Interner Bericht

DLR-IB-FT-BS-2025-198

Development of an Online Algorithm for Classifying Eye-Tracking Data of Helicopter Pilots

Hochschulschrift

Talha Sor

Deutsches Zentrum für Luft- und Raumfahrt

Institut für Flugsystemtechnik
Braunschweig

DLR Deutsches Zentrum
für Luft- und Raumfahrt

Institutsbericht
DLR-IB-FT-BS-2025-198

Development of an Online Algorithm for Classifying Eye-Tracking Data of Helicopter Pilots

Talha Sor

Institut für Flugsystemtechnik
Braunschweig

Deutsches Zentrum für Luft- und Raumfahrt e.V.
Institut für Flugsystemtechnik
Abteilung Hubschrauber

Stufe der Zugänglichkeit: I, Allgemein zugänglich: Der Interne Bericht wird elektronisch ohne Einschränkungen in ELIB abgelegt.

Braunschweig, den 05.12.2025

Unterschriften:

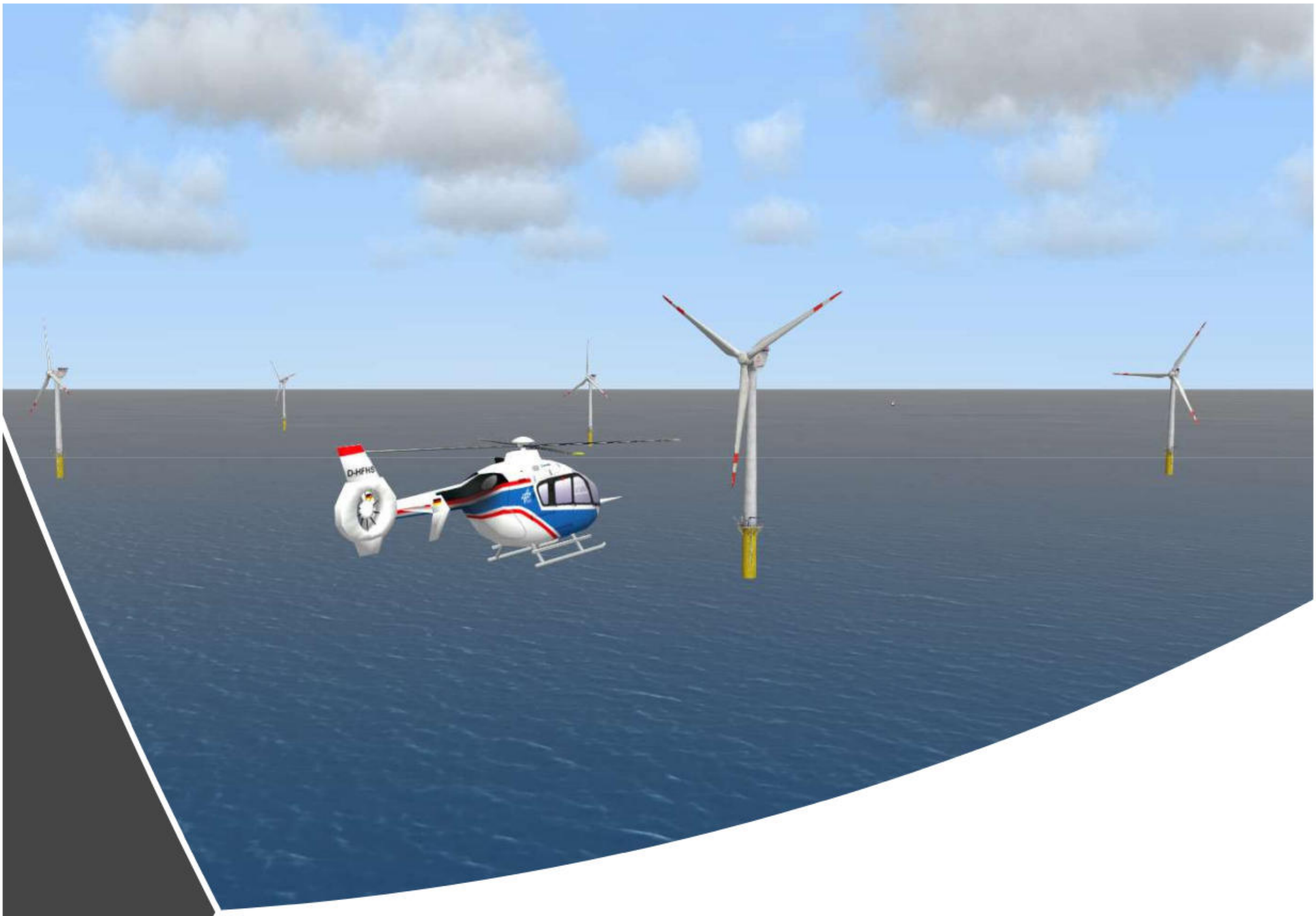
[redacted]

Institutsleitung:

Abteilungsleitung: Dipl.-Ing. Marc Höfinger

Betreuer:in: Daniel H. Greiwe, M.Sc.

Verfasser:in: Talha Sor



Development of an Online Algorithm for Classifying Eye-Tracking Data of Helicopter Pilots

Master Thesis

DLR

DLR German Aerospace Center
Institute of Flight Systems
Rotorcraft
Dr.-Ing. Klausdieter Pahlke and Dipl.-Ing. Marc Höfinger
Lilienthalplatz 7
38108 Braunschweig
Germany
Tel: +49 531 295-2691
Fax: +49 531 295-2641
Web: https://www.dlr.de/ft/en/hub

Talha Sor
Tel: +49 531 295-1473
Mail: talha.sor@rwth-aachen.de

Document Identification:

Report number	IB-2025-198
Title	Development of an Online Algorithm for Classifying Eye-Tracking Data of Helicopter Pilots
Subject	Master Thesis
Author(s)	Talha Sor
Filename	thesis.pdf
Last saved on	5th December 2025

Acknowledgements

Danken möchte ich Daniel Greiwe, M.Sc. vom DLR, und Nikolas Schierhorst, M.Sc. vom IAW, die ein Auge auf meine Arbeit geworfen haben. Die Hilfe insbesondere in manchen Augenblicken, die im Augenmaß nicht auffielen, aber für das geübte Auge besser zu erkennen war, schätze ich sehr. Beide haben meinen Blick geschärft und mein Augenmerk auf Aspekte gelenkt, die mir nicht sofort ins Auge fielen, sodass ich das Wesentliche nicht aus den Augen verlor.

Im Auge behalten möchte ich auch alle Studienteilnehmer, auf die ich während der Studie mein wachsames Auge hielt. Im Auge habe ich zudem meinen Bürokollegen Florian Meier, M.Sc., der zeitgleich am DLR seine Masterarbeit verfasste und mir dabei Auge in Auge im Büro gegenübersaß. Ihm wünsche ich alles Gute für den Berufseinstieg!

Nicht zuletzt bin ich dankbar, dieses Thema ins Auge gefasst und die Umsetzung eines Online-Eye-Tracking-Algorithmus ins Blickfeld – und damit ein Stück weit ins Auge – gerückt zu haben.

Zusammenfassung

Diese Arbeit beschreibt die Entwicklung und Verifikation eines online Algorithmus zur Erkennung von Blickbewegungen, speziell für Hubschrauberpiloten. Vier Algorithmen wurden dabei mithilfe eines synthetischen Datensatzes optimiert, der aus einer Studie, die die Verfolgung eines Stimulus Punktes beinhaltete, generiert wurde. Für die Parameteroptimierung wurde eine neue Methodik genutzt, bei der die Algorithmusparameter durch den Abgleich von detektierten Blickbewegungen mit tatsächlichen Fixationen bewertet wurden. Zwei Algorithmen erwiesen sich als besonders effektiv und übertrafen den Tobii VT-Filter, sodass sie in die Online-Anwendung überführt wurden. Die Verifikation im AVES-Flugsimulator unter variierenden Sicht- und Arbeitsbelastungsbedingungen bestätigte eine zuverlässige, interpretierbare Blickbewegungsklassifikation bei tolerabler Latenz. Potenzial zur weiteren Optimierung wurde bezüglich der Generalisierbarkeit und Konfigurierbarkeit identifiziert.

Abstract

This thesis demonstrates the feasibility of online eye movement event detection tailored for helicopter pilots. Four algorithms (I-VT, I-DT, I-KF, and I-HMM) were optimized using a synthetic dataset from a fixation-saccade invocation task study, employing a novel pipeline combining event matching with a penalization strategy to optimize algorithm parameters. The I-VT and I-KF proved most effective, outperforming Tobii's VT filter, and were adapted for online use. Verification in the AVES flight simulator under varied visual and workload conditions confirmed reliable, interpretable classification with tolerable latency. Limitations in generalizability and configurability indicate areas for future refinement, including head-movement correction and improved usability.

Contents

List of Abbreviations

SA	Situational Awareness
HF	Human Factors
HP	Human Performance
DLR	German Aerospace Center
PCCR	Pupil Center Corneal Reflection
EOG	Electro-OculoGraphy
POG	Photo-OculoGraphy
VOG	Video-OculoGraphy
IR	Infrared
AOI	Area of Interest
VR	Virtual Reality
HCI	Human-Computer Interface
OTW	Outside The Window
EDA	Electrodermal Activity
ECG	Electrocardiogram
PDF	Probability Density Function
TP	True Positive
FN	False Negative
FP	False Positive

TN	True Negative
IoU	Intersection over Union
FOV	Field Of View
EM	Eye Movement
2PASD	Dual Pilot Active Sidestick Demonstrator
ITC	Inside The Cockpit
SL	Saccade Latency
SD	Saccade Duration
MFD	Modal Fixation Duration
ANF	Average Number of Fixations
AFD	Average Fixation Duration
$q_{\textbf{pos}}$	Position Process Noise
$q_{\textbf{vel}}$	Velocity Process Noise
N/A	Not Applicable
UTC	Coordinated Universal Time
AVES	Air Vehicle Simulator
ACT/FHS	Active Control Technology/Flying Helicopter Simulator
OLP	Optical Line Pair
TDP	Take-Off Decision Point
HIGE	Hover In Ground Effect
AHE	Above Helipad Elevation

MTE	Mission Task Elements
SAS	Stability Augmentation System
SAS 3AXES	Stability Augmentation System in pitch, roll, and yaw
ACpr SASy DIC	Attitude Control in pitch and roll, SAS only in yaw, and Direct Control for the collective
GVE	Good Visual Environment
DVE	Degraded Visual Environment
ISA	Instantaneous Self-Assessment
BWR	Bedford Workload Rating
I-VT	Identification by Velocity Threshold
I-TobiiVT	Identification by Tobii's Velocity Threshold
I-DT	Identification by Dispersion Threshold
I-KF	Identification by Kalman Filter
I-HMM	Identification by Hidden Markov Model
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
TCN	Temporal Convolutional Network

List of Figures

List of Tables

1. Introduction

Helicopter operations occur in diverse environments, where pilots may face high cognitive demands and time-critical decision-making tasks. Additional challenges may arise by factors such as adverse weather conditions, low visibility, and unexpected changes in terrain, making situational awareness (SA) and workload management critical for safe operations.[**ziv2020**]

The 2024 Annual Safety Review by EASA reports that over a five year period, around 6 % of all helicopter occurrences were linked to Human Factors (HF) and Human Performance (HP) issues, with 33 % of these related specifically to deficiencies in SA and sensory events.[**easa2024a**] While such figures underline the relevance of SA and workload to aviation safety, real world accidents demonstrate how insufficient SA and high workload can escalate into critical safety threats. For instance, a recently published final report on a Bell 206L-3 accident in South Africa mentions how the helicopter's main rotor blades struck tree branches while transitioning backward, which was a maneuver in the pilot's blind spot. The report attributed the accident to the pilot's attention being diverted to traffic before takeoff.[**sacaa2024**] Similarly, the Aerospatiale SA365N accident near a gas platform in the UK highlights how inadequate visual scanning and a lack of situational awareness can lead to spatial disorientation. In this case, challenging poor visibility contributed to the co-pilot's disorientation during the approach while the commander did not assist the co-pilot in managing the approach profile due to lack of cross-checking instruments, leading to the helicopter descending into the sea and resulting in multiple fatalities [**aaib2008**].

A promising approach to tackle these safety risks is by enhancing the analysis of pilot gaze behavior using eye-tracking technology [**galanda2024**]. Numerous studies on fixed-wing pilots have already demonstrated the potential benefits of eye-tracking technology [**lyu2023**]. These studies underline its potential benefits in the application of monitoring pilot attention and cognitive load [**schwerd2022**, **mohan2019**], detecting pilot fatigue [**naeeri2021**],and optimizing cockpit design as well as improving pilot-machine interfaces to enhance overall safety and efficiency [**li2018**]. Eye-tracking can contribute to further improvements in SA and workload management by refining pilots' eye-scanning techniques and increasing their awareness of gaze behavior [**rainieri2021**].

Although studies particularly targeting helicopter pilots are less extensive, interest in this research field has increased. The complex and dynamic nature of the helicopter cockpit presents unique challenges in both integrating eye-tracking systems and evaluating the collected data. Notable advancements in this area have been made by the German Aerospace Center (DLR), which successfully applied the Tobii Pro Glasses 3 eye-tracking system into the ACT/FHS helicopter and the AVES simulator. Simirlaly to fixed-wing aircrafts, these studies demonstrated that gaze behavior analysis through eye tracking can enable improvements in both pilot training and novel flight assistance systems.[**greiwe2023a**, **maibach2023**]

1.1. Problem Statement and Research Questions

To detect eye movements and thereby determine gaze behavior, eye-tracking algorithms are employed. The eye-tracking algorithm currently applied in helicopter pilot analysis is not specifically set up for helicopter environments [**greiwe2023a**], which can compromise the reliability of the algorithm's eye movement classifications. Moreover, the algorithm is limited to offline gaze data processing, lacking the capability for real-time or near-real-time eye movement classification.[**greiwe2023a**].

In addition, compared to fixed-wing aircraft, gaze data of helicopter pilots is scarce, making the gaze behavior analysis particularly challenging. This scarcity is further exacerbated by the versatility of helicopter flight missions and operational environments, where gaze pattern can differ. Consequently, a robust eye-tracking algorithm is crucial for effectively identifying eye movements of helicopter pilots to enable the accumulation of meaningful data on gaze behavior [**ziv2020**].

This thesis seeks to address this gap by developing an online eye movement event detection algorithm optimized for helicopter pilots. By tailoring the algorithm to this unique use case, the goal is to lay the groundwork for future applications, such as providing instant feedback on pilot workload and situational awareness or enabling adaptive pilot interfaces during flight and training.

This goal is guided by two research questions:

1. How can online eye-tracking algorithms be developed to effectively classify eye movement events of helicopter pilots?

2. To what degree can the feasibility of such algorithms be demonstrated in a preliminary

study in a full-flight simulator?

By answering these questions, the thesis provides both methodological contributions to the design of online event detection algorithms and deployable implementations inside helicopter flight simulators.

1.2. Thesis Overview

The procedure of this thesis consists of several key phases aimed at developing and verifying an eye-tracking algorithm tailored for helicopter pilots. The overall thesis structure is illustrated in Figure 1.1.

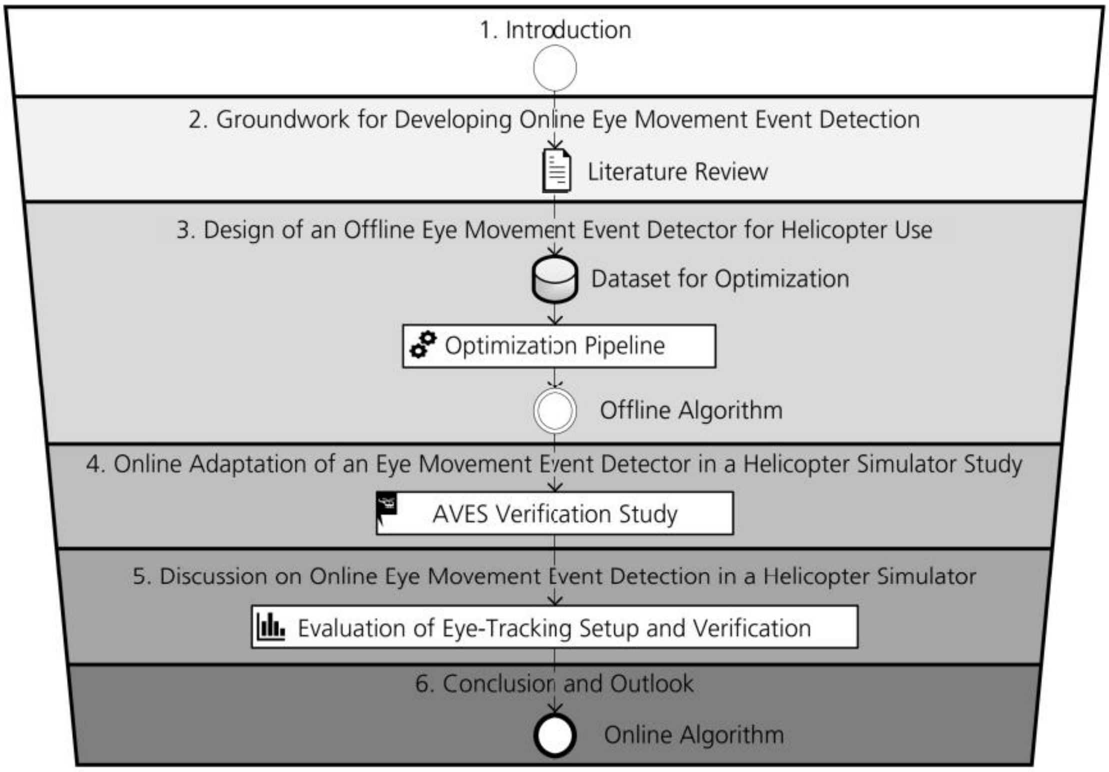

1. Introduction
2. Groundwork for Developing Online Eye Movement Event Detection
Literature Review
3. Design of an Offline Eye Movement Event Detector for Helicopter Use
Dataset for Optimization
Optimization Pipeline
Offline Algorithm
4. Online Adaptation of an Eye Movement Event Detector in a Helicopter Simulator Study
AVES Verification Study
5. Discussion on Online Eye Movement Event Detection in a Helicopter Simulator
Evaluation of Eye-Tracking Setup and Verification
6. Conclusion and Outlook
Online Algorithm

Figure 1.1.: **Thesis Overview.**

First, a comprehensive literature review establishes the foundation for algorithm development.

Based on this groundwork, viable event detection methods are selected and their parameters are optimized using a newly proposed optimization pipeline. After parameter optimization and validation on a test dataset, the most suitable methods for online deployment are identified through a comparative analysis with Tobii's VT filter, which is a well-established and widely used offline eye movement event detector [**olsen2012**].

The selected algorithms are subsequently adapted for online application and verified in the AVES flight simulator. The verification will evaluate the algorithm's performance in predefined flight phases through a dedicated preliminary study.

Algorithm performance is then evaluated using metrics such as detection accuracy, robustness to noise and missing gaze samples and real-time processing capability. Based on these results, the feasibility of the developed approach is discussed, addressing both methodological contributions and limitations of the eye-tracking setup and verification study.

Finally, the thesis concludes with a summary, obtained key findings and an outlook for future research and potential improvements of the online algorithms.

2. Groundwork for Developing Online Eye Movement Event Detection

Developing online algorithms for eye movement classification of helicopter pilots requires foundational knowledge ranging various subjects, from eye physiology to algorithm performance metrics. This chapter lays the groundwork by providing the necessary theoretical background on eye-tracking systems and eye movement event detection methodologies.

To build this foundation, the chapter first introduces an overview of commonly used eye tracker types, with a focus on the pupil center corneal reflection (PCCR) method and video based eye-tracking setups. Following this, key eye movement typologies are described. The next section explores how real-time event detectors are leveraged, especially to assess situational awareness and workload, in order to derive design requirements for the online algorithms. Next, key eye movement event detection approaches are presented and categorized into threshold-based, probability-based, and deep learning models. Finally, to assess the performance of the developed classification algorithms, suitable evaluation metrics are identified. These include event statistics evaluation, sample-level evaluation, and event-level evaluation, which provide performance measures to determine algorithm accuracy and reliability.

2.1. Overview of Common Eye Trackers

This section provides an overview of the primary eye-tracking techniques, with particular emphasis on the PCCR method and video-based setups, as the eye tracker used in this thesis is a video-based system employing the PCCR method.

