}$$

Since the belief triplet is normalized (2), this criterion is equal to

$$p > 0.5. \tag{20}$$

In practice, taking just the threshold value might result in oscillations and causality problems for very short events (e.g. button presses and state changes at the same time). Therefore, we add a last filtering step. This step is done by integrating the belief over a short time. To select the actual tasks of the activity, an impact value is calculated while the belief is greater than 0.5:

$$I = \int\limits_{while\, p(t)>0.5} p(t)\, dt \tag{21}$$

If the impact of the evidence is high enough (in our simulator a threshold value of $I = 0.4$ is used), this task is classified as part of the current activity. The integration is stopped if the current belief drops below 0.5. The drawback of this last filter step is an additional time delay of the detection decision. The integration time depends on the strength of belief. The higher p, the shorter the integration time and the faster the decision.

3.2 General Rule for Calculating Sensor Reliability for a Continuous Value

An observable is a measurable quantity in the simulator. Most of the sub-symbolic observables are continuous, time-dependent signals. If an observable is observed during the process of activity determination (i.e. a detector has been triggered), we call it observation. In the implementation, we are using the scalar reliability sensor model (7). Therefore, every observation is described as symbolic hypothesis S along with a scalar reliability or quality value Z.

For generating a symbolic value from a sub-symbolic, continuous quantity x in the procedural sub-functions (Sect. 2.5), we are using a general threshold criterion to distinguish if a detector has been triggered or not:

$$S(x) = \begin{cases} true & if & x \geq x_d \\ false & if & x < x_d \\ unknown & if & sensor\ out\ of\ order \end{cases} \tag{22}$$

The scalar reliability value is calculated as linear distance to the threshold x_d:

$$\mathrm{Z}(x) = \begin{cases} \dfrac{x - x_d}{x_{max} - x_d} & if & x \geq x_d \\ {[1mm]}\dfrac{x - x_d}{x_{min} - x_d} & if & x < x_d \\ {[1mm]}0 & if & sensor\ out\ of\ order \end{cases} \tag{23}$$

A perfect measurement corresponds to the reliability $Z = 1$ and a totally unreliable measurement corresponds to the reliability $Z = 0$. For practical purposes, the reliability value is bounded, i.e. values below zero are set to 0 and values above 1 are set to 1. Beside the discrimination threshold x_d, this model requires a lower limit x_{min} and an upper limit x_{max} as parameters, which are defined below for every sensor type.

3.3 Button Press Detection

In the cockpit, there are many buttons the pilots can press. Examples are buttons of the CDUs or buttons on the grip of the flight controls (Fig. 6). Buttons can either represent system states (e.g. radio button) or short events, if they are pressed (e.g. line selection keys on the CDU).

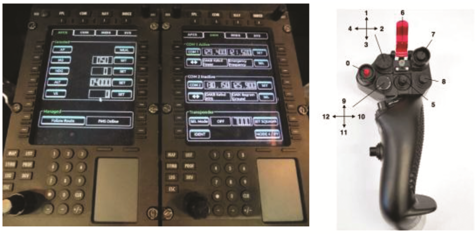


Fig. 6. CDUs and flight controls with buttons [37].

Buttons in aviation must be extremely reliable. Even buttons in the simulator are highly reliable and we do not expect that they fail or deliver inaccurate results during a typical mission of less than two hours. Therefore, a button press is described by a perfect sensor model (4), which is equal to a scalar reliability model (7) with $Z = 1$:

$$S = \begin{cases} true \;\; if \;\; button\,pressed \\ false \;\; if \;\; button\,not\,pressed \end{cases} \tag{24}$$

$$Z = 1 \tag{25}$$

If the state of a button can take more than two discrete values, which can be the case for switches and rotary buttons, a binary model is generated for every possible value.

3.4 Movement Detection of the Flight Controls

For steering the simulated helicopter, a control load system of Reiser Simulation and Training GmbH has been integrated into the simulator (Fig. 7). The system consists of a cyclic stick, which controls the pitch and roll movement of the helicopter, pedals which control the yaw movement and a collective lever which controls the collective pitch of the rotor blades. The flight controls of both pilots are electrically coupled so that they are physically performing the same movements in parallel.

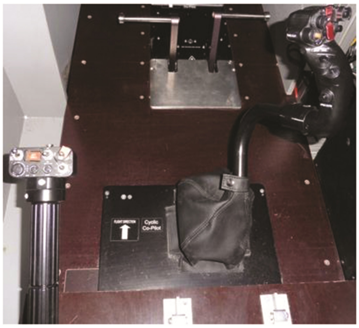


Fig. 7. Flight controls of the helicopter simulator: collective, cyclic and pedals.

The movement detection of the flight controls is based on measuring the rate $r(t)$ (i.e. the time derivative) of the flight controls signal $x(t)$ of each axis:

$$r(t) = \left|\frac{dx(t)}{dt}\right| \tag{26}$$

If one of the rates for the different axes is greater than the threshold r_d, the stick is considered as moved by the pilot:

$$S(r) = \begin{cases} true & if \quad r \geq r_d \\ false & if \quad r < r_d \\ unknown & if \quad sensor\, out\, of\, order \end{cases} \tag{27}$$

For estimating the detection quality, the lower limit in Eq. (23) is no movement at all ($x_{\text{min}} = 0$). The upper limit $x_{\max}$ is determined during the calibration for a fast movement of the flight controls. The reliability value depends on the rate of movement:

$$Z(r) = \begin{cases} \dfrac{r - r_d}{r_{max} - r_d} & if \quad r \geq r_d \\ [2mm]\dfrac{(r_d - r)}{r_d} & if \quad r < r_d \\ [1mm]0 & if \quad sensor\, out\; of\; order \end{cases} \tag{28}$$

3.5 Speech Detection

The goal of speech detection is to measure the auditory interactions of the pilots with the system and detect if they are speaking or not. There is currently no interpretation of the speech signal as such.

The raw data is the amplitude signal $A(t)$ of a microphone, which is integrated in the head set of a pilot (see Fig. 8). Speech detection is based on calculating the power level

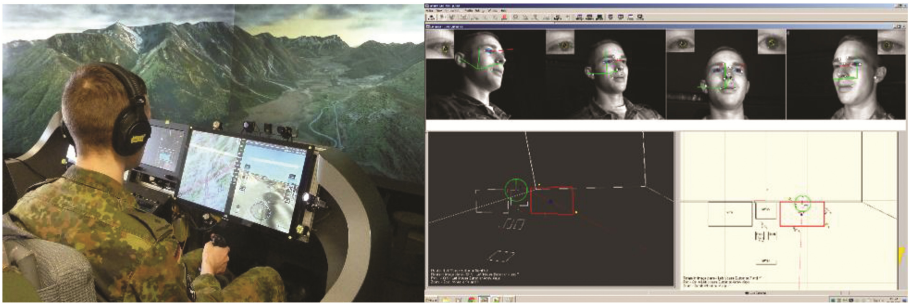


Fig. 8. Integrated gaze tracking system Smart Eye Pro [38].

P by averaging the squared amplitude over a short time interval Δt (here a few hundred milliseconds).

$$P(t) = \frac{1}{\Delta t} \int_{t'=t-\Delta t}^{t} dt' A^2(t') \tag{29}$$

For detecting speech, a power level P_d is used as the discrimination threshold x_d in Eq. (22):

$$S(P) = \begin{cases} true & if \quad P \geq P_d \\ false & if \quad P < P_d \\ unknown & if \quad sensor\ out\ of\ order \end{cases} \tag{30}$$