Several underlying techniques in eye-tracking exist. One of the earliest methods is Electro-OculoGraphy (EOG), which capitalizes on the electrical potential differences of the skin surrounding the ocular cavity. By placing electrodes around the eye, EOG measures voltage changes as the eye moves, providing a relative indication of gaze direction. While this technique is non-invasive and allows for free head movement, its has lower spatial resolution than the other methods.[**duchowski2017**]

Contrary to EOG, the scleral contact lens/search coil method offers one of the most accurate techniques for measuring eye movements. This approach utilizes a special contact lens containing an embedded wire coil, wich is placed on the eye's cornea and sclera. When the eye moves within a surrounding electromagnetic field, the coil generates an electrical current proportional to its movement, which can be measured with high spatial and temporal resolution. However, this method is highly invasive, and requires a stationary head, which restricts its use to specialized research settings.[**duchowski2017**]

More common in modern research and commercial applications are optical methods, which are often categorized under Photo-OculoGraphy (POG) or Video-OculoGraphy (VOG). POG refers to optical tracking methods that use reflected light to measure eye position, while VOG specifically involves video based systems that capture images of the eye using camera combined with infrared illumination. These systems apply computer vision algorithms to track features like the pupil center or corneal reflections. VOG, in particular, offers a non-invasive, relatively easy to setup solution with good resolution and flexibility, but its performance can be affected by head movements or challenging lighting conditions.[**duchowski2017**]

2.1.1. Pupil Centre Corneal Reflection Eye-Tracking Method

Among video-based eye trackers, the PCCR method ist the most widely adopted technique today and forms the technological basis for the eye-tracking setup employed in this thesis.[**duchowski2017**, **martinezmarquez2021**]

The PCCR method uses an infrared (IR) light source to illuminate the eye, creating reflections, known as glints, on the cornea. Because infrared light is invisible to the human eye, these reflections do not distract the user. A high-speed infrared camera captures the pupil and glints, which serve as reference points for tracking eye movements. Gaze direction is then determined by analyzing the vector from the dark pupil center to the bright corneal reflections, as illustrated in Figure 2.1.[**duchowski2017**]

Since the cornea remains relatively stable while the eye rotates, any changes in this

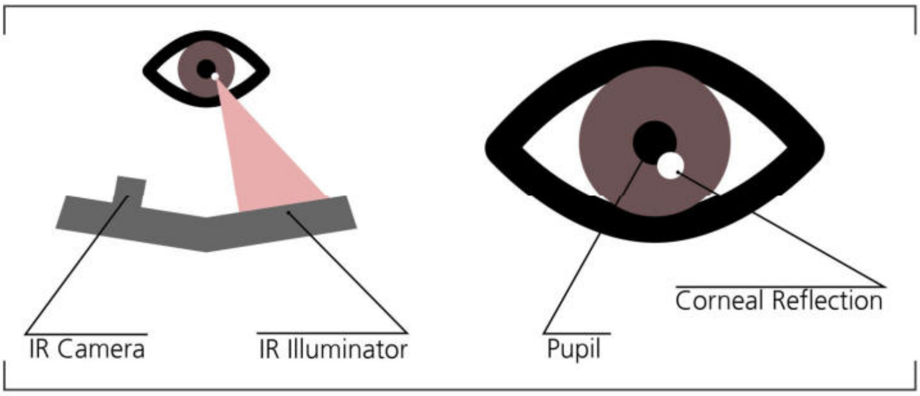


Figure 2.1.: **Schematic of the Pupil Center Corneal Reflection (PCCR) Technique, Including an Illustration of the Iris Center and Corneal Reflection based on srresearch2025.**

vector correspond to shifts in gaze direction. To improve accuracy, modern PCCR-based eye trackers, such as the Tobii system used in this thesis, construct a 3D model of the eye by analyzing multiple detected glints in combination with the known camera positions. This enables compensation for minor head movements and allows for more precise gaze direction estimation. Finally, the computed pupil–glint vector is mapped to gaze coordinates on a screen or within a 3D space, enabling real-time retrieval of gaze data. In the case of 3D gaze coordinates, these are determined by the position of the vergence point of the left and right gaze vectors relative to the eye tracker's scene camera.[**tobii2023**, **tobii2023a**]

Due to individual differences in corneal curvature and eye anatomy, the PCCR method requires a calibration process. During calibration, the system determines how the pupil-glint vector corresponds to specific gaze points by having the user fixate on predefined targets [**hansen2010**]. Although calibration ensures reliable gaze estimation, several adverse factors remain. One such factor is changes in pupil size, which may occur due to lighting conditions, fatigue, or cognitive load. These changes can slightly alter the detected pupil center position and potentially reduce measurement precision. Modern eye-tracking systems attempt to mitigate this issue by continuously adjusting their algorithms or by requiring recalibration.[**tobii2023b**, **hooge2024**]

Head movement is another important factor that can affect accuracy, particularly in single-camera systems. While modern PCCR-based eye trackers often employ multiple IR light sources and stereo cameras to compensate for motion, excessive head movement can still introduce errors. Furthermore, transparent surfaces such as glasses or contact lenses may cause unwanted reflections or distort the corneal reflection, thereby interfering with gaze estimation.[**hansen2010**]

2.1.2. Video based Eye-Tracking Setups

Video based eye-tracking setups are typically categorized into remote eye trackers with a fixed chin rest, remote eye trackers with free viewing, and head-mounted eye trackers.[**valtakari2021**]

The eye-tracking setup that includes a remote eye tracker with a fixed chin rest while viewing a display is depicted in figure 2.2 a. In this configuration, the participant's head is stabilized using a chin rest while they view a display, such as a computer monitor or tablet. A remote eye tracker positioned above or below the display records their eye movements. This setup offers high accuracy, as the chin rest minimizes head movement, leading to more precise gaze measurements. The setup also ensures that visual stimuli are presented consistently across participants, making this method particularly useful for studies that require standardized conditions as well for matching gaze points to Areas of Interests (AOIs) [**holleman2020**]. The controlled environment, while beneficial for experimental consistency, may not accurately reflect real-world viewing conditions and the prolonged use of a chin rest can cause discomfort, which makes long-duration studies challenging for participants [**valtakari2021**].

Another commonly used eye-tracking setup depicted in figure 2.2 b is the remote eye tracker with limited allowable head movement within a “head box”. The remote eye tracker continuously records their gaze movements without physical head restraints [**klein2019**]. Unlike the chin rest setup, this approach provides a more natural experience for participants, so that it is better suited for viewing tasks that aim to capture natural viewing behaviors. However, the improvement in ecological validity, which refers to the extent to which recorded gaze behavior represents real-world behavior, is only moderate, as gaze patterns tend to remain similar across both setups.[**backhaus2024**] Additionally, head movements introduce variability in the data, leading to slightly reduced accuracy compared to a fixed-head setup [**duchowski2017**].

The least restriction on head movement is set for the head-mounted eye tracker as illustrated in 2.2 c, also known as mobile eye tracking. In this setup, the eye tracker is embedded in a wearable device, wich are either glasses or a headset, allowing the participant to move freely while their eye movements are recorded. The head mounted eye tracker is particularly valuable for studying real-world behaviors, as it captures gaze patterns in dynamic and natural environments. The ability to track gaze while participants move through different settings enhances the ecological validity of experiments. The disadvantages of the head-mounted eye trackers are that they tend to have less temporal accuracy and lower spatial accuracy compared to fixed remote setups, as head and body movements and even minor eye tracker slippage can introduce noise into the data [**nierhorster2020**]. Additionally, some participants may find head-mounted trackers less comfortable, especially during extended sessions and the data collected from mobile setups can be more complex to process due to varying distances and viewing angles.[**franchak2020**]. Furthermore, in head-mounted eye tracking the gaze coordinates are defined relative to the head rather than to a fixed point in the surrounding environment. This makes it difficult to distinguish between eye and head movements solely by examining the gaze data [**franchak2020**]. Additionally, unlike a fixed coordinate system, coordinates for areas of interest (AOIs) shift whenever the head moves. This complicates automatic AOI assignment, which is why AOIs are mostly labeled manually in head-mounted eye-tracking setups [**benjamins2018**].

Another emerging approach for using head-mounted eye tracker is the integration into virtual reality (VR). In this setup, the eye-tracking system is embedded within a VR headset, allowing researchers to monitor gaze patterns within an immersive digital environment. This method enables precise control over visual stimuli while maintaining a high level of interactivity. The immersive nature of VR provides a unique setup to study eye movements during simulated real-world tasks and therefore enhances the realism of experiments. However, the integration of eye tracking into VR comes with the same technical challenges mentioned before. Besides, the VR headsets can also lead to discomfort for some users, particularly if they experience motion sickness.[**adhanom2023**]

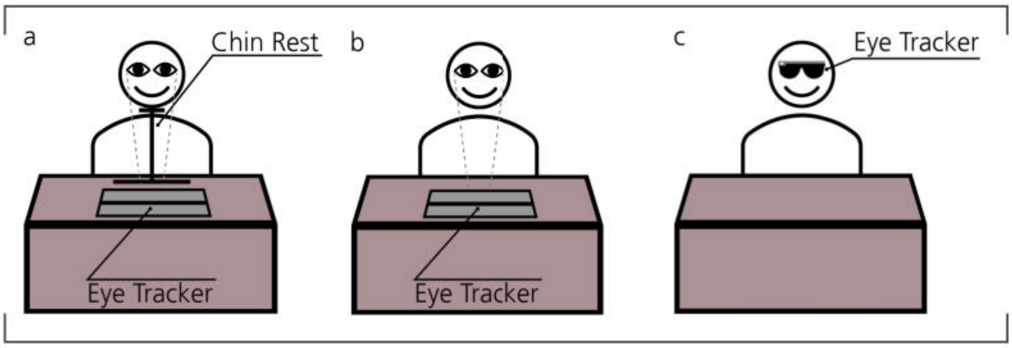


Figure 2.2.: **Video-Based Eye-Tracking Setups: (a) Head-Restricted with Chin Rest, (b) Head-Boxed, and (c) Head-Free adapted from Valtakari et al. [valtakari2021].**

2.2. Eye Movement Typologies in Eye Tracking

Eye-tracking technologies rely on understanding the anatomy and movement behavior of the eye. Key anatomical aspects are depicted in Figure 2.3 and include the cornea, iris, pupil, retina. Furthermore, the shape of the eye ball is maintained and primarily protected by the sclera, which is the tough, white outer layer of the eye that provides structure and protection, while the optic nerve is responsible for the transmittance of visual information from the retina to the brain.[**holmqvist2017**]

When light enters the eye, it first passes through the cornea, a transparent, strongly curved outer layer situated in front of the pupil. It can reflect infrared light, which is used in the PCCR method to determine the position of the eye and the gaze direction. After passing through the cornea, light enters the pupil, a circular aperture controlled by the iris, which is a colored, muscular ring that adjusts the amount of light entering the eye. The light then reaches the light-sensitive layer at the back of the eye called retina. The retina contains photoreceptors called rods and cones that convert light into electrical signals. Rods are specialized for low-light and grayscale vision, while cones detect color and fine details in bright conditions. The central region of the retina is defined as the fovea. It is densely packed with cones and is responsible for sharp, detailed vision. Eye-tracking focuses on identifying the fovea's direction, as it aligns with the point of highest visual interest. Finally, the retina's electrical signals are transmitted via the optic nerve to the visual cortex, where images are processed and perceived.[**holmqvist2017**]

Eye-trackers often detect the pupil's center to track eye movements, and changes in pupil size can reveal information about attention, cognitive load, or emotional states.[**holmqvist2017**]

For eye-tracking purposes, the visual angle is defined and describes how large an object appears to the eye, based on its size and viewing distance. Consequently, rather than representing the object's actual physical size, it reflects its apparent size from the observer's perspective and is typically measured in degrees.[**duchowski2017**]

The visual angle also helps quantify how visual resolution decreases with increasing distance from the fovea. To delineate the visual field, a differentiation between foveal vision, which offers high-resolution central vision, and peripheral vision, which is characterized by lower resolution, is made. In visual-cognition research, central vision is typically defined as spanning 0° to 5° of eccentricity, with regions beyond 5° classified as peripheral vision. While the fovea is specialized for perceiving fine detail and stationary objects, peripheral vision is more sensitive to motion and changes in the environment. Consequently, peripheral vision helps in detecting and locating potential targets, whereas

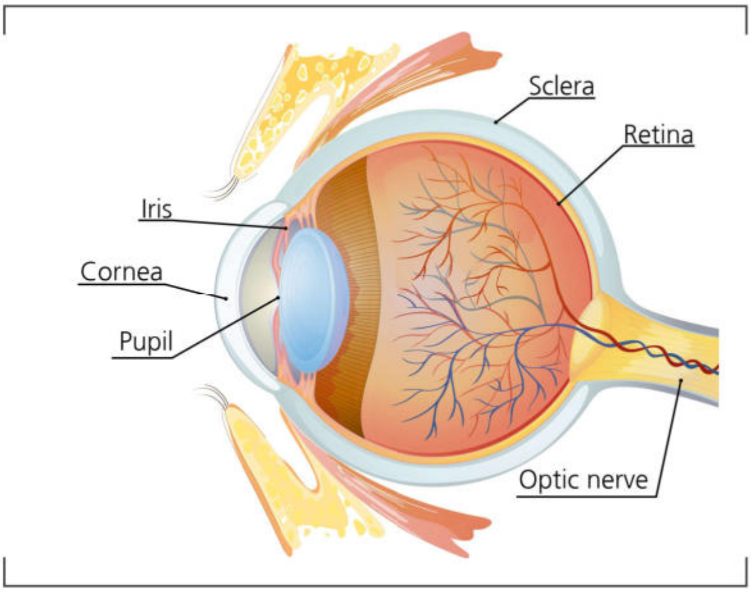


Figure 2.3.: **Eye Anatomy based on duchowski2017.**

foveal vision is responsible for identifying and verifying them.[**klein2019**]

Another important measurement in eye-tracking is the angular velocity, which describes the speed of eye movements and is typically measured in degrees per second (°/s). The angular velocity is derived from the amplitude and frequency. Here, amplitude refers to the angular distance the eye moves across the visual field, measured in degrees of visual angle, while frequency corresponds to the inverse of the movement duration and is often defined by the eye tracker's sampling rate. Frequency is used instead of time duration because manufacturers typically specify the sampling rate of their devices, making it straightforward to calculate the velocity of eye movements between samples.[**duchowski2017**]

Eye tracking focuses on two primary types of eye movements: fixations and saccades. In eye movement analysis, these movements provide evidence of voluntary, overt visual attention and are explored in greater detail in the following sections.[**duchowski2017**]

2.2.1. Fixations

Fixations are stable gaze positions where visual attention is primarily directed, during which visual information from the point of focus is extracted and processed [**klein2019**].

Although visual attention is primarily directed at fixations, it does not always align with it. In cases of covert attentional shifts, attention can shift toward a peripheral stimulus before the fixation point changes, which is in contrast to overt attentional shifts where gaze and attention move together. [**zhaoping2023**].

The average duration of a fixation is around 200 to 300 ms, though this can vary widely and can be much shorter or longer depending on the context [**mahanama2022**]. Although the term "fixation" might implicate a stationary eye focus on a object of interest, it consists of three fixational movements called tremor, drift, and microsaccades. These movements are crucial for maintaining visual perception, as vision would otherwise fade in the absence of continuous retinal stimulation.[**klein2019**]

2.2.2. Saccades

Saccades are rapid and ballistic eye movements whose trajectory cannot be altered once initiated and are typically performed 3–5 times per second. They are triggered by the desire to focus on a specific target and function to reposition the eyes from one fixation point to another, aligning the high-resolution fovea with objects of interest for detailed visual inspection.[**duchowski2017**, **mahanama2022**]

Saccades occur frequently, with an average rate of two to three per second and are typically short, covering typically a visual angle of 3 to 80 degrees of visual angle [**collewijn1988**]. To prevent visual blurring caused by their high-speed motion, vision is briefly suppressed during saccades in a phenomenon known as saccadic suppression. This suppression ensures that rapid eye movements do not interfere with the perception of a stable visual environment [**thiele2002**].

An example of a saccadic eye movement is illustrated in Figure 2.4. Initially, gaze is focused on a central point, represented by a dot in the top square. At a specific time a peripheral stimulus indicated by a dot appears, which prompts a shift of gaze toward it. The temporal dynamics of this saccadic eye movement are displayed and shows changes in position and velocity over time. From the figure, it is evident that the gaze remains steady until approximately 250 ms. At this point, the gaze shifts toward the peripheral stimulus, marked by a steep change in position. The total change in position is the distance between the central fixation and the peripheral target and is defined as the saccade amplitude. After the saccade, the position stabilizes into a fixation on the peripheral target. The lower plot in the figure shows the velocity of eye movements. During the initial latency period, before the saccade begins, the velocity remains close to zero. The same latency of approximately 250 ms of the saccade is observed between the onset of the peripheral stimulus and the initiation of the saccade. The latency occurs due to the time required for saccade programming and typically ranges from 100 to 300 ms [**fischer1993**]. Once the saccade begins, the eye movement rapidly accelerates, reaching its peak velocity shortly afterward. This peak velocity corresponds to the rapid gaze shift toward the target and is typically around 125 to 500 degrees per second [**collewijn1988**]. Following the peak, the velocity decreases to zero as the gaze reaches and fixates on the peripheral target. For the remainder of the trial, the gaze remains steady on the target, as indicated by the flat velocity curve and stabilized position.

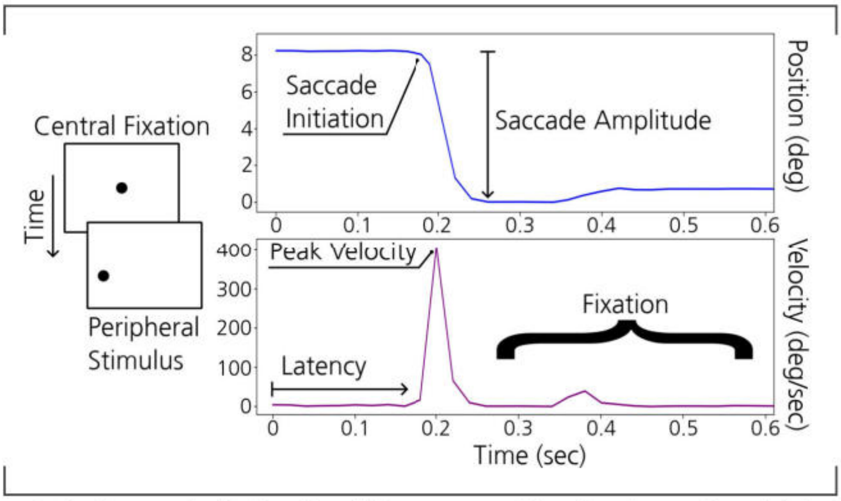


Figure 2.4.: **Saccade Behavioral Response Metrics based on klein2019.**

Furthermore, saccades are not only performed under volitional control, as in the described example, but can also be triggered reflexively by visual cues [**fischer1993**]. Reflexive saccades are driven by salient inputs such as brightness, motion, contrast, or novelty that automatically capture attention, whereas volitional saccades are guided by cognitive processes, including task demands, instructions, or specific goals that shape saccadic responses [**pomplun2006**]. In real-world scenarios, these two mechanisms often interact. For instance, in visual search tasks, saccade patterns are influenced both by the saliency of the stimulus and by its task relevance, reflecting the interplay between reflexive and volitional saccades [**mcpeek2000**].

2.2.3. Limitations and Challenges in Eye Movement Tracking

While fixations and saccades are primary eye movement types classified in most identification algorithms, additional factors must be considered to enhance analysis of the gaze data produced by the algorithms. These include not only other eye movement types, but also the presence of noise in gaze data, which can impact algorithm performance.[**klein2019**]

Noise in eye-tracking data originates from multiple sources, including blinks, drifts, data loss, physiological artifacts and measurement inaccuracies inherent to eye-tracking devices.[**agtzidis2019**] Environmental factors, such as vibrations affecting the eye-tracking camera or infrared interference from sunlight, can introduce distortions. Similarly, hardware limitations, including inaccuracies in gaze measurement, obscurity due to head movements and slippage of head-mounted trackers caused by facial motion, further contribute to data inconsistencies. [**klein2019**] Ensuring high-quality gaze data is crucial, as poor or inconsistent measurements can lead to misinterpretation of gaze behavior. Unequal data quality between participants can further impact research validity.[**wass2014**]

In addition to noise-related challenges, other eye movement types, such as smooth pursuits and physiological nystagmus, are present in the gaze data but are not classified by the event detection algorithms considered in this thesis. The presence of these unclassified movements can adversely affect the detection accuracy of the algorithms. Understanding these eye movements is therefore crucial for recognizing instances of low detection quality.[**andersson2016**]

Smooth pursuit refers to the eye's ability to visually track a moving target. While it shares similarities with fixation, in which both cases the visual attention is directed at a stimulus, the key difference is that in smooth pursuit the stimulus is moving. Similarly, unlike saccades, which involve abrupt shifts in visual focus, smooth pursuit maintains continuous fixation on the moving target. As illustrated in Figure 2.5, smooth pursuit is triggered by the movement of a target stimulus. In both the upper and lower diagrams, the dashed line represents the target's path and velocity, respectively, showing a fixed initial position followed by a constant-velocity motion after target motion onset. The solid line represents the eye's corresponding position and velocity. Initially, the eyes remain stationary. Once the target begins to move, the eyes first follow its direction. After matching the target's velocity, the eye then executes a saccadic movement to correct any residual difference between its position and the target's position. In addition to this smooth pursuit performed solely with eye movement, a smooth pursuit can also be performed by a coordinated movement of the head.[**lisberger2010**, **duchowski2017**]

Physiological nystagmus is divided into vestibular- and optokinetic nystagmus. Vestibular nystagmus occurs when the vestibular system, which plays a key role in maintaining balance and spatial orientation, is disrupted. Located in the inner ear, this system helps stabilize gaze during head movements. In contrast, optokinetic nystagmus is a reflexive

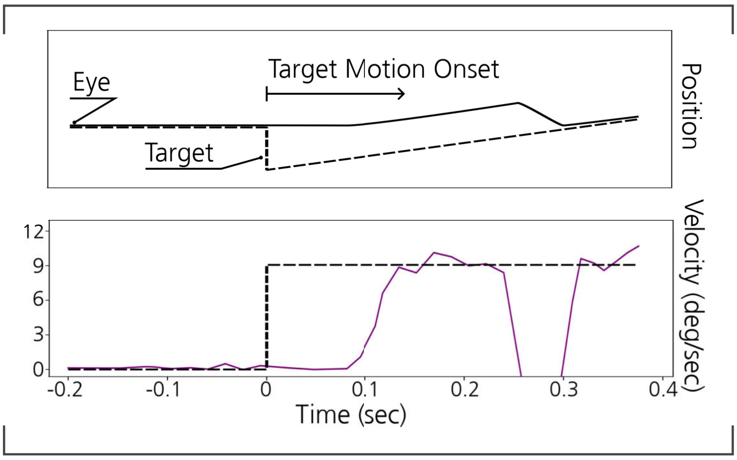


Figure 2.5.: **Smooth Pursuit Behavioral Response Metrics based on lisberger2010.**

eye movement that helps stabilize retinal images when the eyes track a moving visual scene. It alternates between smooth pursuit and rapid saccades that reset the gaze to its original position.[**klein2019**]

2.3. Applicability of Real-Time Eye Movement Event Detectors

To design and implement an online algorithm capable of classifying eye movement events, it is crucial to first understand the design requirements for such a filter in (near) real-time applications. These requirements include factors such as acceptable latency, relevant input metrics and the type of eye-tracking data necessary to support downstream use cases, such as situational awareness- and workload monitoring and pilot-machine interaction.