Two distinct power levels can be identified. The first one is the signal level if the pilot is speaking P_{signal}, the second one is the noise level P_{noise} if the pilot is not speaking, but the signal is still disturbed by noise from the environment. These parameters result from a calibration, where the power level is recorded for some seconds. The closer the current power level P is to the detection threshold, the worse is the reliability of the measurement:

$$Z(P) = \begin{cases} \dfrac{P - P_d}{P_{signal} - P_d} & if \quad P \geq P_d \\ [2mm]\dfrac{P - P_d}{P_{noise} - P_d} & if \quad P < P_d \\ [1mm]0 & if \quad sensor\ out\ of\ order \end{cases} \tag{31}$$

3.6 Gaze Tracking

For the purpose of getting evidences from visual interactions with the cockpit displays, a commercial gaze tracking system (Smart Eye Pro, Smart Eye AB) has been implemented in the simulator [38]. It is a video-based system, which consists of four cameras for the left pilot and four cameras for the right pilot running at 60 Hz. The cameras are placed around the MFD (see left image of Fig. 8). The gaze tracking method is based on measuring the cornea reflection in the infrared spectrum. There is no intrusion of the pilots in their working domain. A 3D world model of the simulator displays and the outside view has been implemented in the system [38] (see right image of Fig. 8 in the lower left corner).

The raw sub-symbolic values of the eye tracker are pixel coordinates on a simulator display. For creating evidences, semantic information about the objects the pilots are looking at are necessary. These objects can be the airspeed indicator or tactical symbols in the tactical map and are provided by the MFDs [36]. The semantic information on which display, page and object the pilot is looking on is gained by combining the raw pixel coordinates with the layout on the cockpit displays (Fig. 9).

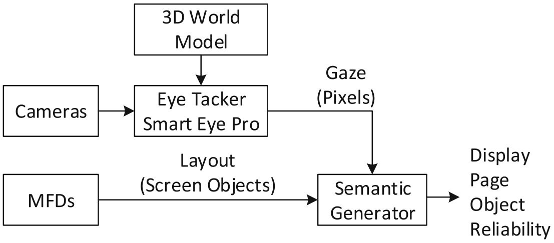
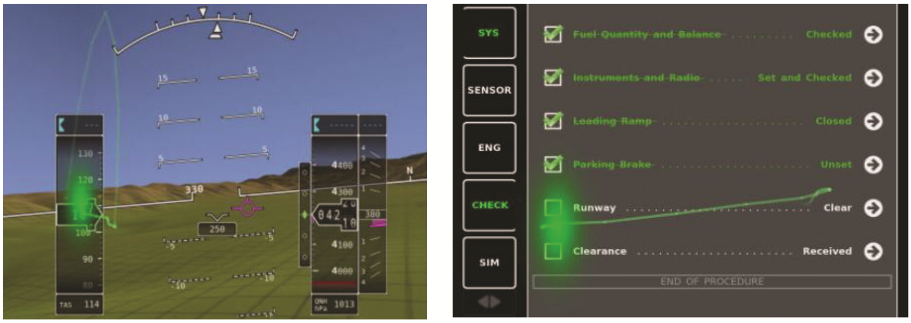


Fig. 9. Gaze tracking system for deriving object oriented semantic data.

In order to take account for measurement inaccuracies, the gaze on the screen is not only described by single pixel coordinates, but rather by a normalized Gaussian distribution over the 2D surface (see Fig. 10):

$$f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} exp\left(-\frac{1}{2(1-\rho^2)}\left[\frac{(x-x_0)^2}{\sigma_x^2} + \frac{(y-y_0)^2}{\sigma_y^2} - \frac{2\rho(x-x_0)(y-y_0)}{\sigma_x\sigma_y}\right]\right) \tag{32}$$



☑ ~~Fuel Quantity and Balance~~ .... Checked

☑ ~~Instruments and Radio~~ ...... Set and Checked

☑ ~~Loading Ramp~~ .... Closed

☑ ~~Parking Brake~~ .... Unset

□ Runway .... Clear

□ Clearance .... Received

END OF PROCEDURE

Fig. 10. Gaze tracking on PFD and during processing a check list.

The raw pixel coordinates of the current gaze coordinates delivered by the gaze tracker are given by x_0 and y_0. The widths of the gaze distribution in the x- and y- screen axis directions are expressed by two different standard deviations σ_x and σ_y. The Pearson correlation coefficient ρ indicates the correlation between the two axes and therefore describes the shear deformation of the gaze spot. These parameters are not constant over the screen, but depend on the current gaze position x_0 and y_0:

$$\sigma_x = \sigma_x(x_0, y_0) \;\; \sigma_y = \sigma_y(x_0, y_0) \;\; \rho = \rho(x_0, y_0) \tag{33}$$

We are using bilinear models to describe the dependence of the parameters on the screen position:

$$\begin{aligned} \sigma_x(x_0, y_0) &= \beta_{x1} + \beta_{x2}x_0 + \beta_{x3}y_0 + \beta_{x4}x_0y_0 \\ \sigma_y(x_0, y_0) &= \beta_{y1} + \beta_{y2}x_0 + \beta_{y3}y_0 + \beta_{y4}x_0y_0 \\ \rho(x_0, y_0) &= \beta_{r1} + \beta_{r2}x_0 + \beta_{r3}y_0 + \beta_{r4}x_0y_0 \end{aligned} \tag{34}$$

The constant parameters of these bilinear models are gained during the calibration process of the gaze tracking system individually for each pilot. During this calibration, multiple gaze points on the screen are sampled. Hereby, a least square problem is solved. This is done by a QR-decomposition contained in the Eigen C++ library [39].

Every object displayed on a screen is represented by a polygon of pixel coordinates (see Fig. 11). For determining a symbolic hypothesis S, if the pilot is looking on this object or not, and a corresponding scalar reliability value Z (see Sect. 3.2), the integral of the gaze distribution over each single screen object is calculated.

$$I = \iint_{Screen\ Object} dx\, dy f(x, y) \tag{35}$$

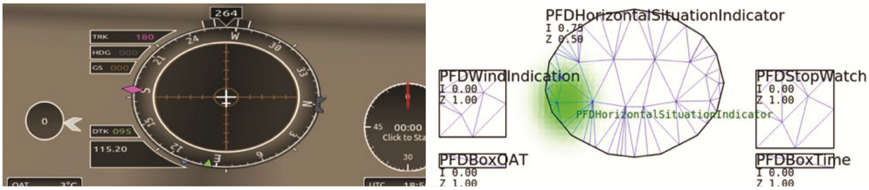


Fig. 11. Example for estimating the reliability of a gaze tracking evidence. The left image shows an excerpt from the PFD and the right image the triangulated screen objects with the Gaussian gaze distribution on the horizontal situation indicator (green spot). (Color figure online)

This integration is calculated technically, by decomposing the polygonal screen objects into triangles with a Delaunay triangulation algorithm [40] [41, pp. 1131–1141],and integrating over the individual triangles by using a Gauss-Legendre quadrature [41, pp. 179–200]. Figure 11 show an example for screen objects on the primary flight display (PFD).

With Eqs. (22) and (23) ($x_{min} = 0, x_{max} = 1, x_d = 0.5, x = I$), the symbolic value and reliability are given by:

$$S(I) = \begin{cases} true \;\; if \;\; I \geq 0.5 \\ false \;\; if \;\; I < 0.5 \end{cases} \tag{36}$$

$$Z(I) = \begin{cases} 2(I - 0.5) \;\; if \;\; I \geq 0.5 \\ -2(I - 0.5) \;\; if \;\; I < 0.5 \end{cases} \tag{37}$$

3.7 Touchscreen Input Detection

The multi-function displays (MFDs) in the simulator cockpit [36] are equipped with a touch-sensitive surface (Fig. 12). The pilot can interact with the MFDs by tapping on the surface with one or more fingers. Furthermore, the displays support pan, swipe and pinch gestures.