This chapter investigates the application of real-time eye movement event detectors focusing first on their role in human-machine interfaces and subsequently on their utility for assessing situational awareness and workload. While eye movement related metrics such as blink rate and pupil dilation can also be tracked in real time [**zagermann2016**], this chapter focuses exclusively on fixation and saccade eye movements as these are most relevant for the development of an online eye movement classification algorithm, which is the ultimate goal of this thesis.

2.3.1. Human-Computer-Interface

Real-time eye movement event detectors are employed in human-computer interface (HCI) applications that require rapid gaze-based input. As illustrated in Figure 2.6, these applications can be mainly categorized into two types: (a) gaze-driven input interfaces and (b) feedback interfaces that respond to user gaze behavior.

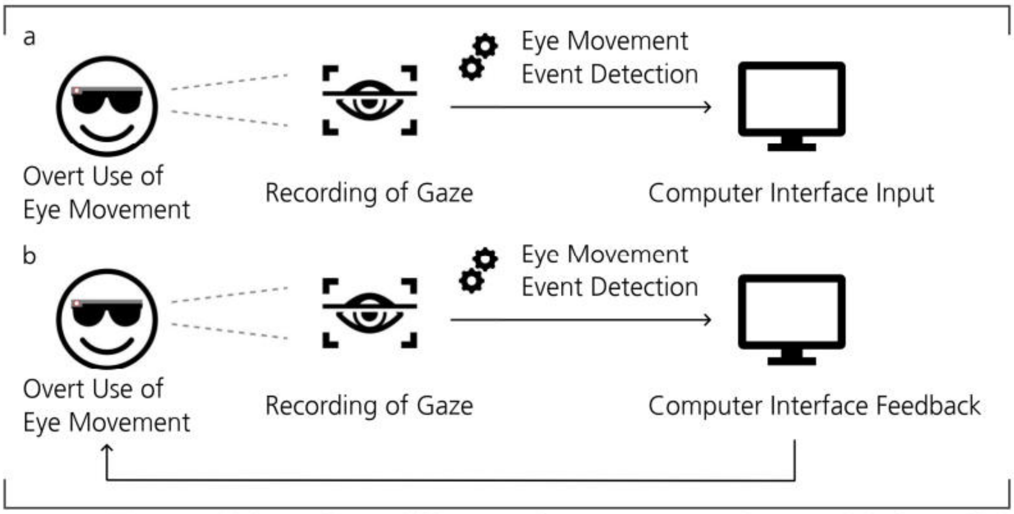

a
Overt Use of Eye Movement
Recording of Gaze
Eye Movement Event Detection
Computer Interface Input
b
Overt Use of Eye Movement
Recording of Gaze
Eye Movement Event Detection
Computer Interface Feedback

Figure 2.6.: **Types of Gaze-Based Human-Computer Interfaces: (a) Gaze-Driven Input Interfaces and (b) Feedback Interfaces Responding to User Gaze Behavior.**

The gaze-driven input control shown in Figure 2.6 (a) refers to interfaces where real-time eye movement event detectors allow users to issue commands directly with their gaze. A common application is controlling a computer cursor solely via eye movements, with target selection based on gaze pointing or blinks [**biswas2016**]. For instance, **hariharan2024** developed such a human-computer interface as an assistive technology for individuals who cannot operate a mouse, achieving a total target selection latency of less than 100 ms. In an aviation context, **murthy2020** designed a real-time gaze-based pointing system for cockpit environments to reduce cognitive load. They reported average pointing and selection times of under 2 seconds, noting that the system required a minimum latency of 200 ms due to the need for raw gaze data smoothing.

The next class of interfaces are real-time feedback systems (see Figure 2.6 (b)), where eye movement events dynamically trigger context-sensitive responses. For example, **schwerd2024** developed a system that continuously tracks pilot gaze behavior, specifically fixation counts and durations on defined AOIs, to assess situational awareness.

If a pilot fails to attend to critical information, the system automatically issues visual or auditory alerts to redirect attention to unnoticed system state changes. In a flight simulator study, these adaptive alerts significantly reduced detection times of critical changes and pilot task performance, although some participants expressed concerns about the transparency of the alert logic.

These real-time applications leverage fast classification of eye movement events, primarily fixations, to dynamically adjust system behavior or user interface states. As these systems aim to enhance situational awareness and reduce cognitive workload, the following subsection explores these two concepts in greater detail.

2.3.2. Situational Awareness and Workload Monitoring

Real-time eye movement event detection can be leveraged to improve SA in safety-critical scenarios, such as in hazard identification, error detection, and activity monitoring [**martinezmarquez2021**]. Situational awareness it self is defined by **endsley1988** refers to "the perception of the elements in the environmentwithin a volume of time and space, the comprehension of their meaning and the projectionof their status in the near future".

In aviation, a pilot's SA and level of expertise can be inferred from the distribution and duration of fixations on relevant AOIs [**li2019**]. Research has shown that experienced pilots more frequently perform express (short) fixations compared to novices [**ziv2016**, **greiwe2022**]. For example, during cockpit scanning, expert pilots tend to fixate more efficiently on instruments without excessive dwell time, reflecting greater SA and operational skill [**ziv2016**]. Conversely, novice pilots rely more heavily on Outside The Window (OTW) cues and exhibit less effective scanning behavior [**greiwe2023a**]. Real-time detection of such fixation patterns can support the assessment of SA during flight or be applied in pilot training [**greiwe2022**].

Real-time fixation analysis is also used to estimate and mitigate cognitive workload. In aviation, workload is defined as "the integrated mental and physical effort required to satisfy the perceived demands of a specified flight task" [**roscoe1987**]. Researchers have found that fixation durations increase under high workload conditions and during immediate hazard detection, reflecting heightened visual demands [**velichkovsky2002**, **marquart2015**]. Conversely, a higher cognitive load results in lower fixation rates [**zagermann2016**], highlighting a clear correlation between fixation metrics and cognitive workload.

In a recent study, **liu2024** employed real-time eye-tracking data to estimate workload within a 10 second time window. Their system demonstrated the capability to detect periods of high workload in near real-time, with a processing latency of less than 0.1 s, making it applicable for adaptive cockpit designs. For instance, such systems might automatically dim non-essential displays or recommend breaks during overload conditions.

In addition to eye movement metrics, other measures can be incorporated to complement them and further enhance workload assessment, as recently demonstrated by **walocha2025**. Their tool integrated a range of physiological and behavioral measures, including pupil diameter with electrodermal activity (EDA) and electrocardiogram (ECG) signals, capturing skin conductance and cardiovascular responses, respectively. These measurements were also analyzed within a 10 second time window, highlighting their potential for future online applications in adaptive user interfaces.

Finally, although temporal windows around 10 s are common, no universally accepted standard exists, and longer windows generally tend to improve classification accuracy [**tervonen2021**].

2.4. Exploration of State-of-the-Art Eye Movement Event Detectors

Eye movement identification involves a variety of approaches, including threshold-based algorithms, probability-based models, and advanced deep learning techniques. Each approach has its own strengths and challenges, making their examination essential for selecting suitable event detection techniques in a helicopter use case. As the previous section showed, human–computer interfaces can be effectively implemented using only fixation metrics. Accordingly, the detection methods examined in this thesis solely identify fixations and saccades, consistent with common event detection approaches [**salvucci2000**].

2.4.1. Threshold-Based Algorithms

Threshold-based algorithms utilize predefined thresholds based on velocity or dispersion to classify eye movements from gaze data. Two widely recognized algorithms in this domain are the Velocity-Threshold Identification (I-VT) algorithm and the Dispersion-Threshold Identification (I-DT) algorithm.[**salvucci2000**]

The I-VT algorithm classifies gaze points as fixations or saccades by comparing their angular velocity to a predefined threshold. Gaze points with velocity below the threshold are labeled as fixations, while higher velocities indicate saccades [**salvucci2000**]. The underlying assumption of this filter is that during fixations, eye movements are slow, while during saccades, eye velocity is high [**duchowski2017**]. A critical aspect of the I-VT algorithm is the selection of an appropriate velocity threshold. Figure 2.7 illustrates this relationship, which shows the angular eye movement velocity over time in the top plot, while the lower plot displays the resulting classifications. If the threshold is set too low, noise may be misclassified as saccades, whereas too high it suppresses noise but risks missing genuine saccades. As the figure demonstrates, the threshold directly influences the accuracy of the event detections derived from raw gaze data [**kosel2023**].

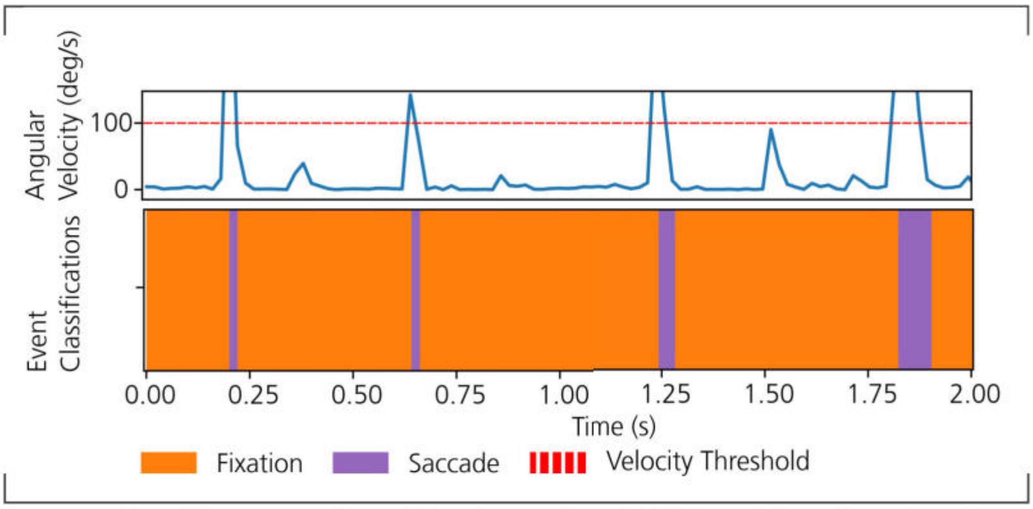


Figure 2.7.: **Eye Movement Classifications of an I-VT Algorithm Showing Angular Eye Movement Velocity Over Time in the Top Plot an the Resulting Classifications in the Lower Plot.**

In contrast to the I-VT algorithm, the I-DT algorithm identifies fixations by evaluating the spatial dispersion of consecutive gaze points. The I-DT approach assumes that fixation points are spatially clustered, as illustrated in Figure 2.8. Hereby raw gaze samples are shown as black dots, while detected fixations appear as red dots and the instructed target fixation locations as gray squares. The algorithm groups consecutive gaze points that fall within a predefined dispersion threshold as a single fixation. In addition to this spatial criterion, a minimum fixation duration threshold is applied to filter out noise and measurement variability. As seen in Figure 2.8, both the dispersion threshold and the fixation duration threshold strongly influence which gaze samples are ultimately classified as fixations [**salvucci2000**].

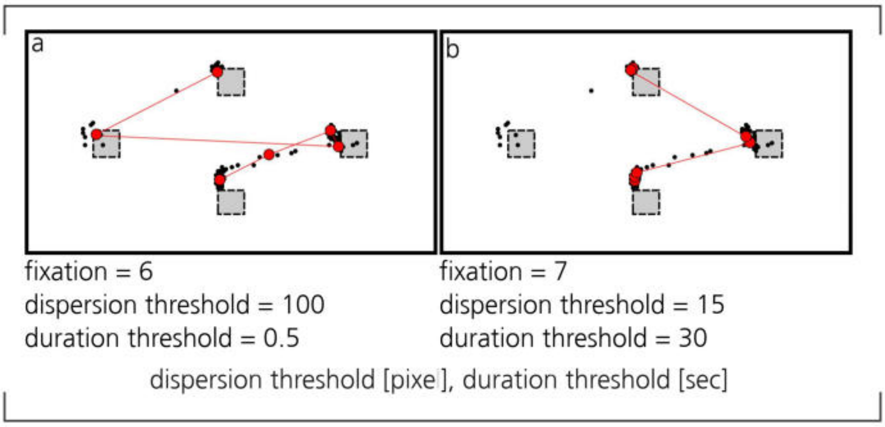


Figure 2.8.: **Influence of Dispersion and Duration Thresholds in the I-DT Algorithm based on yoo2021.**

2.4.2. Probability-Based Algorithms

Probability-based algorithms use statistical models to classify eye movements by estimating the likelihood that a given movement belongs to a particular class, rather than applying strict thresholds. Two widely used approaches are Hidden Markov Models (I-HMMs) and the Kalman Filter (I-KF).

Figure 2.9 illustrates a two-state I-HMM for eye movement detection. The model consists of four key components called hidden states, observations, transition probabilities, and observation probabilities. The hidden states represent the system's unobservable conditions, which are fixation and saccade, while the observations are measurable variables, such as angular velocity and visual angle, that describe eye movements.[**salvucci2000**]

The model links states and observations through probabilities. Transition probabilities determine the chance of moving from one hidden state to another, such as shifting from a fixation to a saccade, which is depicted in the figure by the arrows connecting the two states. On the other hand, emission probabilities define the likelihood of a specific measurement occurring within a certain hidden state. For example, a low angular velocity is highly probable during a fixation state. This relationship is depicted in the figure by a probability density function (PDF) over velocity. These probabilities are typically estimated through training, which adjusts model parameters to best fit the observed data. Once trained, the HMM infers the most likely sequence of hidden states from the observed data.[**salvucci2000**]

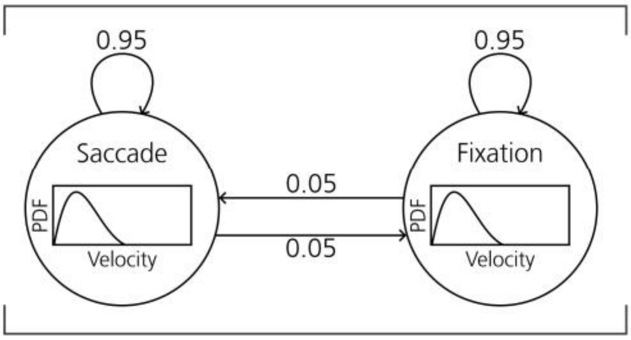


Figure 2.9.: **Schematic Depiction of a Two-State Hidden Markov Model based on salvucci2000.**

The I-KF operates similarly, maintaining a system state that includes position and velocity, which is both predicted and corrected over time.

At each time step, the filter predicts the next state from the previous one using the motion model, then corrects this prediction based on the actual measurement as depicted in Figure 2.10. This recursive process produces smoothed state estimates. Following correction, the discrepancy between the corrected prediction and the actual measurement is quantified as a chi-squared error and compared to a predefined threshold. If the error falls below this threshold, the sample is classified as a fixation and otherwise it is labeled as a saccade. The chi-square threshold is therefore critical as lower values result in stricter fixation detection, while higher values allow more variability within fixations.[**sauter1991**, **komogortsev2007**].

2.4.3. Deep Learning Models

Several types of deep learning models are commonly used for eye movement identification. Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) are particularly well-suited for sequential data like eye-tracking signals, as they can consider temporal dependencies when classifying each sample [**seth2025**]. In terms of real-time applicability, CNNs have shown promising results, especially Temporal Convolutional Networks (TCNs) **elmadjian2021.**

Hereby, TCNs require structured input, so that instead of treating each gaze point in-

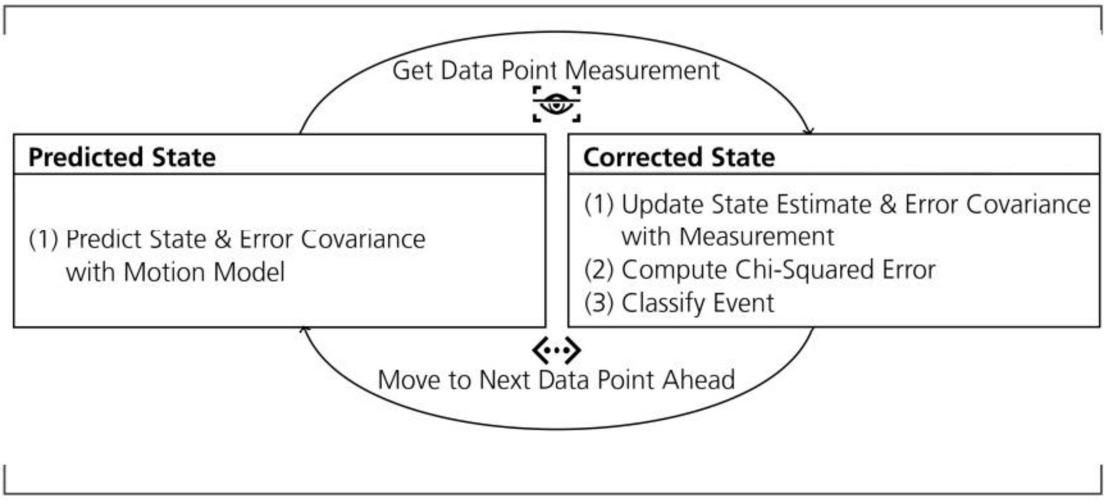

Get Data Point Measurement
Predicted State
(1) Predict State & Error Covariance with Motion Model
Corrected State
(1) Update State Estimate & Error Covariance with Measurement
(2) Compute Chi-Squared Error
(3) Classify Event
Move to Next Data Point Ahead

Figure 2.10.: **Time- and Measurement Update Process in a Kalman Filter based on welch2006.**

dependently, a sliding window is applied to group gaze points over a short duration, capturing temporal dependencies. Thereby, a TCN consists of stacked 1D convolutional layers applied along the temporal dimension, as illustrated in Figure 2.11. Each convolutional layer applies a set of learnable kernels across the input sequence that contain gaze position and velocity to extract local temporal patterns. The kernel size determines how many consecutive points are processed at each step. In the figure, it is represented by the number of connections from the previous layer that feed into a single node in the current layer. Besides, the stride specifies the step size for moving the kernel, while dilated convolutions introduce gaps between the kernel's elements. By using stride, dilations D, and stacking multiple hidden convolutional layers, the network efficiently captures both short- and long-range temporal dependencies, as demonstrated in the figure. After feature extraction by the convolutional layers, the output is converted into probabilities for multi-class classification, where the predicted eye movement class corresponds to the highest probability.[**elmadjian2021**, **bai2018**]

2.5. Suitable Metrics for Eye Movement Event Detection

Evaluating the performance of eye movement identification algorithms requires the use of appropriate evaluation metrics. Several evaluation methods already exist, with some

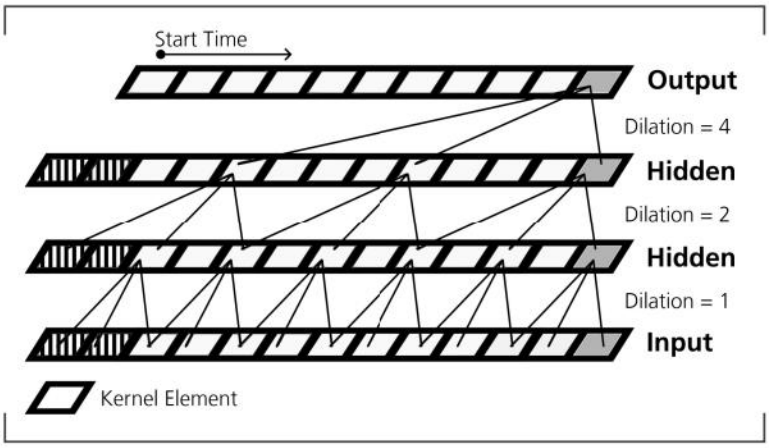


Figure 2.11.: **Schematic of a Dilated Causal Convolution (Dilation Factors 1, 2, and 4; Kernel Size 3), based on bai2018.**

serving as key metrics in the field of eye tracking. This chapter introduces these metrics to assess both the effectiveness and applicability of the algorithms.