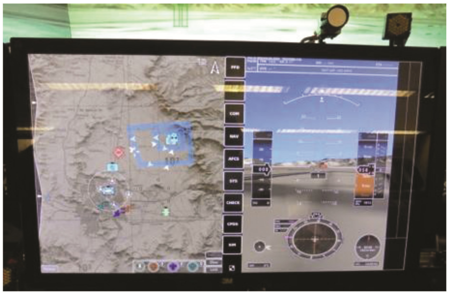


Fig. 12. Multi-touch multi-function Display (MFD) in the simulator cockpit.

Touchscreen inputs may be faulty since it is not easy to hit small buttons or symbols on the touchscreen with the finger. Therefore, an error model for touchscreen inputs is assumed. Similar to the error model for the gaze tracker, the touch point is not only a single point, but rather an axial symmetric Gaussian distribution:

$$\begin{aligned} f(x,y) &= \frac{1}{2\pi\sigma^2} exp\left(-\frac{r(x,y)^2}{2\sigma^2}\right) \\ r(x,y) &= \sqrt{\left(x-x_0\right)^2 + \left(y-y_0\right)^2} \end{aligned} \tag{38}$$

The measured coordinates of the touch event are x_0 and y_0. The parameter σ defines the width of this distribution. This parameter is determined during calibration as the standard deviation of some touch samples. The calculation for deriving the symbolic value S and the reliability Z on the basis of this distribution is the same as for the gaze tracking system (see Sect. 3.6).

3.8 Situation Assessment

Not only the current observations from monitoring the pilots is important, but also the task context given by the situation. Therefore, the tactical situation and system states are also collected to feed the evidential reasoning algorithm. Beside threats from enemy forces, we are considering flight states and flight phases of the helicopter, states of the cockpit-displays, as well as states of the UAVs and their sensors. Weather, time of day and other environmental quantities are neglected in the current setup. Evidences, of the context are mainly used to create rejecting evidences (9). For simplicity, no real situation sensors are modeled at the moment but all observed quantities are taken as perfect reliable observations similar to button presses.

4 Current Status and Future Work

The method presented in Sect. 2 has been implemented in our helicopter mission simulator as described in Sect. 3. The developed software modules are part of a workload-adaptive associate system, which uses a common task model as central representation of task oriented knowledge [18]. To determine all the possible tasks, which can occur during a mission, a task analysis has been performed together with helicopter pilots and UAV operators of the German Armed Forces [42].

The task model contains currently 226 tasks in total and 85 of them are used for activity determination. The other tasks are abstract and used for hierarchical structuring the model and for other functions of the associate system like mission planning. Currently, there are about 1300 automatically generated observables. 440 of them are linked to the 85 tasks in the task model as evidences by a knowledge engineer. With the help of an inheritance mechanism between tasks in the task model about 730 evidences are generated in total. This results in an average of 8.6 evidences per task. Details about abstract tasks and inheritance relations are described in [18].

The total processing time of the algorithms running in the simulator takes between 40 and 80 ms on a state-of-the-art personal computer and therefore complies with the soft real-time requirement stated in Sect. 2.1.

The system is currently being hardened and the parameters are tuned. In the future, full mission simulations with aviators of the German Armed Forces are planned to evaluate this method for activity determination and the closed loop with activity determination as part of the embracing associate system.