The presented metrics are categorized based on the depth of analysis they provide. The evaluation begins with general event statistics, followed by a sample-level evaluation, which examines time-aligned data points for a more detailed comparison, and an event-level evaluation, enabling a precise assessment of the algorithm's accuracy and reliability in classifying eye movements.[**startsev2022**]

2.5.1. Event Statistics Evaluation

Evaluating the performance of algorithms with event statistical values fundamentally relies on a comprehensive understanding of the various eye movement types, as detailed in Chapter 2.2. By leveraging this knowledge, general statements can be formulated and compared to the labeled data produced by the algorithm.

For instance, measurable parameters include mean fixation duration and saccade rate typically ranging from 200–300 ms and 3–5 per second, respectively (see 2.2.1 and 2.2.2). Distributions that deviate significantly from established norms may indicate inaccuracies in event detection [**startsev2022**].

As noted by **startsev2022**, while such methods can indicate whether an algorithm yields reasonable results, they do not assess the precision and accuracy of event detection. Nonetheless, this approach offers a straightforward preliminary evaluation without necessitating a separate dataset for comparison.

2.5.2. Sample-Level Evaluation

The sample-level evaluation involves comparing the detected eye movements generated by the algorithm with a ground truth. A ground truth refers to a dataset, which time samples are annotated with an corresponding eye movement event. The primary requirement for this method is the trustworthiness of the labeled data. Commonly, ground truth datasets are hand-labeled by multiple human coders and reviewed by experts to ensure robustness and reliability.[**agtzidis2016intp**] However, human biases and varying interpretations of eye movement categories mean that these datasets are not infallible, though they are still considered to be the gold standard for validating algorithms.[**andersson2016**, **hooge2017**]. Thus, ground truths that are accurately labeled can serve as a benchmark, against which the algorithm's output can be compared.

In this comparison, time-aligned gaze points are directly assessed to determine whether the labeled gaze point matches the corresponding ground truth gaze point [**startsev2022**]. The comparative results can be summarized in a confusion matrix to provide a structured view of the classifier's performance. A typical confusion matrix for a binary eye movement classification problem distinguishing between fixations and saccades, based on **goutte2005**, is shown below:

Actual\Predicted	Fixation	Saccade
Fixation	TP	FN
Saccade	FP	TN

This matrix distinguishes four possible outcomes when evaluating classification performance. True Positives (TP) refer to fixation instances that are correctly classified as fixations. False Negatives (FN) arise when fixation instances are incorrectly classified as saccades. Conversely, False Positives (FP) denote saccade instances that are mistakenly classified as fixations. Finally, True Negatives (TN) correspond to saccade instances that are correctly identified as saccades.

By categorizing the outcomes of classification into TP, TN, FP, and FN, several performance metrics can be derived. One of them is the F1-score, which balances both precision and sensitivity:[**manning2008**]

$$\text{F1-score} = \frac{2TP}{2TN + FP + FN}$$

In addition, the most widely used metric for evaluating event detection algorithms is the Cohen's kappa [**startsev2022**]. Its computation relies on two quantities. The observed agreement p_o measures the proportion of correctly classified instances:[**cohen1960**]

$$p_o = \frac{TP + TN}{TP + TN + FP + FN}$$

The expected agreement p_e is derived from the marginal distributions of predicted and actual classes. With the total number of instances defined as $N = TP + TN + FP + FN$, it is calculated as:[**cohen1960**]

$$p_e = \left(\frac{(TP + FN)(TP + FP)}{N^2}\right) + \left(\frac{(FP + TN)(FN + TN)}{N^2}\right)$$

Finally, Cohen's kappa is computed as:[**cohen1960**]

$$\text{Cohen's kappa} = \frac{p_o - p_e}{1 - p_e}$$

While the F1-score is a popular metric that offers a single metric that balances precision and recall, it can overestimate performance in imbalanced datasets. This is known as the accuracy paradox. For example, in an eye-tracking dataset dominated by fixations, an algorithm that mostly predicts fixations could still achieve a high F1-score even if it performs poorly at detecting the less frequent saccades.[**startsev2022**] The Cohen's kappa addresses this, as it adjusts for the agreement that would be expected by chance, providing a more reliable measure of true classification performance. Kappa values range from -1 (systematic disagreement) to 1 (perfect agreement), with 0 indicating chance-level performance [**cohen1960**].

Although a sample-level evaluation is a straightforward way to compare individual data points, it can significantly overestimate performance. This is demonstrated in the Figure 2.12, where a ground truth dataset of only fixation samples (green) are compared to the output samples of two different algorithm predictions. Even though one algorithm makes more classification errors by breaking a single fixation event (blue) into multiple segments with three incorrectly detected one sample long saccade events (red) on the right, versus just one three sample long saccade event on the left, both algorithms appear to have the same accuracy under a sample-level evaluation.[**startsev2022**]

This misleading assessment is problematic, because it fails to capture the adverse impact on eye movement metrics like mean fixation duration, which form the basis for most analyses in gaze behavior research. To avoid this, an event-level evaluation is necessary.[**zemblys2017**]

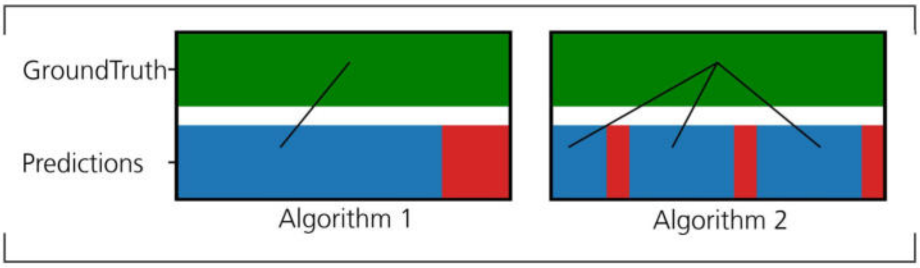


Figure 2.12.: **Limitations of Sample-Level Performance Evaluation based on startsev2022.**

2.5.3. Event-Level Evaluation

For event-level matching, an event matching method is required. **startsev2022** compares and evaluates various event matching techniques and ultimately recommends the maximum intersection over union (IoU) matching method. Compared to other approaches, maximum IoU is more robust in detecting events that align closely with ground truth events in time, while allowing for slight variations in event duration and minor temporal shifts.

The maximum IoU method compares detected events such as fixations and saccades with ground truth events by quantifying their overlap. It measures the degree of overlap between a ground truth event and a predicted event relative to the total time span covered by both events.[**everingham2014**] Thereby, the IoU value ranges from 0, indicating no overlap, to 1, representing a perfect match.[**startsev2022**]

If multiple predicted events overlap with the same ground truth event, the one with the highest IoU is selected as the best match. Any remaining overlapping events are either assigned to another ground truth event or treated as errors or mismatches, ensuring that each event is matched only once. As a result, the same classification categories

TP, TN, FP, FN used in sample-level evaluation can also be applied to event-level evaluation.[**startsev2018**]

Consequently, an event-level confusion matrix can be constructed, alongside performance metrics such as the F1-score and Cohen's kappa. Once the events in the predicted dataset are matched to those in the ground truth dataset and categorized as TP, TN, FP, or FN, these metrics can be calculated in the same way as in sample-level evaluation, allowing for a consistent assessment of classifier performance.[**startsev2018**]

3. Design of an Offline Eye Movement Event Detector for Helicopter Use

The primary objective of this chapter is to identify algorithms capable of reliably detecting fixations and saccades under the specific conditions of a helicopter simulator, and to determine at least one algorithm suitable for online deployment. The actual online deployment and its verification will be presented in the following chapter.

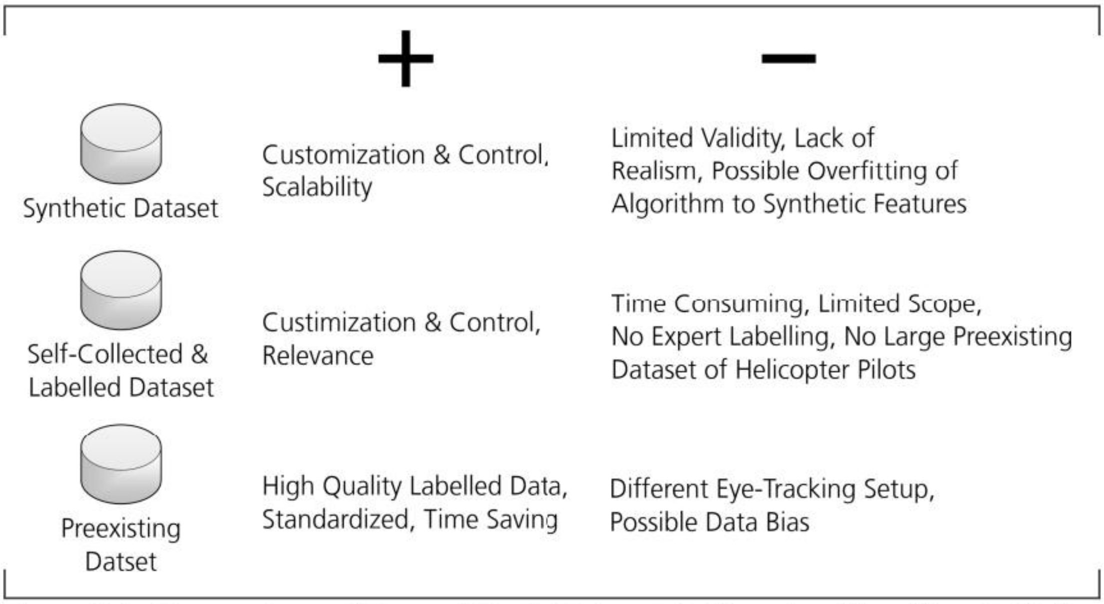
To this end, candidate algorithms and a suitable ground truth are first selected. The algorithms are then optimized using a fixation–saccade invocation task study in a helicopter simulator, followed by a presentation of the resulting performance outcomes. The chapter concludes with a critical evaluation of the algorithms, discussing methodological limitations, comparing them against a baseline and selecting the most promising candidates for subsequent online verification.

3.1. Selection of Eye Movement Event Detectors and Ground Truth

Because no eye movement event detection algorithms have been specifically developed for helicopter pilots, this section begins with the selection of suitable ground truth data, which forms the basis for developing offline algorithms optimized for helicopter use. It then outlines the choice of candidate algorithms, including the criteria and rationale guiding their selection.

3.1.1. Selection of Ground Truth

There are three main forms of datasets that can serve as ground truth for optimizing eye movement event detection algorithms in a helicopter environment. These are synthetic datasets, self-collected, and manually labeled datasets and preexisting open-source datasets. The choice of dataset is critical, as the right hyperparameter optimization depends directly on the quality of the selected ground truth. The distinct advantages and limitations of each approach are summarized in Figure 3.1.



	+	−
Synthetic Dataset	Customization & Control, Scalability	Limited Validity, Lack of Realism, Possible Overfitting of Algorithm to Synthetic Features
Self-Collected & Labelled Dataset	Custimization & Control, Relevance	Time Consuming, Limited Scope, No Expert Labelling, No Large Preexisting Dataset of Helicopter Pilots
Preexisting Datset	High Quality Labelled Data, Standardized, Time Saving	Different Eye-Tracking Setup, Possible Data Bias

Figure 3.1.: **Comparison of Ground Truth Dataset Options for Algorithm Optimization.**

A synthetic dataset refers to artificially designed tasks that elicit specific eye movement patterns, such as the fixation–saccade invocation task. In such tasks, participants are presented with visual stimuli that trigger sequences of fixations and saccades, enabling the algorithm's performance to be directly evaluated against expected outcomes.

The key advantage of this approach is that it provides clearly defined ground truth events, ensuring reproducibility across participants and studies. However, these tasks are inherently artificial, which limits their ecological validity and may fail to capture the full variability of gaze behavior encountered in realistic helicopter operations.

A self-collected and manually labeled dataset involves collecting gaze data within a helicopter flight simulator and annotating its eye movement events by trained experts.

This approach has the advantage of capturing authentic gaze behavior under simulator conditions, thereby providing high quality ground truth that reflects the operational environment of helicopter pilots. However, the collection and manual labeling process is extremely resource intensive and time consuming, particularly since no preexisting datasets for helicopter pilots are available. Consequently, scalability is very limited, making it impractical for larger datasets required in algorithm optimization.

Open-source datasets are publicly available collections of eye-tracking data that have been hand-labeled by experts.

Their main advantages lie in their accessibility and standardization. They are freely available, provide high quality manual annotations and can be used for cross-study comparisons. Nevertheless, these datasets have drawbacks for an event detection of helicopter pilots. They are typically collected in laboratory settings on monitor-based setups, often assuming a constant viewing distance. This contrasts with helicopter simulators, where the primary vision system and instrument panels differ in distance. Furthermore, such datasets rarely use head-mounted eye trackers, reducing ecological validity even further. As a result, their direct application to helicopter pilot studies is limited due to differences in visual environments and task demands.[**startsev2022**]

Among these options, the synthetic dataset is the most suitable for hyperparameter optimization of event detection algorithms in helicopter flight simulators. While no precedent synthetic dataset was generated in a flight simulator before and overfitting of the algorithms to the datasets synthetic features is possible, distinct fixation patterns of helicopter pilots are derivable in literature on helicopter pilot eye movement behavior [**greiwe2022**, **greiwe2023**, **greiwe2023a**]. Thus, a synthetic dataset mimicking helicopter pilot eye movements can be created using a fixation-saccade invocation task, as already done successfully for the development of event detection algorithms in other contexts [**komorgotsev2010**, **olsen2012a**].

The open-source datasets lack uniform eye movement definitions and oftentimes contain more labels than fixations and saccades, requiring reclassification for a fixation filter design. Moreover, these datasets are collected with eye trackers that differ from the one used in this thesis, producing incompatible data formats (e.g., 2D pixel coordinates rather than 3D Euclidean distances) and exhibiting distinct noise characteristics due to differences in precision, accuracy, and calibration. These differences risk overfitting the optimization process to the characteristics of a different setup and eye tracker. In addition, many open-source datasets are recorded at higher sampling rates than the eye tracker used in the thesis, meaning they capture subtle saccades that may be lost during downsampling, potentially leading to aliasing effects and the removal or distortion of short saccades.[**startsev2022**]

In contrast, self-collected and manually labeled datasets would provide the highest ecological validity, but are not viable within this thesis. The lack of available expert annotators at DLR, combined with the time constraints of the project, make manual labeling infeasible. Using an existing event detection algorithm to label self-collected data as a substitute ground truth like Tobii's own velocity threshold filter is also unsuitable, since such algorithms lack the labeling accuracy of established ground truths and are not optimized for helicopter use, thereby undermining the primary aim of this thesis.

This makes a synthetic dataset based on a study incorporating a fixation–saccade invocation task the most reliable and practical ground truth dataset for developing eye movement event detectors tailored to helicopter pilot use.

3.1.2. Selection of Eye Movement Event Detectors

As outlined in the literature review in Section 2.4, three main algorithmic approaches for eye movement event detection exist. These approaches must be evaluated and compared to determine their suitability for optimization and application inside a helicopter flight simulator.

While threshold-based algorithms are simple, easy to implement and computationally efficient, their effectiveness is highly dependent on the choice of threshold. However, their performance is highly dependent on the choice of threshold, which also makes them more sensitive to noise compared to other approaches. Within this class, the I-DT reduces threshold-related noise sensitivity by integrating gaze positions over a time window, rather than classifying events based on single gaze samples. The I-VT, by contrast, typically requires smoothing filters, such as a moving average, to achieve similar robustness to noise [**salvucci2000**, **olsen2012**].

Both I-DT and I-VT are well-suited for real-time applications, as their simplicity and computational efficiency make them ideal for online use, although they generally lack the accuracy of other event detection approaches.[**salvucci2000**]

Probability-based methods offer several advantages over threshold-based algorithms. By assigning probabilities to each event classification rather than a strict threshold, they are more robust to noise and less affected by spurious fluctuations or outliers. This approach also provides greater adaptability, as parameters can be more precisely fine-tuned for helicopter pilots. Furthermore, the assigned parameters provide quantitative measures that support more nuanced analyses or downstream decision-making. [**jurafsky2025**, **salvucci2000**]

Despite these benefits, probability-based methods also have limitations. They are typically more computationally demanding than threshold-based algorithms, since they must continuously calculate the likelihood of each eye movement state and determine the most probable sequence of events, which requires additional calculations. Due to their dependence on more parameters, a larger dataset for optimization may also be required. Additionally, implementing these models often demands expertise in statistics, as their performance is sensitive to the assumptions made about the underlying probability distributions. [**birawo2022**, **salvucci2000**] Probability-based algorithms are also suitable for real-time applications, although their responsiveness varies significantly depending on the specific model. Simpler models like the I-KF can operate with low latency, whereas more complex approaches, such as HMMs, introduce higher computational costs. While they may have a small increase in latency compared to threshold-based algorithms due to their computation time, they are preferred in cases when increased robustness to noise is a higher priority than minimal extra latency. [**komogortsev2007**, **salvucci2000**]

Unlike the other approaches, deep learning models can directly process raw gaze coordinates without requiring preprocessing steps such as angular velocity calculation. They automatically learn the most informative features and capture complex, non-linear patterns that human experts might overlook. This eliminates the need for predefined thresholds or probability models, making them inherently more robust to noise and small fluctuations [**hoppe2016**]. Their ability to learn from data also enables effective generalization across individual differences. Combined with their capacity to model temporal dependencies, these properties allow deep learning models to achieve high accuracy in eye movement classification.[**elmadjian2021**]

Despite their advantages, deep learning techniques face significant challenges. First, they require substantially larger amounts of labeled training data than threshold- or probability-based approaches, which limits their practicality in domains where annotated datasets are scarce [**birawo2022**]. Another major challenge is overfitting, which happens when models learn the training data and its noise too well, leading to poor generalization on new gaze data. This problem is exacerbated if the dataset is biased or unrepresentative, further compromising accuracy. Retraining models for new tasks or datasets are also more demanding than adjusting simpler algorithms. Finally, interpretability is another critical limitation. Deep learning models are often treated as "black boxes," making it difficult to explain or validate their predictions.[**hoppe2016**, **kumar2020**]

Recent advances demonstrate that deep learning models can achieve low latency suitable for real-time applications. **elmadjian2021** showed that CNN-LSTM, CNN-BiLSTM and TCN architectures can operate with very low computational overhead, achieving processing latencies below 2 ms on standard hardware. Event detection accuracy can be improved by using a 40–60 ms look-ahead window, although the gain is small.

Importantly, the TCN operates effectively without any look-ahead, underlining its strong potential for online deployment.

Based on the discussed advantages and disadvantages of all three eye movement event detection approaches, an overview of their performance is summarized in the radar chart shown in Figure 3.2. The criteria are evaluated based on whether each approach is generally considered to not meet, sufficiently meet, or exceed the requirements for algorithm development and subsequent online applicability in the specific context of this thesis.

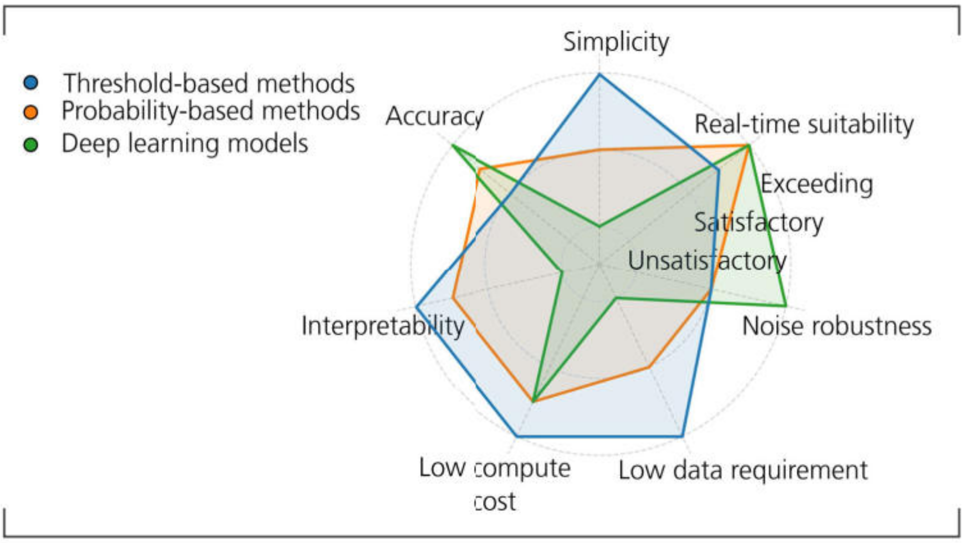


Figure 3.2.: **Comparison of Eye Movement Detection Approaches.**

As shown in Figure 3.2, deep learning models do not satisfy the requirements in terms of interpretability and data efficiency, since, as outlined in the previous section, no large-scale dataset is available and interpretability of algorithm results is crucial for later evaluation of optimization results and for validating their event detections. They are therefore excluded from further analysis. For threshold- and probability-based algorithms, the choice is less obvious. Since no single method clearly emerges as the most suitable for helicopter applications, a representative set of algorithms with different detection principles is selected. From the threshold-based category, the I-VT and I-DT algorithms are chosen. Both are widely established methods. The I-VT serves as Tobii's standard event detection filter [**olsen2012**], while the I-DT has demonstrated strong performance in prior work [**salvucci2000**]. Accordingly, the I-VT used in this thesis is implemented based on Tobii's I-T Fixation Filter, and the I-DT follows the implementation described by **salvucci2000**.