References

1. Uhrmann, J., Strenzke, R., Schulte, A.: Task-based guidance of multiple detached unmanned sensor platforms in military helicopter operations. COGIS (COGnitive systems with Interactive Sensors) (2010)
2. Whittle, R.: MUM-T is the word for AH-64E: Helos Fly, Use Drones. Breaking Defense (2015). http://breakingdefense.com/2015/01/mum-t-is-the-word-for-ah-64e-helos-fly-use-drones/. Accessed 4 Dec 2016
3. Cooley, M.: Human centred systems: An urgent problem for systems designers. AI Soc. **1**(1), 37–46 (1987)
4. Wiener, E.L., Curry, R.E.: Flight-deck automation: promises and problems. Ergonomics **23**(10), 995–1011 (1980)
5. Wiener, E.L.: Human Factors of Advanced Technology (“Glass Cockpit”) Transport Aircraft. NASA Contractor Report No. 177528, Moffett Field, CA (1989)
6. Sarter, N.B., Woods, D.D.: How in the world did we ever get into that mode? mode error and awareness in supervisory control. Human Factors J. Hum. Factors Ergonomics Soc. **37**(1), 5–19 (1995)
7. Endsley, M.R.: Automation and situation awareness. In: Parasuraman, R., Mouloua, M. (eds.) Automation and Human Performance: Theory and Applications, pp. 163–181. Lawrence Erlbaum Associates, Mahwah (1996)
8. Billings, C.E.: Benefits and costs of aviation automation. Aviation Automation: The Search for a Human-Centered Approach, pp. 181–218. Lawrence Erlbaum Associates, Mahwah (1997)
9. Onken, R., Prévot, T.: CASSY - cockpit assistant system for IFR Operation. In: 19th ICAS Congress Proceedings, vol. 3, pp. 2598–2608, Anaheim, CA (1994)
10. Walsdorf, A., Onken, R., Eibl, H., Helmke, H., Suikat, R., Schulte, A.: The crew assistant military aircraft (CAMA). In: The Human-Electronic Crew: The Right Stuff? 4th Joint GAF/RAF/USAF Workshop on Human-Computer Teamwork, Kreuth (1997)

11. Miller, C.A., Hannen, M.D.: User acceptance of an intelligent user interface: a rotorcraft pilot's associate example. In: Proceedings of the 4th International Conference on Intelligent User Interfaces, IUI 1999, pp. 109–116, Redondo Beach, CA (1999)
12. Onken, R., Schulte, A.: System-Ergonomic Design of Cognitive Automation: Dual-Mode Cognitive Design of Vehicle Guidance and Control Work Systems. System-Ergonomic Design of Cognitive Automation. Springer, Heidelberg (2010)
13. Maiwald, F., Schulte, A.: Enhancing military helicopter pilot assistant systems through resource adaptive dialogue management. In: Vidulich, M.A., Tsang, P.S., Flach, J.M. (eds.), Advances in Aviation Psychology. Ashgate Studies in Human Factors and Flight Operations, pp. 177–196. Ashgate Publishing, Ltd., Farnham (2014)
14. Parasuraman, R., Bahri, T., Deaton, J.E., Morrison, J.G., Barnes, M.: Theory and Design of Adaptive Automation in Aviation Systems. NAWCADWAR-92033-60. Naval Air Warfare Center, Aircraft Division, Warminster (1992)
15. Veltman, J.A., Jansen, C.: The Role of Operator State Assessment in Adaptive Automation. TNO Defence Security and Safety, Soesterberg (2006). TNO-DV3 20
16. Young, M.S., Brookhuis, K.A., Wickens, C.D., Hancock, P.A.: State of science: mental workload in ergonomics. Ergonomics **58**(1), 1–17 (2015)
17. Schulte, A., Donath, D., Honecker, F.: Human-system interaction analysis for military pilot activity and mental workload determination. In: IEEE International Conference on Systems, Man, and Cybernetics, SMC 2015, pp. 1375–1380, Kowloon Tong (2015)
18. Honecker, F., Brand, Y., Schulte, A.: A task-centered approach for workload-adaptive pilot associate systems. In: Proceedings of the 32rd Conference of the European Association for Aviation Psychology – Thinking High and Low: Cognition and Decision Making in Aviation, Cascais (2016)
19. O'Donnell, R., Eggemeier, T.: Workload assessment methodology. In: Boff, K., Kaufman, L., Thomas, J. (eds.) Handbook of Perception and Human Performance. Wiley, New York (1986)
20. Schmitt, F., Schulte, A.: Mixed-initiative mission planning using planning strategy models in military manned-unmanned teaming missions. In: IEEE International Conference on Systems, Man, and Cybernetics, SMC 2015, pp. 1391–1396, Kowloon Tong (2015)
21. Donath, D., Schulte, A.: Behavior based task and high workload determination of pilots guiding multiple UAVs. In: 6th International Conference on Applied Human Factors and Ergonomics and the Affiliated Conferences, AHFE 2015, vol. 3, pp. 990–997. Elsevier, Las Vegas (2015)
22. Butchibabu, A., Sparano-Huiban, C., Sonenberg, L., Shah, J.: Implicit coordination strategies for effective team communication. Hum. Factors **58**(4), 595–610 (2016)
23. Eagleman, D.M.: Time and the brain: how subjective time relates to neural time. J. Neurosci. **25**(45), 10369–10371 (2005)
24. Dempster, A.P.: Upper and lower probabilities induced by a multivalued mapping. Ann. Math. Stat. **38**(2), 325–339 (1967)
25. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, Princeton, London (1976)
26. Shortliffe, E.H.: Computer-Based Medical Consultations: MYCIN (Artificial Intelligence Series). Elsevier Science Ltd., Amsterdam (1976)
27. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers Inc, San Francisco (1988)
28. Hanson, R., Stutz, J., Cheeseman, P.: Bayesian Classification Theory. Technical report FIA-90-12-7-01. NASA Ames Research Center. Artificial Intelligence Research Branch, Moffett Field (1991)