Within the probability-based category, the I-KF, based on **komogortsev2007**, and I-HMM, based on **salvucci2000**, are selected. The I-KF is relatively simple, with fewer parameters, while the I-HMM is more complex, having more parameters. Including both allows for the evaluation of two distinct probability-based approaches, which supports a more comprehensive assessment and helps identifying the most effective algorithm for helicopter applications.

3.2. Algorithm Optimization Methodology

This chapter presents the methodology employed for algorithm optimization using a fixation-saccade invocation task study.

The chapter begins with a description of the study setup, including experimental environment, equipment and participant details, to ensure reproducibility. After that, the fixation-saccade invocation task is detailed, which forms the basis for collecting gaze data for algorithm optimization. The study procedure section outlines the study workflow and task execution. Finally, the evaluation method of the offline algorithms is presented, which includes the metrics and optimization method used to assess and optimize performance based on the collected fixation-saccade invocation data.

3.2.1. Study Setup

The eye tracker used for the study is the Tobii Pro Glasses 3 (Figure 3.3). It is equipped with a Full HD scene camera that captures the user's Field Of View (FOV) (95 ° horizontally and 106 ° vertically) and a microphone for audio recording. The gaze is measured with an average accuracy of 0.6 ° by using the PCCR method described in Section 2.1.1. The PCCR method utilizes 8 infrared illuminators and 2 infrared cameras on each eye to enable gaze recording. The data is sampled at 50 Hz and stored on a dedicated recording unit that is connected to the eye tracker glasses.[**tobii2023c**]

The fixation-saccade task is conducted in DLR's 2PASD (Dual Pilot Active Sidestick Demonstrator) fixed based flight simulator depicted in Figure 3.4. The simulator features two pilot stations equipped with cyclic and collective sidesticks, as well as pedal inceptors. The primary visual system comprises five LCD screens that have a diagonal size of 55

[1] Source: DLR (CC BY-NC-ND 3.0)

Figure 3.3.: **Tobii Pro Glasses 3.**[1]

inch (resolution of 1920×1080 pixels, refresh rate 60 Hz), arranged in an arc around the cockpit. The central screen is mounted in landscape orientation, while the surrounding screens are in portrait orientation, providing a combined FOV of 150° horizontally and 50° vertically.

In addition, each pilot seat is equipped with a 17 inch instrument display (resolution of 1280×1024 pixels, refresh rate 60 Hz) in front and two smaller 10 inch displays (resolution of 768×1024 pixels, refresh rate 60 Hz) stacked vertically in between to provide further instrument readouts.[**dikarew2024**]

Eleven participants (10 male, 1 female; age range: 20–50 years, mean = 28.6, standard deviation = 6.6) took part in the study. Some wore corrective lenses or had blue eyes, potentially inducing additional noise. Consequently, the dataset spans a wide range of noise levels, providing a suitable ground truth for an exhaustive evaluation of the algorithms' performance.

3.2.2. Fixation-Saccade Invocation Tasks

To generate a synthetic dataset, a set of six fixation–saccade invocation tasks are designed, in which fixations and saccades are elicited by presenting a sequence of visual stimuli that participants are instructed to follow. The tasks are incorporated in a fixation–saccade invocation study that builds on earlier work by **komorgotsev2010** and **olsen2012a**, but differs in its objective, as it is designed specifically for helicopter pilots

[2] Source: DLR (CC BY-NC-ND 3.0)

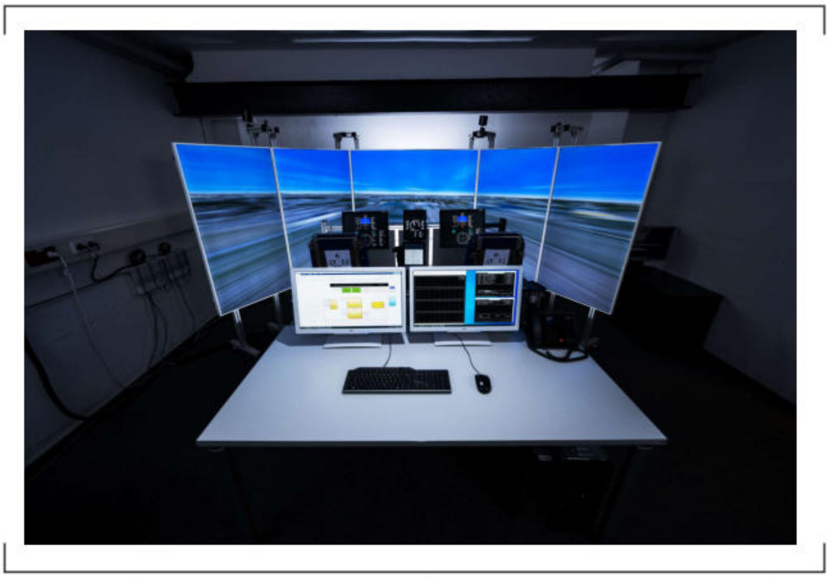


Figure 3.4.: **2PASD (Dual Pilot Active Sidestick Demonstrator) Simulator.**[2]

and conducted within a helicopter flight simulator, which requires additional considerations.

To ensure algorithm parameters are tuned specifically for helicopter pilots, the stimulus presentation patterns must be designed to cover a broad spectrum of visual demands encountered in helicopter operations. To achieve this, the stimulus presentation patterns are derived from previous studies [**greiwe2022**, **greiwe2023**, **greiwe2023a**].

Furthermore, the cockpit monitors are nearer to the pilot seat than the TV screens for the primary vision system. In order to make the visual stimuli equally discernible in both screens, the stimulus size is set in visual angle to 1 degree, as also defined by **komorgotsev2010**. Moreover, the stimulus appears as a white dot with a smaller black dot at its center, similar to a typical calibration marker, and is displayed on a black background.

In all tasks, the stimulus jumps between 10 fixed locations to elicit saccades and subsequent fixations, except in one task, where it moves smoothly across the visual field to elicit three different smooth pursuit. Additionally, all six tasks are presented either in the center display, side display or cockpit monitor as highlighted in Figure 3.5.

In the baseline task T1, the stimulus appears only on the center display of the simulator's primary vision system. It jumps between predefined positions, eliciting short- to medium-

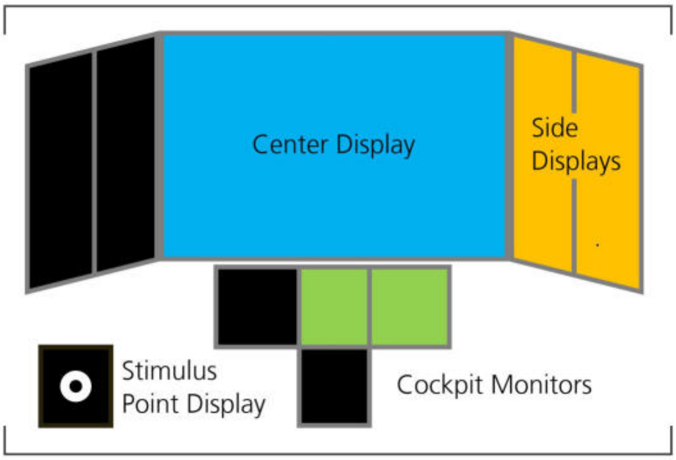


Figure 3.5.: **Stimulus Presentation Displays in the 2PASD.**

amplitude saccades ranging from 5 ° to 20 °. This task represents basic gaze behavior directed Outside The Window (OTW).

In task 2 (T2), the stimulus alternates between the center primary display and the cockpit instrumentation monitors. The jumps require participants to shift gaze between the external visual scene and cockpit displays, with amplitudes between 10 ° and 25 °, mimicking typical transitions between OTW and Inside The Cockpit (ITC).

Task 3 (T3) extends T2 by presenting stimulus jumps within clusters of cockpit displays and the center screen, resulting in saccade amplitudes between 3 ° and 25 °. By concentrating sequences of jumps in small spatial clusters before moving to another region, the task simulates monitoring strategies in which pilots scan instruments and visual cues OTW before returning focus outside.

In task 4 (T4), stimulus jumps are limited to the side displays of the simulator. This requires saccades of 3 ° to 15 ° into the periphery, reproducing conditions in which pilots must monitor lateral visual cues presented outside the central field of view.

In the final jumping point task T5, the stimulus appears exclusively on the cockpit monitors. This design emphasizes instrument-scanning behavior, requiring sequences of short- to medium-amplitude saccades ranging from 3° to 10° within the cockpit instrument panel.

Unlike the previous jump-based tasks, task 6 (T6) involves continuous motion of the stimulus along trajectories on the side displays. Participants are required to follow the moving dot, which gradually accelerates over time and elicits smooth pursuit eye movements combined with saccades when fixating on to the next moving stimulus position. This task reflects situations in which pilots track moving cues in the peripheral field.

The timing for the jumping stimulus presentation in the fixation–saccade invocation tasks T1 to T5 is shown in Figure 3.6. The timeline consists of the four successive intervals that are Saccade Latency (SL), Saccade Duration (SD), Modal Fixation Duration (MFD) and the overlap.

The first 250 ms of the stimulus presentation accounts for the SL. After the SL another 50 ms has to be accounted for the SD. The durations of both values are chosen based on section 2.2.2. After SL and SD, cognitive processing on the stimulus is assumed to occur. For this processing interval the modal fixation duration of 190 ms is used, as it is also used and suggested as the reference fixation duration by **greiwe2022** for helicopter pilots eye movement analysis. An additional overlap of 90 ms is included at the end of the display time, during which the current and subsequent stimuli are shown simultaneously. This overlap prevents a blank screen interval that might induce wandering and ensures that saccades between targets are more likely to be direct, thereby reducing intermediate search fixations. Given the study's video frame rate of 10 Hz, the overlap used by **olsen2012a** is rounded up to 90 ms. As this overlap duration is shorter than a typical saccadic latency, any fixation that begins during this period is therefore attributed to the previous target or to transition behavior and not the new stimulus.

In total, the sequence of these intervals sums to an effective display time of 580 ms per stimulus. This fixation display time falls between the 1 second duration prescribed by **komorgotsev2010** and the 500 ms duration applied by **olsen2012a**.

3.2.3. Study Procedure

The study procedure for each participant follows the flow chart shown in Figure 3.7. It begins with an explanation of the study's purpose and the collection of informed consent for gaze data recording. Afterwards, the eye tracker is calibrated to the participant using a one-point manual calibration with a calibration card, as recommended for Tobii head mounted eye trackers to ensure highest data accuracy during recording [**tobii2023c**].

Following calibration, a familiarization phase is conducted. In this phase, participants are introduced to the general concept of the tasks and explicitly informed that no restrictions

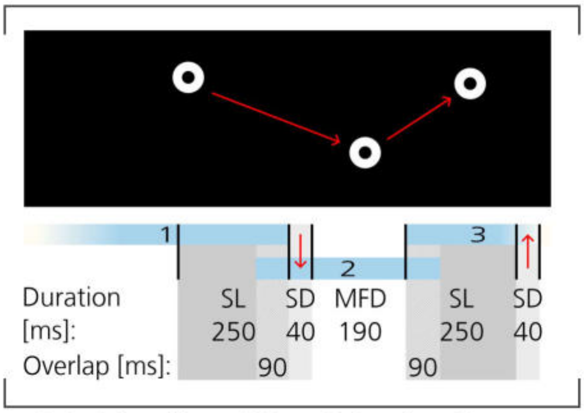


Figure 3.6.: **Timeline of One Stimulus Presentation.**

are imposed on head movement or position. To demonstrate the two task types, two example scenarios are demonstrated in the simulator, resembling the jumping point task T1 and the smooth pursuit task T6. This phase solely ensures task comprehension and reduction in novelty effects and is not recorded for the algorithm evaluation.

After that, the fixation-saccade invocation tasks are performed, during which all eye-tracking- and simulator data will be recorded during this phase. Prior to each task a brief instruction stating the specific displays where stimuli will appear for that particular task is provided. Crucially, no demonstrations of the stimuli patterns is shown during this phase. This is to prevent participants from anticipating the exact timing or location of stimuli within the specified areas, which encourages more natural and reactive eye movements for data collection. All tasks are ordered in a unique sequence for each participant, following a counterbalanced Latin square design. The final task order matrix is shown in Table 3.1, where rows (P1–P12) represent the number of participants and columns ("1st"-"6th") indicate the sequential order in which each task (T1–T6) is performed. This design ensures that each task appears equally often in each sequence position, thereby controlling for order effects such as practice and fatigue [**lewis1989**].

Since counterbalanced Latin squares require a round number of participants, the matrix is constructed for 12 participants even though only 11 take part in the study. Besides, the test matrix is structured into two blocks to facilitate the training and testing of the algorithms, resulting in each participant completing every task twice.

Throughout the tasks gaze data is continuously recorded and saved for later algorithm evaluation. After each task, participants complete a short questionnaire assessing perceived task difficulty and mental workload on a Likert scale. In addition, a post-study questionnaire collects demographic and participant specific information. Together, these

Table 3.1.: **Task Ordering Matrices.**

Participant	**Training Block**						**Testing Block**					
	1st	**2nd**	**3rd**	**4th**	**5th**	**6th**	**1st**	**2nd**	**3rd**	**4th**	**5th**	**6th**
P1	T1	T2	T6	T3	T5	T4	T4	T1	T2	T6	T3	T5
P2	T2	T3	T1	T4	T6	T5	T5	T2	T3	T1	T4	T6
P3	T3	T4	T2	T5	T1	T6	T6	T3	T4	T2	T5	T1
P4	T4	T5	T3	T6	T2	T1	T1	T4	T5	T3	T6	T2
P5	T5	T6	T4	T1	T3	T2	T2	T5	T6	T4	T1	T3
P6	T6	T1	T5	T2	T4	T3	T3	T6	T1	T5	T2	T4
P7	T5	T4	T1	T2	T6	T3	T3	T5	T4	T1	T2	T6
P8	T6	T5	T2	T3	T1	T4	T4	T6	T5	T2	T3	T1
P9	T1	T6	T3	T4	T2	T5	T5	T1	T6	T3	T4	T2
P10	T2	T1	T4	T5	T3	T6	T6	T2	T1	T4	T5	T3
P11	T3	T2	T5	T6	T4	T1	T1	T3	T2	T5	T6	T4
P12	T4	T3	T6	T1	T5	T2	T2	T4	T3	T6	T1	T5

questionnaires support the traceability of possible anomalies such as poor data quality or unexpected gaze behavior by providing contextual information for gaze data evaluation. The questionnaire is appended in Appendix C.

3.2.4. Evaluation of the Offline Algorithms

To evaluate the offline algorithms, a hyperparameter evaluation pipeline is established, as shown in the schematic overview in Figure 3.8.

Within this pipeline, the optimization process iteratively explores combinations of parameter values for each algorithm, with the objective of identifying the settings that yield the highest possible performance. The raw gaze data and ground truth annotations used for optimization are taken from the training dataset, which is organized into the five jumping point tasks (T1–T5), each containing data from all participants. For every ground truth and raw gaze data pair, the offline algorithm is executed with a proposed hyperparameter set. The resulting eye movement classifications are then compared against the ground truth using an IoU based scoring method. Details of both the IoU-based scoring method and the ground truth fixation generation are provided in Section 3.2.4 and 3.2.4.

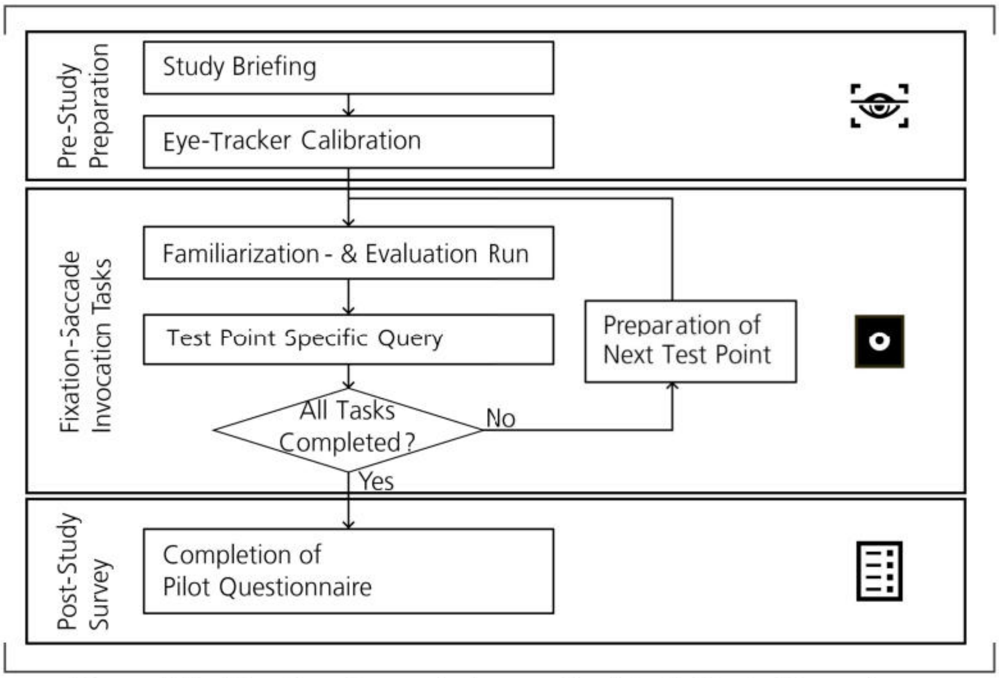

Pre-Study Preparation
Study Briefing
Eye-Tracker Calibration
Fixation-Saccade Invocation Tasks
Familiarization - & Evaluation Run
Test Point Specific Query
All Tasks Completed?
No
Yes
Preparation of Next Test Point
Post-Study Survey
Completion of Pilot Questionnaire

Figure 3.7.: **Fixation-Saccade Invocation Task Study Procedure.**

The scoring method produces Cohen's kappa and F1-score values for each task. To obtain a single optimization objective, the unweighted average of Cohen's kappa across all five tasks is calculated and maximized, which ensures equal contribution of each task. Only Cohen's kappa is used as the optimization metric, since it provides a more reliable measure of agreement than the F1-score (see Section 3.2.4). Nevertheless, the F1-scores are still reported for comparison with other eye movement event detectors, given their frequent use in the literature.

The optimization is conducted with the Optuna framework, which is described in detail by **takuya2019**. Optuna adaptively selects new hyperparameter sets to evaluate, guided by the Kappa results of previous trials. This allows the framework to focus on promising regions of the parameter space while efficiently discarding unpromising ones, leading to faster convergence compared to random or exhaustive search.

After a predefined number of trials, the framework reports the optimal hyperparameter set for each algorithm, along with their corresponding Kappa and F1-scores. These best-performing configurations are then validated on the testing dataset and the resulting Kappa and F1 values are used for the final evaluation of each algorithm.

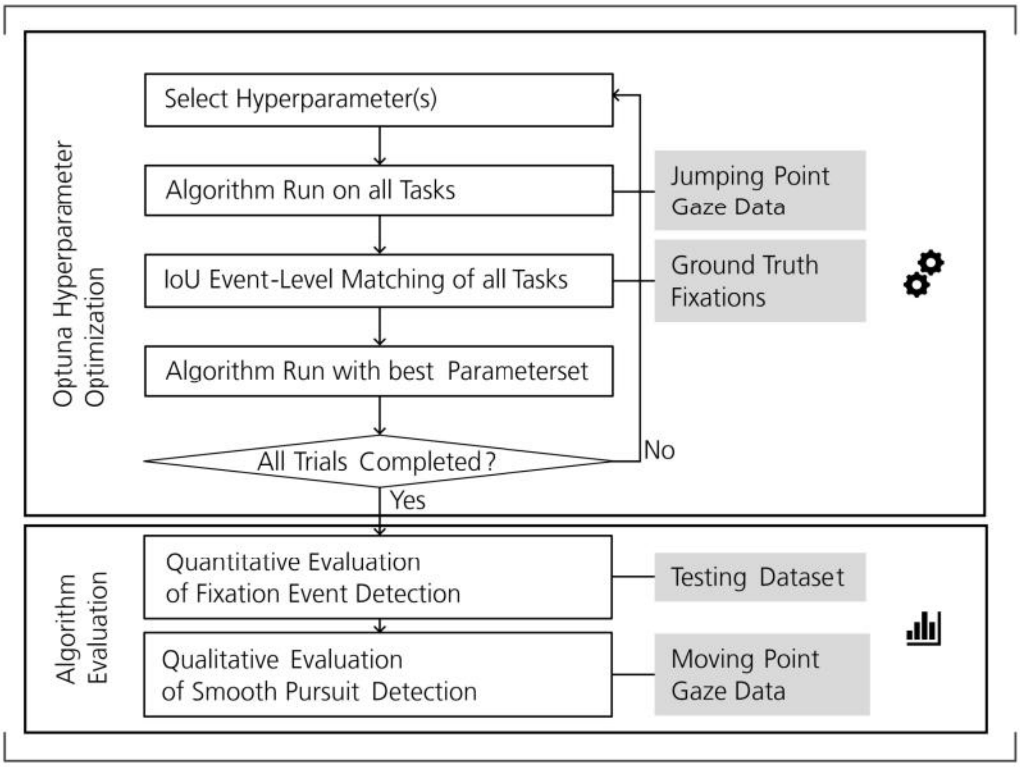

Optuna Hyperparameter Optimization
Select Hyperparameter(s)
Algorithm Run on all Tasks
IoU Event-Level Matching of all Tasks
Algorithm Run with best Parameterset
All Trials Completed?
No
Yes
Jumping Point Gaze Data
Ground Truth Fixations
Algorithm Evaluation
Quantitative Evaluation of Fixation Event Detection
Qualitative Evaluation of Smooth Pursuit Detection
Testing Dataset
Moving Point Gaze Data

Figure 3.8.: **Overview of Algorithm Hyperparameter Optimization and Evaluation.**

Furthermore, the parameter spaces explored for each algorithm are summarized in Table 3.2. For all algorithms, the post-processing parameters are fixed according to Tobii's I-VT filter settings to reduce the number of hyperparameters and enhance interpretability of the results. In addition, the I-VT applies Tobii's moving median window size, which is not used for the other algorithms, so that its detection method aligns with the I-TobiiVT filter. The hyperparameter intervals for the remaining variables of all algorithms are defined based on established knowledge of fixation and saccade characteristics, as outlined in Section 2.2.

Algorithm Scoring Method

The IoU based scoring method is chosen over the approaches of **komorgotsev2010** and **olsen2012a** due to their scoring metrics such as Average Number of Fixations (ANF), Average Fixation Duration (AFD) describing eye movement behavior. Consequently, for these behavioral metrics to deliver accurate assessment of the algorithm

Table 3.2.: **Overview of Algorithm Specific Parameter Spaces and Fixed Values Used in the Evaluation.**

Algorithm	Parameter	Range
I-TobiiVT	Moving median window size	3 (fixed)
	Velocity threshold	100 °/s (fixed)
	Minimum fixation duration	60 ms (fixed)
	Maximum gap duration between fixations	75 ms (fixed)
	Maximum gap angle between fixations	0.5° (fixed)
I-VT	Velocity threshold	5 – 150 °/s
I-DT	Dispersion duration threshold	0.01 – 0.8 s
	Dispersion threshold	0.1 – 1.0°
I-HMM	Training epochs	50 – 500
	Convergence tolerance	$1e^{-8}$ – $1e^{-2}$ (log scale)
	Start probability	0.01 – 0.99
	Transition probability	0.7 – 0.99
	Mean saccade velocity	1 – 200 °/s
	Mean saccade angle	0 – 30°
	Mean fixation velocity	0 – 50 °/s
	Mean fixation angle	0 – 30°
	Variance saccade velocity	1 – 500
	Variance saccade angle	0.001 – 20 (log scale)
	Variance fixation velocity	1 – 200
	Variance fixation angle	0.001 – 20 (log scale)
I-KF	Chi square threshold	30 – 200
	Velocity process noise	$1e^{-5}$ – 1.0 (log scale)
	Position process noise	$1e^{-4}$ – 1.0 (log scale)

event detection accuracy, the subject must follow the jumping point stimulus exactly as instructed whereas the IoU based scoring is independent of it, which provides a more robust evaluation framework. Moreover, IoU based matching enables the calculation of the well-established classification performance metrics F1 score and Cohen's kappa, which are widely used for evaluating eye movement detection accuracy.[**startsev2022**]

In this scoring method, ground truth fixations and predicted events are matched based on their temporal overlap using the same IoU thresholding procedure described in Section 2.5. If a ground truth fixation is matched with a predicted fixation a TP is counted and conversely if matched with a predicted saccade counted as FN. For all other matching scenarios, a special penalization strategy is applied, which is illustrated using a ground truth and prediction matching example in Figure 3.9.

In the case of under-segmentation, where one predicted fixation spans multiple ground truth fixations, a single TP is assigned and the surplus ground truth fixations are penalized as FN. Conversely, in cases of over-segmentation, where multiple predicted fixations cover a single ground truth fixation, one TP is counted and the excessive predicted fixations are penalized as FP.

TN are derived indirectly, since the dataset does not provide ground truth annotations for saccades. Consequently, no over- or under-segmentation is applied in this case, as the non-fixation segments in the ground truth merely indicate that at least one saccade must have occurred, without specifying its exact amount or duration. Accordingly, within each ground truth non-fixation segment, the presence of one or more predicted saccades is consistently counted as a single TN.

With the obtained counts of TP, FP, FN, and TN, both Cohen's kappa and the F1 score are computed to quantify detection accuracy and guide hyperparameter optimization.

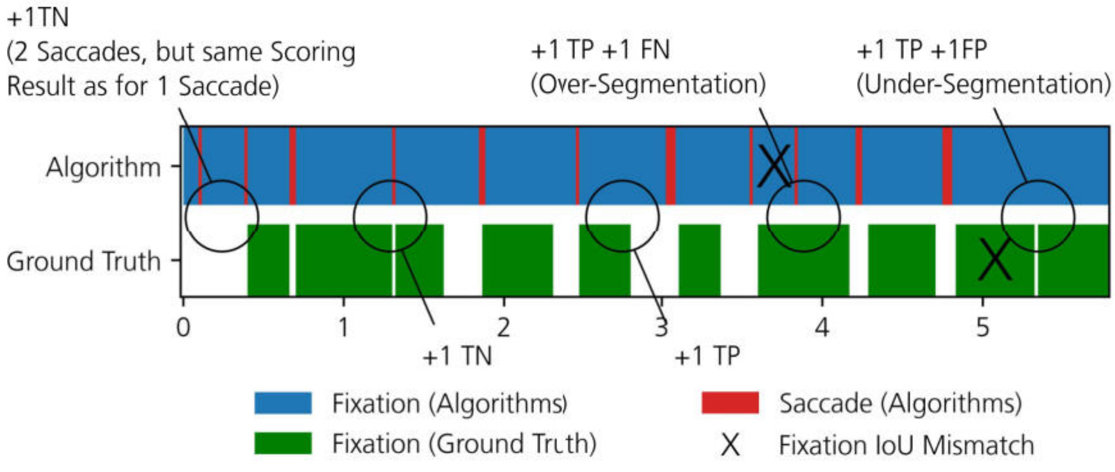


Figure 3.9.: **Example of Ground Truth and Prediction Matching for Event Scoring.**

Ground Truth Generation

The ground truth fixations are identified by measuring the temporal and spatial alignment between the jumping stimulus point and the participant's gaze.

Due to small physiological movements and noise, such as those induced by suboptimal calibration of the eye tracker to the participant's eyes or inaccuracies in the device's gaze estimation, a spatial tolerance between the stimulus point and the gaze point is set up. Hereby, angular deviation of the gaze from the ideal stimulus position is measured. Gaze samples that fall within the defined angular threshold are classified as part of a stable fixation on the stimulus and samples outside this threshold are labeled as unknown eye movements, since both fixations and saccades could occur outside the stimulus points position and cannot be verified.

Angular thresholds of around 0.6 were investigated, as this corresponds to the accuracy of the eye tracker. As shown in Figure 3.10, which illustrates ground truth fixations for different thresholds in one example time sequence, a threshold above 0.5° incorrectly merges separate fixations into a single event, whereas the 0.4° threshold correctly distinguishes them. It should be noted, however, that smaller thresholds generally yield shorter fixation durations. Despite this, a 0.4° threshold was chosen, as shorter ground truth fixations can still be reliably matched with the IoU event matching approach. In addition, a post-processing step is applied to regroup fixations that were incorrectly split due to the stricter threshold.

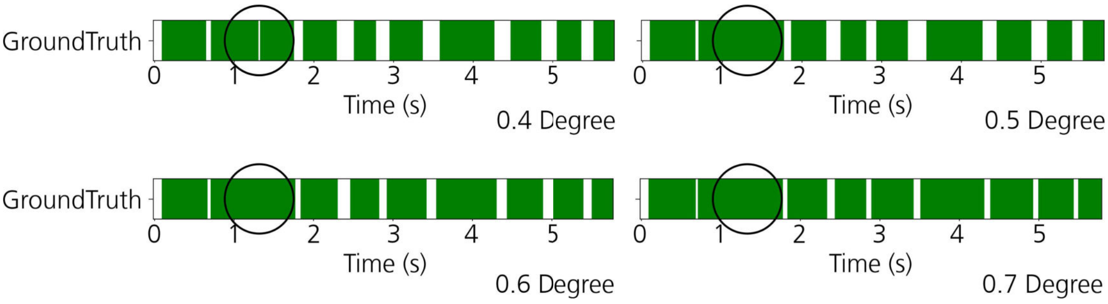


Figure 3.10.: **Ground Truth Fixation Detection at Different Angular Thresholds (0.4°–0.7°) in One Example Time Sequence.**

Temporal alignment is ensured by verifying that gaze points are not only spatially close to the stimulus but also approximately matched in time. The temporal alignment window spans from the onset of stimulus presentation until 300 ms after stimulus offset. This 300 ms margin accounts for the combined saccade latency and duration and is included after stimulus offset as well, since the gaze lingers on the same stimulus after its offset before transitioning to the next stimulus point.

3.3. Overview of Optimization Results

The following section presents the results from the offline algorithm optimization based on the fixation-saccade invocation task study.

First, the optimization results of the I-VT, I-DT, I-KF, and I-HMM algorithms from the Jumping Point Task are reported, followed by a summary of the testing results of all algorithms. After that, the results from the Moving Point Task are presented.

3.3.1. Optimization Results from the Jumping Point Task

The optimization results are presented by showing the parameter sets alongside their corresponding objective values, while also specifically reporting the parameter set values with the highest Cohen's kappa and F1 scores achieved. Since the Optuna framework minimizes the objective function for optimization, the objective value is defined as 1 - Kappa, so that the optimal value is 0 and the worst is 1. Depending on the number of parameters varied, these results are visualized using a graph, a contour plot or a parallel coordinate plot.

I-VT Results

The results of the hyperparameter optimization process for the I-VT algorithm are presented in Figure 3.11, which illustrates the objective value as a function of the velocity threshold.

The data points on the plot are colored on a gradient from light to dark blue, corresponding to the trial number in which the point was calculated. The objective value follows a parabolic curve with a minimum located at an objective value of approximately 0.04 at a velocity threshold of 44.03 corresponding to a Cohen's kappa of 0.96 and F1 score of

0.98. Deviations from this optimal threshold result in an increased objective value, rising more sharply for lower thresholds and more gradually for higher thresholds.

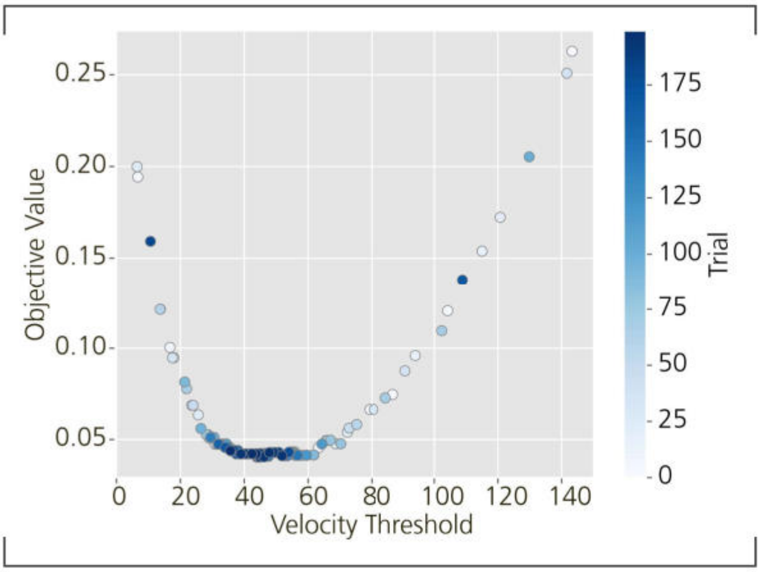


Figure 3.11.: **I-VT Hyperparameter Optimization Results.**

I-DT Results

The contour plots in Figure 3.12 illustrate the distribution of objective values using a blue gradient, with the left plot showing the full range of dispersion and duration thresholds and the right plot providing a close-up view of the optimal parameter region.

The best-performing parameter set is located at a dispersion threshold of 0.324 and a duration threshold of 0.036, achieving an average F1 score of 0.917 and a Cohen's kappa of 0.843. The close-up view further highlights that the optimal parameter space spans duration thresholds between 0.03 and 0.06 and dispersion thresholds between 0.25 and 0.4.

I-KF Results

The results of the I-KF optimization process state for the optimal parameters an average F1 score of 0.978, a Cohen's kappa of 0.957 and a Chi-square threshold of 84.30.

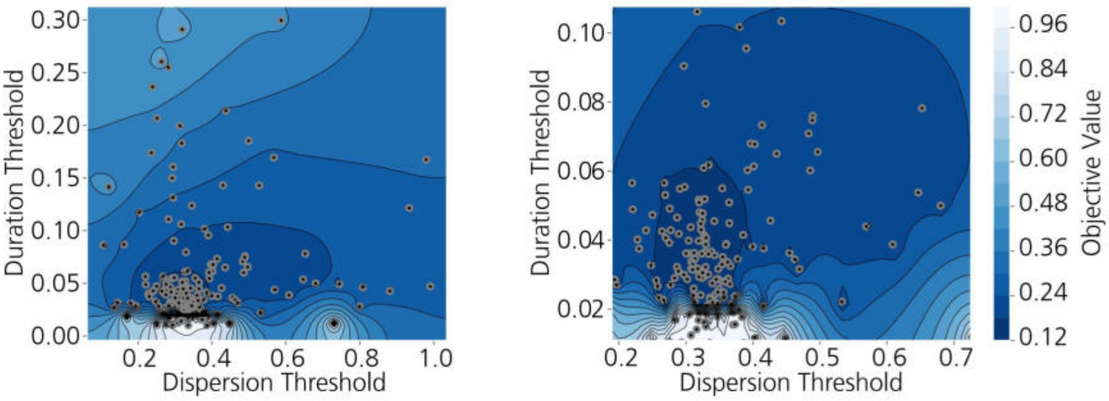


Figure 3.12.: **Contour Plot of the I-DT Optimization Results on the Left and a Close-Up View of the Optimal Parameter Region on the Right.**

Furthermore, the optimal position process noise (q_{pos}) is 0.0015 and the optimal velocity process noise (q_{vel}) is 0.2275.

The parallel coordinate plot in Figure 3.13 depicts the correlations between the objective value and the three optimized hyperparameters. Each vertical axis corresponds to a parameter and each blue line represents a single trial from the optimization. The line color reflects the objective value, with darker lines indicating lower objective values. By identifying the darkest lines, the optimal regions for each parameter can be identified, highlighting the convergence to the optimal parameter set.

I-HMM Results

Finally, the I-HMM results are presented using a parallel coordinate plot, analogous to that of the I-KF, shown in Figure 3.14. Unlike the I-KF plot, no single converging optimum is apparent. The darker lines are distributed across a wide range of values along all parameter axes.

The best-performing parameter configuration, as determined by the optimization process, is detailed in Table 3.3. This configuration achieved a best average F1 score of 0.938 and a Cohen's kappa of 0.88.

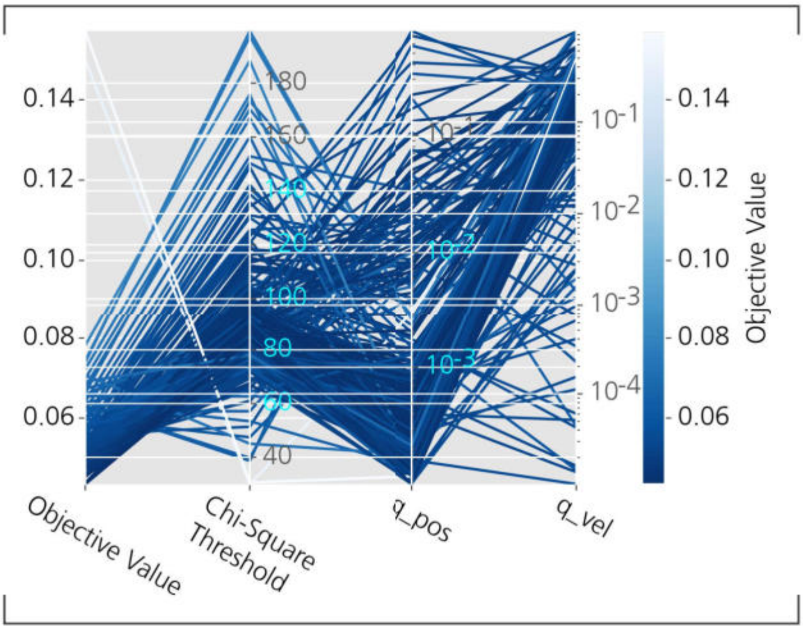


Figure 3.13.: **Parallel Coordinate Plot of the I-KF Hyperparameter Optimization Results.**

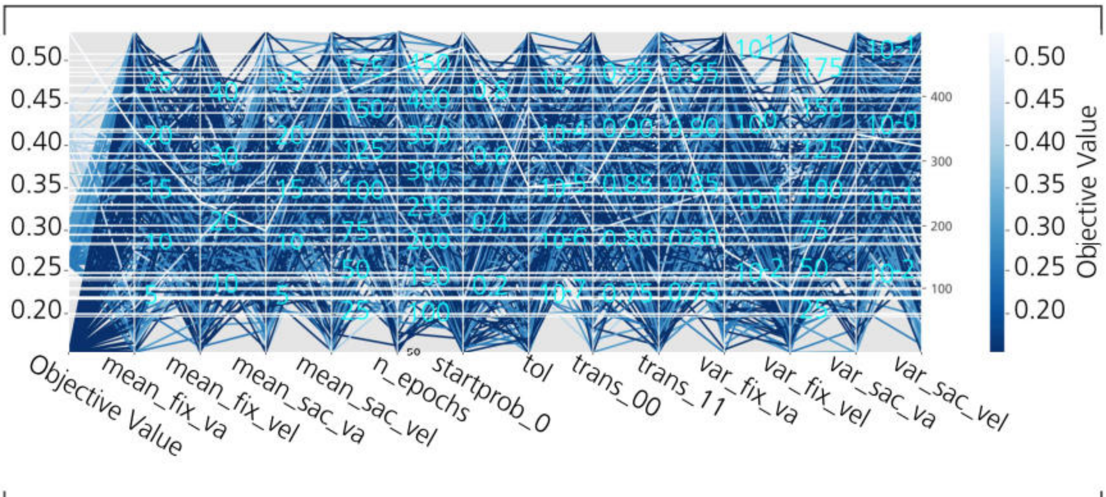


Figure 3.14.: **Parallel Coordinate Plot of the I-HMM Hyperparameter Optimization Results.**

Table 3.3.: **Best Parameter Configuration for the I-HMM Algorithm.**

Category	Parameter	Value
Training	Epochs (n_epochs)	353
	Convergence tolerance (tol)	0.0005
State probabilities	Start probability ($startprob\_0$)	0.123
	Transition probability ($trans\_11$)	0.735
State means	Mean saccade velocity ($mean\_sac\_vel$)	63.8 °/s
	Mean saccade angle ($mean\_sac\_va$)	20.2°
	Mean fixation velocity ($mean\_fix\_vel$)	3.15°/s
	Mean fixation angle ($mean\_fix\_va$)	17.6°
State variances	Variance saccade velocity (var_sac_vel)	57.7°/s
	Variance saccade angle (var_sac_va)	0.114°
	Variance fixation velocity (var_fix_vel)	64.1°/s
	Variance fixation angle (var_fix_va)	3.22°

Testing Results of All Algorithms

The results from the testing dataset reinforce the performance trends observed during training. For direct comparison, the F1 scores and Cohen's kappa values are reported for all tasks and all algorithms across both datasets, with the final row summarizing the average results in Figure 3.15.

As in training, the I-VT and I-KF algorithms achieve the highest scores, with F1 scores ranging from 0.94 to 1.00 and Cohen's kappa values between 0.88 and 0.99. Their average Kappa values remain consistently high, at 0.97 for I-VT and 0.95 for I-KF.