29. Lewis, D.D., Ringuette, M.: A comparison of two learning algorithms for text categorization. In: Third Annual Symposium on Document Analysis and Information Retrieval, vol. 33, pp. 81–93 (1994)
30. Sahami, M., Dumais, S., Heckerman, D., Horvitz, E.: A Bayesian approach to filtering junk e-mail. Learning for Text Categorization: Papers from the AAAI Workshop. Technical report WS-98-05, 62, pp. 98–105 (1998)
31. Wittig, T.: Maschinelle Erkennung von Pilotenabsichten und Pilotenfehlern über heuristische Klassifikation. Fortschritt-Berichte VDI. VDI-Verlag, München (1994)
32. Gordon, J., & Shortliffe, E. H. (1984). The Dempster-Shafer theory of evidence. Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project, vol. 3, pp. 832–838 (1984)
33. Dempster, A.P.: The Dempster-Shafer calculus for statisticians. Int. J. Approximate Reasoning **48**(2), 365–377 (2008)
34. Smets, P.: Belief functions: the disjunctive rule of combination and the generalized Bayesian theorem. Int. J. Approximate Reasoning **9**(1), 1–35 (1993)
35. Yaghlane, B.Ben, Mellouli, K.: Inference in directed evidential networks based on the transferable belief model. Int. J. Approximate Reasoning **48**(2), 399–418 (2008)
36. Brand, Y., Schulte, A.: Human agent interfaces as a key element for the dialog between human crews and cognitive automation. AIAA Infotech@ aerospace, pp. 1–12 (2015)
37. Weinmann, A., Brand, Y., Honecker, F., Meyer, C., Rudnick, G., Ruf, C., Schmitt, F.: Flight and user manual IFS helicopter mission simulator. Institute of Flight Systems. Aerospace Enineering Department. Bundeswehr University Munich, Munich (2016)
38. Mehler, J.: Integration und Evaluierung eines Blickbewegungsmesssystems für Pilot und Kommandant in einen Hubschraubersimulator. Universität der Bundeswehr München, Bachelorarbeit (2014)
39. Jacob, B., Guennebaud, G., et al.: Eigen: C++ template library for linear algebra. Version 3. https://eigen.tuxfamily.org. Accessed 27 Jan 2017
40. Delaunay, B.: Sur la sphère vide. A la mémoire de Georges Voronoi. Bulletin de l'Académie des Sciences de l'URSS, Classe des sciences mathématiques et na **6**, 793–800 (1934)
41. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes - The Art of Scientific Computing, 3rd edn. Cambridge University Press, New York (2007)
42. Winkler, B.: Design und Evaluierung von militärischen MUM-T-Missionen mittels Expertenwissen als Grundlage für zukünftige Messkampagnen. Universität der Bundeswehr München, Bachelorarbeit (2016)