In contrast, I-HMM largely underperforms relative to the other algorithms, with F1 scores between 0.80 and 0.96 and Kappa values from 0.65 to 0.92, resulting in an average Kappa of 0.79. I-DT performs slightly better, achieving F1 scores between 0.86 and 0.97 and an average Kappa of 0.88, while TobiiVT reaches an average Kappa of 0.91.

Similar to the training dataset, the weakest results across all algorithms are observed in Task T4, where F1 scores fall between 0.80 and 0.97 and Kappa values drop to 0.65–0.93. Task T3 also shows below-average performance for I-HMM and I-DT, with Kappa values ranging from 0.67 to 0.88 and F1 scores from 0.82 to 0.94.

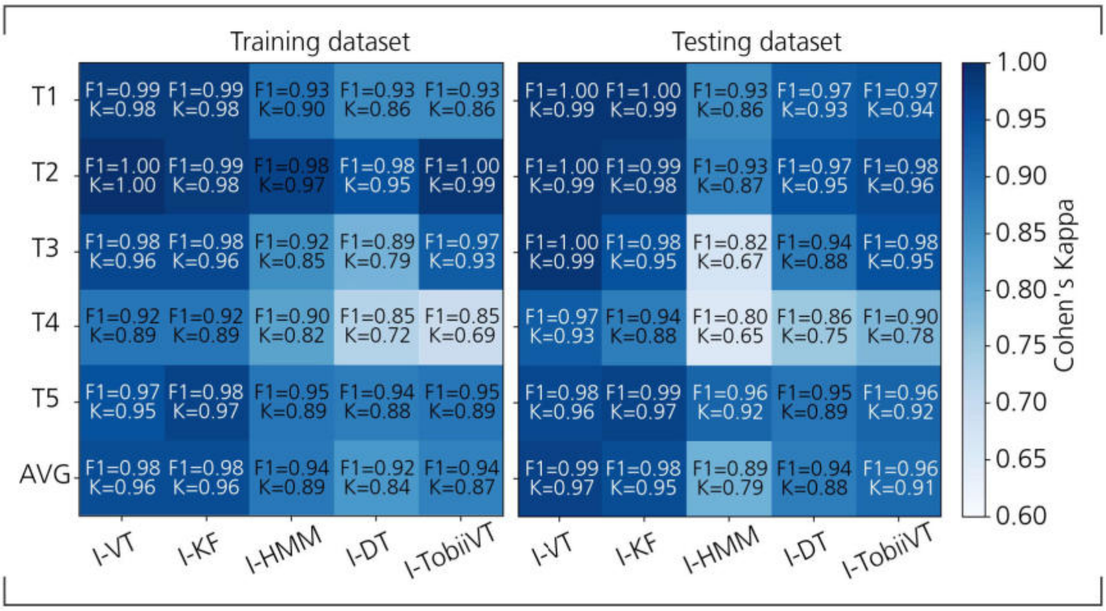


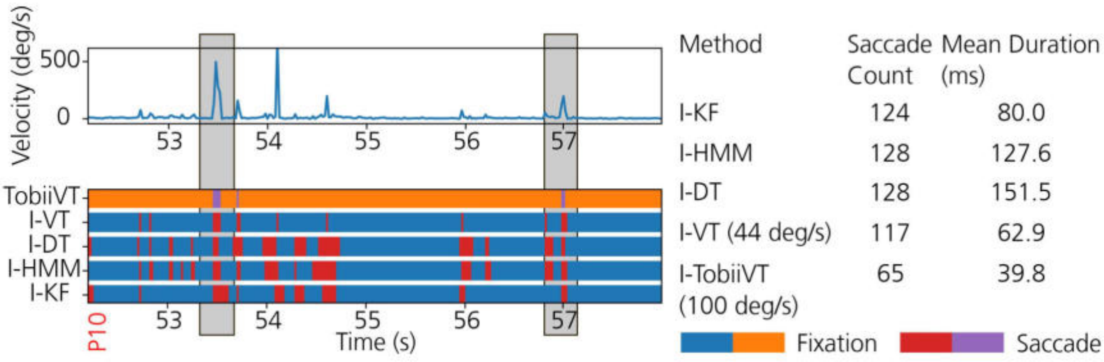
Figure 3.15.: **Highest F1- and Cohen's kappa Scores for the Training- (Left) and Testing Dataset (Right).**

3.3.2. Results from the Moving Point Task

In contrast to the jumping point tasks, the results of the moving point task are presented qualitatively. Figure 3.16 compares the detection outputs of all algorithms on this task.

The diagram on the left displays the detection outputs for participant 10. The top plot depicts angular velocity over time, while the timeline plot below illustrates the classifications of the five algorithms, with fixations marked in blue and saccades in red. The gray boxes indicate the intervals where a stimulus point jump occurs, during which a saccade is expected. In this example, the I-DT and I-HMM algorithms classify saccades more frequently outside these intervals and, overall, assign longer saccade durations compared to the other methods, particularly in contrast to the I-TobiiVT algorithm.

The table on the right quantifies these observations. It shows that I-HMM and I-DT have the highest saccade counts (128 each), significantly more than the I-VT (117), I-KF (124) and especially the I-TobiiVT (65). More notably, the table highlights a significant difference in mean saccade duration. The I-HMM and I-DT algorithms have the longest mean saccade durations, at 127.6 ms and 151.5 ms respectively, while I-VT (62.9 ms), I-KF (80 ms) and particularly TobiiVT (39.8 ms) produce values closer to typical saccade durations.



Method	Saccade Count	Mean Duration (ms)
I-KF	124	80.0
I-HMM	128	127.6
I-DT	128	151.5
I-VT (44 deg/s)	117	62.9
I-TobiiVT (100 deg/s)	65	39.8

Figure 3.16.: **Comparison of Algorithm Saccade Detection in Moving Point Task.**

3.4. Evaluation of Offline Eye Movement Event Detectors

First, the methodological constraints of the fixation–saccade invocation task study are examined to identify potential sources of bias and limitations in the generalizability of the results. Subsequently, the algorithms are evaluated with respect to their strengths and weaknesses in comparison to the baseline I-TobiiVT algorithm. This evaluation forms the basis for the final selection of the algorithms to be advanced for online application.

3.4.1. Limitations of the Fixation-Saccade Invocation Task Study

One apparent limitation of the fixation–saccade invocation task study is that it was performed exclusively in the 2PASD flight simulator, which is not representative of all simulator setups. Simulator setups can vary considerably in their layout and size, most notably in the distance between the pilot seat, the primary vision system and the flight instruments. Because this distance directly influences the differential visual angle calculated between gaze samples, optimized parameters, especially those dependent on this measure, may differ significantly across simulator environments.

Furthermore, although most algorithms converged toward an optimum, the precision of this convergence may have been constrained by the study's sample size. A larger invocation task study may have allowed for more accurate determination of the optimal parameters, particularly in the case of the I-HMM. This was the only algorithm for which a clear convergence was not evident, likely due to its large parameter set combined with the relatively small dataset. Evidence for this is provided by the comparison of kappa and F1 scores between the training and testing datasets in Figure 3.15, which differed more substantially for the I-HMM than for the other algorithms. This pattern strongly suggests overfitting to the training dataset.

Another important factor influencing the optimization process is the underlying assumption that all algorithms should perform equally well across the five distinct jumping point tasks, which were designed to represent typical eye movement patterns inside a helicopter. In real helicopter operations, however, some of these patterns occur more frequently than others. It can therefore be argued that the tasks should be weighted according to their frequent use. Such weighting would favor algorithms that perform strongly on the most common patterns, while allowing for weaker performance on less frequent ones.

The optimization process was also strongly influenced by the chosen penalization strategy. Because Cohen's kappa was used as the optimization objective, the penalization strategy directly affected the final hyperparameter selection. As the primary aim is to develop an algorithm optimized for fixation detection and since the ground truth labeling only allowed for reliable fixation identification, the fixation penalization strategy was designed in greater detail. Nevertheless, alternative strategies are possible, such as equally elaborate penalization for saccades or approaches that impose no additional penalty for over- or under-segmentation.

For the moving point task, angular velocity peaks were observed during the time windows in which saccades were expected, confirming that velocity peaks are associated with saccades. However, comparable peaks also appeared outside these windows as seen in Figure 3.16. If participants had followed the moving point strictly as instructed, the total number of saccades across all 11 participants should have ranged between 22 and 33, since the task was designed to elicit only 2–3 saccades per participant. In practice, far more saccades were recorded. These additional events may have resulted from eye-tracking noise, overshooting and corrective refixations or difficulties in maintaining a smooth pursuit, all of which can result to a velocity peak. Given that the observed saccade counts deviate significantly from the expected range and this deviation may in fact be due to true saccades, as suggested by the velocity peaks, no reliable quantitative analysis is possible.

Nevertheless, some qualitative insights can be drawn. The much longer saccade durations and higher counts reported by I-HMM and I-DT are likely false detections, given that typical saccade durations range between 30–80 ms and participants consistently rated the workload as low to moderate, which is consistent with shorter saccade durations [**zagermann2016**]. For the other algorithms, however, no precise judgment can be made, since participants may have produced more saccades than required and subjective workload reports provide only indirect evidence. This again highlights the limited qualitative evaluation and unsuitability of the moving point task for quantitative assessment.

3.4.2. Final Selection for Online Deployment

The performance of each algorithm is assessed across several criteria, including performance on the jumping point and moving point tasks, computational complexity, real-time applicability, and robustness. Figure 3.17 summarizes this qualitative evaluation for the I-VT, I-KF, I-DT, and I-HMM algorithms, using the I-TobiiVT algorithm as a baseline.

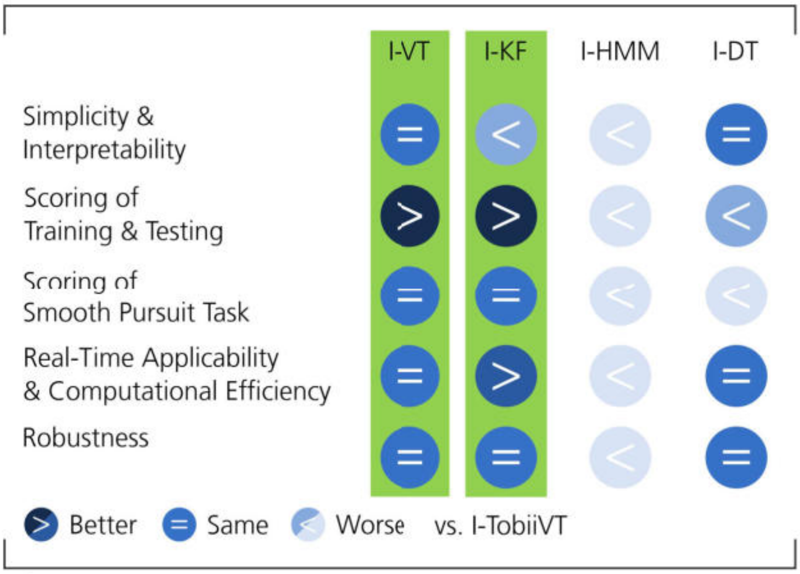
The I-VT algorithm performs almost identically to the I-TobiiVT baseline across all criteria, as both rely on the same detection method and apply a moving median smoothing filter. However, the optimized velocity threshold of 44 deg/s obtained through hyperparameter optimization improves its performance on the jumping point tasks compared to the baseline velocity threshold of 100 deg/s, while performance in the smooth pursuit condition of the moving point task remain similar.

The I-KF algorithm is a probability-based method, making it slightly more complex to set up and understand. In the jumping point tasks, the I-KF outperforms the baseline, whereas in the moving point task, its performance is comparable, although it detects more and longer saccades. Similarly to the I-VT, this is likely due to its higher sensitivity to changes in eye movement velocity. In addition, the I-KF's real-time applicability is slightly better than the I-TobiiVT, as its built-in smoothing filter avoids the one-sample delay associated with the moving median filter. Furthermore, its robustness is similar to the baseline due to the presence of its own smoothing step.

The I-DT algorithm, like the I-TobiiVT, is threshold-based and therefore simple to set up and interpret. Its performance on the jumping point tasks is slightly lower and it performs particularly poorly in the moving point task. This is likely due to its higher sensitivity to noise, a consequence of its low dispersion threshold, as reflected in the saccade frequency and duration results. In terms of computational efficiency and robustness, it is comparable to the baseline, since its two-sample duration threshold requires only a single-sample lookahead without the need for a smoothing filter.

The I-HMM algorithm performs worse than the I-TobiiVT across all evaluation categories. Its large number of parameters and complex detection method reduce interpretability, and it underperforms in both the jumping point and moving point tasks. A likely explanation is that the available data were insufficient for the algorithm to converge to the right optimal solution, leading to overfitting on the relatively small training dataset and resulting in a noise-sensitive configuration similar to that observed for the I-DT method. Moreover, a particular weakness of the I-HMM occurs when it initializes in the wrong state. Given its state probabilities (see Table 3.3), the model tends to persist in the current state, systematically mislabeling subsequent events by for instance classifying fixations as saccades or saccades as fixations, until sufficient contradictory evidence triggers a state correction. This behavior further reduces its robustness, especially when the model is not optimized well.

Based on the qualitative evaluation above, the I-KF and I-VT are selected for online deployment, as both generally perform better than the I-TobiiVT baseline. Between these two, no clearly superior option emerges, which is why both are retained for subsequent online verification. Conversely, the I-HMM and I-DT are excluded due to their generally lower performance to the baseline. Since the smooth pursuit task revealed significantly different detection behavior from the baseline, where the baseline suggested slightly better performance, the I-VT and I-KF parameter values were chosen more conservatively than the theoretical optimum. This reduces faulty fixation segmentations caused by false saccade classifications during smooth pursuit. In addition, parameter optimization for both algorithms indicates a performance plateau rather than a sharp single optimum, meaning that slightly more conservative values achieve nearly the same accuracy. Consequently, the final parameters applied are a velocity threshold of 55°/s for the I-VT, a chi-square threshold of 85, a q_{pos} of 0.002, and a q_{vel} of 0.25 for the I-KF.



	I-VT	I-KF	I-HMM	I-DT
Simplicity & Interpretability	=	<	<	=
Scoring of Training & Testing	>	>	<	<
Scoring of Smooth Pursuit Task	=	=	<	<
Real-Time Applicability & Computational Efficiency	=	>	<	=
Robustness	=	=	<	=

> Better = Same < Worse vs. I-TobiiVT

Figure 3.17.: **Comparison of Eye Movement Event Detection Algorithms Against the I-TobiiVT Baseline.**

4. Online Adaptation of an Eye Movement Event Detector in a Helicopter Simulator Study

This chapter presents the implementation and application of the online adaptation of an eye movement event detector in a helicopter simulator study. The chapter is structured as follows. First, the development of the selected detection algorithms into (near) real-time application is described. Next, the verification methodology for the online algorithms is described, outlining the simulator setup, mission profile, study procedure and evaluation method. The evaluation method used to verify the online algorithms includes a user survey, a pilot questionnaire and metrics for both algorithm performance and eye movement behavior. Finally, the results of the study are presented, including pilot feedback, workload assessments, algorithm performance and the relationship between eye movement metrics and subjective workload ratings.

4.1. Development of Online Event Detection Algorithms

The development of the online eye movement event detection algorithms builds upon the offline event detection algorithms introduced in the previous chapter by adapting its preprocessing, classification and postprocessing stages into a real-time processing pipeline. First, the overall algorithmic design is described, detailing the steps involved in transforming raw gaze data into classified eye movement events. Subsequently, the methods for storing the resulting data and visualizing detection outputs in near real time are presented. The implementation of both algorithms is available through DLR's internal GitLab repository.[1]

[1] The source code for the online I-VT and I-KF algorithms is available at: https://gitlab.dlr.de/ft-hub-eyetracking

4.1.1. Online Algorithm Design and Processing Pipeline

The online algorithms operate in a sequential workflow, where raw gaze data retrieved from the eye tracker is continuously transformed and processed. This enables the classification of eye movement events, which are then stored in an output file and visualized in near real time.

Figure 4.1 illustrates the data flow within this pipeline, beginning with eye-tracking data retrieval, followed by a three stage processing sequence for online event detection and concluding with data storage and real-time visualization of the classified events.

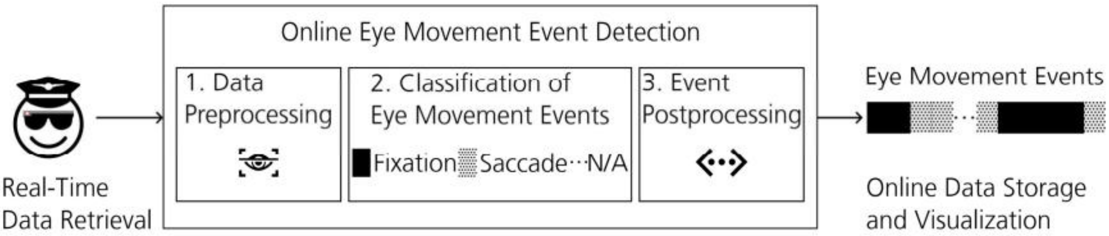

Online Eye Movement Event Detection
1. Data Preprocessing
2. Classification of Eye Movement Events
█ Fixation ▒ Saccade ··· N/A
3. Event Postprocessing
Eye Movement Events
Real-Time Data Retrieval
Online Data Storage and Visualization

Figure 4.1.: **Online Eye Movement Event Detection Pipeline.**

In general, both scripts are written in C# and connect to the eye tracker streaming server via a WebSocket to receive gaze data in real-time. Because gaze data are rendered continuously and arrive at irregular intervals, the scripts retrieve the data asynchronously by subscribing to the gaze stream with an initial request. This asynchronous approach allows the program to run continuously while simultaneously listening for new data from the WebSocket as it becomes available.

The WebSocket connection is initially established through an HTTP handshake. Once the handshake is complete, the connection is upgraded to a WebSocket, which remains open until explicitly closed by either the server or the user running the online algorithm. To prevent the WebSocket from closing automatically, a keep-alive loop is implemented, which sends periodic messages every four seconds. Termination of the WebSocket connection is handled via cancellation tokens to enable graceful shutdown when the user presses Ctrl+C or Enter.

Besides the keep-alive loop, another loop exists in each script, which implements the main processing logic. The main loop is responsible for processing and outputting data in near real time. It begins with data preprocessing, where the raw eye-tracking data are prepared. Following this, the processed data enter the sample-level eye movement classification stage, in which the algorithm categorizes each sample into one of three event types: Fixation, Saccade or N/A (Not Applicable). Finally, in the data postprocessing stage, the classified samples are further analyzed and formatted for output.

All three stages contribute to the overall latency from raw gaze data retrieval to the generation of postprocessed, classified samples, as illustrated in Figure 4.2. The timeline in the figure highlights the contribution of each stage to the total latency, showing how preprocessing, classification, and postprocessing cumulatively add to the delay. The exact implementation and impact of each stage on latency are discussed in detail in the following sections.

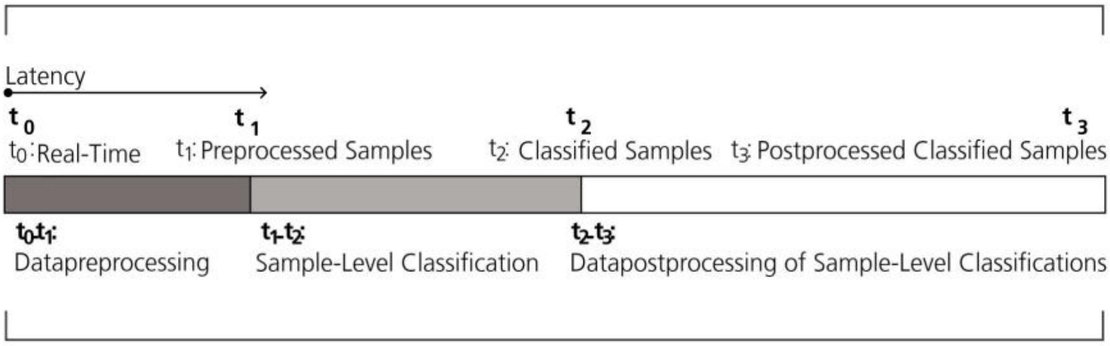

Latency
t_0
t_1
t_2
t_3
t_0: Real-Time
t_1: Preprocessed Samples
t_2: Classified Samples
t_3: Postprocessed Classified Samples
t_0-t_1:
Datapreprocessing
t_1-t_2:
Sample-Level Classification
t_2-t_3:
Datapostprocessing of Sample-Level Classifications

Figure 4.2.: **Latency Breakdown of Online Algorithms.**

Data Preprocessing

The preprocessing step is responsible for transforming the incoming eye-tracking data from the WebSocket into a format compatible for the subsequent event classification. Additionally, the preprocessing pipeline also includes noise reduction that is applied to the eye-tracking signal. The data received from the WebSocket not only contains gaze measurements but also messages related to the initialization and maintenance of the WebSocket connection. Non-gaze related messages are therefore filtered out beforehand to ensure that only relevant gaze samples are processed.

For the I-VT algorithm, the preprocessing first extracts timestamped left-eye and right-eye gaze direction vectors from each received gaze sample. If valid measurements are available from both eyes, an average gaze direction vector is computed as the mean of the left and right vectors.

To reduce high-frequency noise in the gaze data, a three sample moving median filter is applied, consistent with the approach used in the Tobii I-VT [**olsen2012**]. This filter computes the median for each component of the gaze vectors using the previous, current and subsequent sample. If fewer than three valid samples are available, the current sample is used as a fallback, ensuring that no data is lost during the denoising process.

Following denoising, the angular velocity of the gaze is calculated according to the method described in Section 2.2. Velocity is computed primarily from the average gaze vector, with fallback to the right-eye or left-eye vector if necessary.

Finally, the preprocessed sample of the I-VT, now containing median-denoised gaze directions and an estimated angular velocity, is forwarded to the main classification loop for event detection.

For the I-KF algorithm, a separate preprocessing step is not required, as the filter inherently performs both denoising and velocity estimation within its internal computations. Specifically, the Kalman filter maintains independent filter states for the average, left and right eye gaze directions to apply the same fallback logic as in the I-VT velocity computation. At each incoming sample, the filter updates these states using the current and previous gaze vectors, applying a linear Kalman prediction and correction step. The filtered gaze direction is taken directly from the filter's estimated position state, while the filtered velocity is derived from the estimated rate of change of the gaze position.

Consequently, the Kalman filter introduces no additional latency beyond the standard main loop operations, as the filtering occurs concurrently with the classification stage. In contrast, the I-VT preprocessing relies on a three sample moving median to denoise the gaze vectors, which results in a one sample delay in the pipeline. Given the sampling frequency of the Tobii Glasses 3, this equates to approximately 20 ms additional latency for each gaze sample before classification.

Classification of Eye Movement Events

Classification for both algorithms is performed according to their respective eye movement detection methods. For the I-VT and I-KF algorithms, the prescribed parameter values, as defined in the previous chapter, are used to classify each sample into either a fixation or a saccade. If no velocity can be computed for a sample, it is classified as N/A (Not Applicable). It is classified as N/A, as it is uncertain whether the inability to classify is due to a blink, an eye tracker related signal issue or signal loss e.g. when looking outside the eye-tracking frame.

Once each sample has been classified, the individual samples are grouped into eye movement events. This is achieved by comparing the type of each classified sample with the type of the previous sample. A new event is started whenever the sample type changes and the previous event is closed with its start and end timestamps recorded. Consequently, each event is defined by its type (fixation, saccade, or N/A) along with its temporal boundaries.

Event Postprocessing

For both the I-VT and I-KF algorithms, the event postprocessing step is identical. It is implemented to refine the initial classification of eye movement events by discarding short fixations and merging adjacent ones. The two rules, namely Discard Short Fixations and Merge Adjacent Fixations, are based on the same as the Tobii I-VT Attention Filter and use the same parameterization [**olsen2012**].

The Discard Short Fixations rule operates on a five sample sliding window. If the window contains fixation samples that are flanked by saccades on both sides and the total duration of the consecutive fixation samples within that window is shorter than 60 ms, the fixation is deemed false. All samples in that window are then reclassified as saccades.

The Merge Adjacent Fixations rule uses a six sample sliding window to detect short saccade runs occurring between two fixations. If the duration of such a saccade segment is shorter than 75 ms and the angular distance between the endpoints of the surrounding fixations is smaller than 0.5 °, the saccade samples are reclassified as fixations.

The refined output of this postprocessing step represents the final event classification results with the angular distance computed from the best available gaze direction vectors (average, right-eye, or left-eye), following the same approach as in the preprocessing step. Due to the use of a six sample window for the Merge Adjacent Fixations rule, both algorithms incur an additional latency of six samples, corresponding to approximately 120 ms. This is a substantial increase compared to the single sample latency added by the moving median filter and the previous lightweight computation for the preprocessing and event classification step.

4.1.2. Online Data Storage and Visualization

This section outlines how the classified eye-tracking data is stored in an output file for downstream use and how the data is displayed in the terminal to provide online visualization.

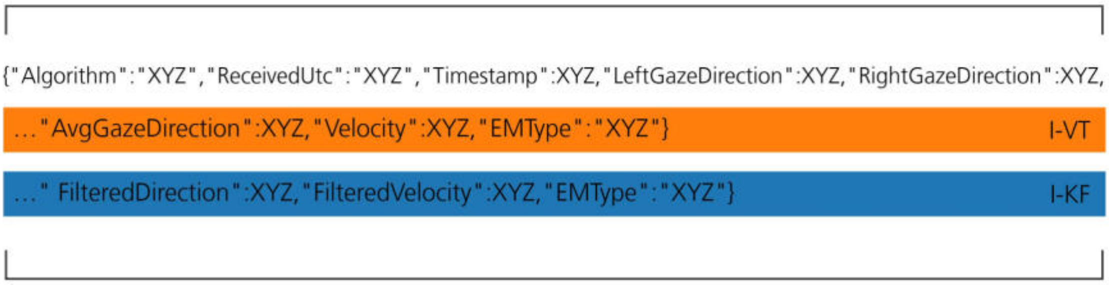
Figure 4.3 illustrates the JSONL output format of a single processed eye-tracking data sample for both the I-VT and I-KF algorithms.

The first five data fields are identical for both algorithms. The Algorithm field specifies whether the sample was processed using I-KF or I-VT, which facilitates downstream analysis or integration with other applications. The ReceivedUTC field provides the UTC (Coordinated Universal Time) timestamp of the sample and can, for example be used to synchronize the eye-tracking data with the eye tracker's live video stream at a later stage. The next three fields are Timestamp, LeftGazeDirection and RightGazeDirection and contain the raw gaze data received from the eye tracker server. These values form the basis of event detection and are stored explicitly so that the applied detection method can be backtraced and the data can be reused for more advanced eye movement analyses.

The next two fields differ between the algorithms. For I-VT, the fields are AvgGazeDirection and Velocity (see Figure 4.3, orange bar), representing the computed average gaze direction and angular velocity described in the preprocessing step. For I-KF, the corresponding fields are FilteredDirection and FilteredVelocity (see Figure 4.3, blue bar), which reflect the denoised estimates produced by the Kalman filter.

Both formats conclude with the same EMType field, which stores the final event classification, which is either Fixation, Saccade or N/A. As new gaze samples are processed, additional lines are appended to the JSONL file. This structure makes the output compatible with other online applications that require continuous access to both raw gaze data and eye movement classifications.

In addition to writing results to an output file, the script provides an online visualization of event statistics directly in the terminal, as shown in Figure 4.4. If the WebSocket connection is established successfully, the filter prints a confirmation message in the terminal and indicates that pressing Ctrl+C or Enter will terminate the script. Conversely, if the connection fails, a "failed to connect" message is displayed. The latency of each received sample is printed in the [RTLatency] row in milliseconds. Latency is calculated by timestamping both the moment a sample is first received UTC and the moment the received UTC value is written to the output file. Thus, the difference between these



```
{"Algorithm":"XYZ","ReceivedUtc":"XYZ","Timestamp":XYZ,"LeftGazeDirection":XYZ,"RightGazeDirection":XYZ,
..."AvgGazeDirection":XYZ,"Velocity":XYZ,"EMType":"XYZ"}
..." FilteredDirection":XYZ,"FilteredVelocity":XYZ,"EMType":"XYZ"}
```

Figure 4.3.: **JSONL Data Structure of Processed Eye-Tracking Samples for I-VT and I-KF.**

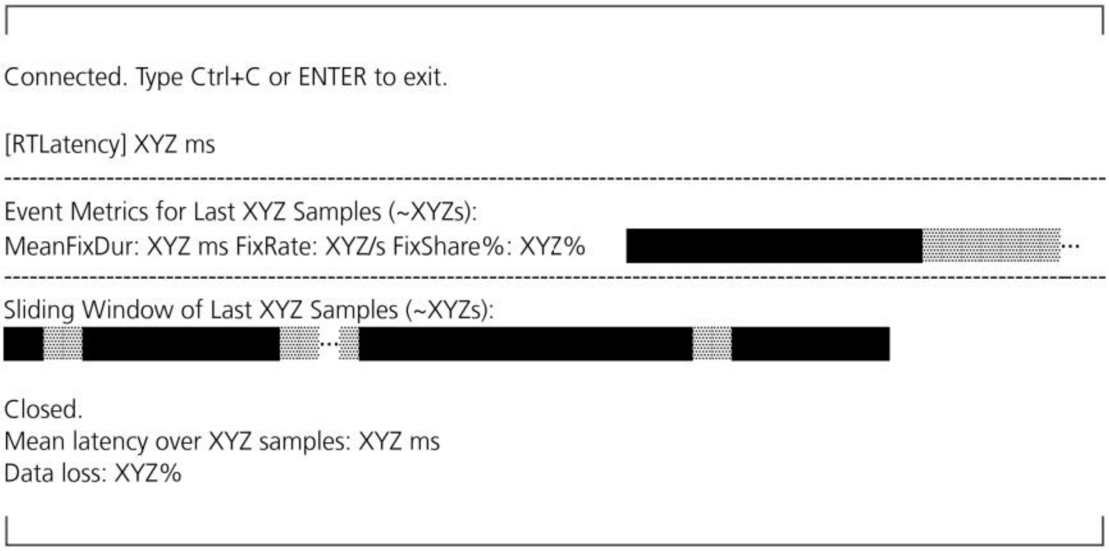
timestamps yields the sample latency.

The subsequent output is structured into two main sections that are continuously updated as new gaze samples are processed.

The first section summarizes the event metrics average fixation duration, the fixation rate (number of fixation per second) and the percentage of fixation samples relative to all detected eye movement events. These values are computed over an evaluation period defined by a fixed number of recent samples. To provide an intuitive overview, a horizontal bar chart is rendered, where fixations (■), saccades (□), and N/A events (·) are displayed in proportion to their relative share.

The second section is a sliding window of the a number of specified last samples and visualizes the sequence of recent eye movement classifications in temporal order. Here, the individual characters right indicating fixation (■), saccade (□), and N/A (·) correspond to classified events.

Finally, upon termination of the program, the terminal reports two additional metrics. The mean processing latency across all processed samples and the data loss calculated as the number of N/A samples divided by the total number of samples. This design allows the user to monitor eye movement events online without relying on additional plotting libraries or graphical interfaces, making the visualization computationally efficient and lightweight to implement. Moreover, the visualization can be customized by modifying the evaluation period for the event metrics or the sliding window, as well as both bar chart lengths, thereby allowing the terminal output to be tailored to different test setups or user preferences"



```
Connected. Type Ctrl+C or ENTER to exit.

[RTLatency] XYZ ms
--------------------------------------------------------------------------------
Event Metrics for Last XYZ Samples (~XYZs):
MeanFixDur: XYZ ms FixRate: XYZ/s FixShare%: XYZ%    ██████████████▒▒▒▒▒▒···
--------------------------------------------------------------------------------
Sliding Window of Last XYZ Samples (~XYZs):
██▒▒██████████▒▒···▒████████████████▒▒███████

Closed.
Mean latency over XYZ samples: XYZ ms
Data loss: XYZ%
```

Figure 4.4.: **Online Terminal Visualization of Eye Movement Events and Metrics.**

4.2. Methodology for Online Algorithm Verification

This section outlines the methodology used to verify the feasibility and performance of the online eye movement event detection algorithms in a helicopter simulator environment. The study was designed to ensure both ecological validity and reproducibility. First, the study setup is described, including the simulator configuration and pilot data. The following section presents the flight mission profile, which defines the flight maneuvers forming the basis of the study tasks. The study procedure is then outlined, covering pilot familiarization, data collection and evaluation protocols. Finally, the methods for online algorithm evaluation are introduced, including the applied metrics and data sources.

4.2.1. Study Setup

The study setup consists of the flight simulator, in which the experiment was conducted, and the equipment required for the execution of the online algorithms.

The AVES (Air Vehicle Simulator) is DLR's research flight simulator, designed among other purposes for flight test preparation and special mission training of DLR's flying test bed EC135 ACT/FHS (Active Control Technology/Flying Helicopter Simulator). It comprises a motion platform with interchangeable cockpits that can be operated in either motion or fixed-base mode [**holger2013**]. The cockpit used in this study is a replica of the ACT/FHS cockpit. The ACT/FHS itself is DLR's highly modified EC135 equipped with a full-authority fly-by-light control system and an experimental flight control system [**kaletka2005**]. To model the flight dynamics of the ACT/FHS in the simulator, HeliWorX was used, which is based on the real-time simulation model SIMH. The ACT/FHS replica cockpit inside the AVES, together with the dome visual projection, is shown in Figure 4.5 [**strbac2022**, **hamers1997**].

The dome visual projection of AVES provides a horizontal field of view from –120° to +120° and a vertical field of view from –53° to +40°. The measured resolution is approximately 5 arc-min per optical line pair (OLP), indicating the finest detail the system can resolve. Its brightness is 13.5 cd/m^2 (candelas per square meter) with a contrast ratio of 7:1 [**holger2013**].

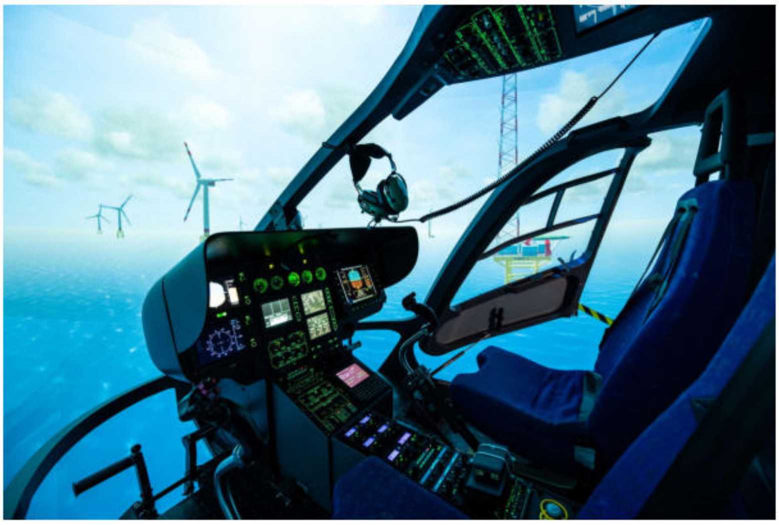


Figure 4.5.: **AVES ACT/FHS Cockpit with Dome Projection of a Maritime Offshore Wind Farm.**[2]

The equipment used for the verification study in the AVES simulator included the Tobii Glasses 3, as previously introduced and described in Chapter 3 and an off-the-shelf laptop on which the online algorithms were executed. The laptop operated independently of the AVES simulator environment. Its specifications are summarized in Table 4.1. No special hardware requirements were imposed, as the algorithms are designed to perform reliably on any standard off-the-shelf laptop.

[2] Source: DLR (CC BY-NC-ND 3.0)

Table 4.1.: **Laptop Specifications**

Component	Specification
Processor (CPU)	Intel(R) Core(TM) i7-10510U
CPU Speed	1.80 GHz (Base) / 4.90 GHz (Boost)
CPU Cores / Threads	4 Cores / 8 Threads
Installed RAM	16 GB DDR4
Storage Type	SSD (Solid State Drive)
Graphics Card (GPU)	Intel(R) UHD Graphics (Integrated)
Operating System (OS)	64-bit Operating System, x64-based processor

The pilot data for the simulator study are summarized in Table 4.2. The participant is not a certified test pilot but has accumulated a total of 105 total flight hours, with no prior experience on the EC135 helicopter type. Notably, the pilot is familiar with research flight testing procedures, which is relevant as such experience increases a pilot's familiarity with flight maneuvers outside routine mission tasks and with structured test procedures.

The participant uses visual aids, which are worn via corrective lenses integrated into the Tobii Glasses 3 eye tracker that match the pilot's prescription. Additionally, the pilot has blue eyes, a feature that may affect eye-tracking data quality.

Table 4.2.: **Pilot Data Summary**

Parameter	Value
Test Pilot (Y/N)	N
Flight Hours, Total	105
Flight Hours, EC135	0
Familiar with Research Flight Tests (Y/N)	Y

4.2.2. Flight Mission Profile of Study

The simulated mission replicates an offshore hoist operation in the Alpha Ventus offshore wind park, located off the North Sea coast of Lower Saxony, using DLR's research helicopter ACT/FHS. Although the ACT/FHS is heavily modified from the original EC135, it remains representative of light utility helicopters typically employed in offshore wind farm operations [**strbac2022**]. The Alpha Ventus wind farm serves as a research field for multiple institutions, providing opportunities to test offshore flights and highlighting both the current demand and future prospects of helicopter operations in such environments[3].

[3] https://www.alpha-ventus.de/forschung (accessed: 26 August 2025)

Simulator flights in the wind farm have already been conducted using the AVES simulator, confirming its suitability for this study.

An additional advantage of this environment is the increased pilot workload, as visual references in wind farm settings are generally limited. This characteristic enables a study design in which pilot workload can be varied by adjusting the environmental visibility range, which acts as the primary visual reference [**lehmann2017**].

The simulated offshore hoist operation does not incorporate global wind effects. An overview of its four distinct flight phases is given in Figure 4.6. It involves (1.) a helipad departure with an initial confined-space hover and backward flight to the Take-Off Decision Point (TDP), (2.) an approach to a wind turbine while maintaining stable airspeed and altitude, (3.) a stabilized hover over a hoist target under temporal and positional constraints, and (4.) a repositioning to a second turbine with controlled acceleration, deceleration and hover stabilization. Collectively, these maneuvers cover a broad spectrum of flight tasks. Consequently, both the visibility range and flight control inputs are varied in the study to elicit a wide range of workload demands.

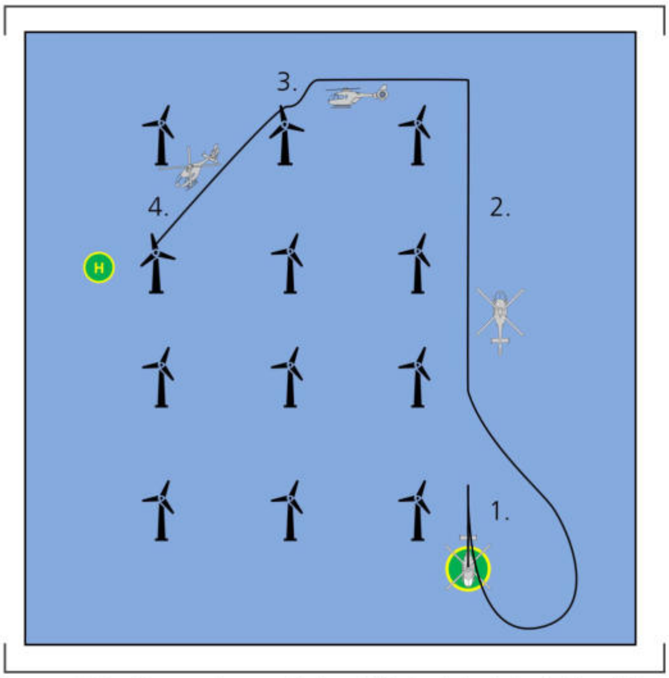


Figure 4.6.: **Overview of the Flight Task in Plan View.**

The mission begins with a takeoff specifically designed for departures from a helipad. The maneuver follows the Airbus Helicopter EC135 flight manual and is illustrated in

Figure 4.7. Initially, the pilot establishes a hover in ground effect (HIGE) at 4+15 ft, followed by rearward motion of no more than 3+6 m. From this position, the helicopter is flown backward to the TDP at 120±30 ft above helipad elevation (AHE). At the TDP, the helicopter transitions into forward flight, accelerating to a target speed of 65±20 knots.

The EC135 flight manual specifies boundaries for climb power upon reaching the TDP. However, these constraints are neglected in the simulation to simplify pilot instructions and task training. Additionally, the tolerances for the flight limits were formulated collaboratively with the pilot prior to the study, as the EC135 manual does not provide explicit tolerance values.

Performance - Helipad T/O	Flight Limits
Hover with HIGE at x ft height and x m rearward movement	4+15; 3+6
Fly backward to TDP of x ft AHE	120±30
Transition to forward flight and accelerate to x kts	65±20

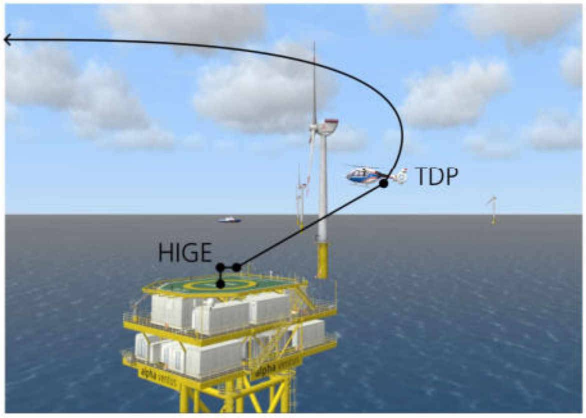


Figure 4.7.: **Helipad Take-Off Performance Table and Corresponding Flight Maneuver Illustration.**

After departure, the pilot performs a controlled approach to the first wind turbine, as illustrated in Figure 4.8. The maneuver begins with an almost 180° left turn to align the helicopter with the intended flight path. The helicopter then flies alongside the wind park, which is positioned on its left side. Throughout this segment, a steady airspeed of 100±10 knots and an altitude of 300±30 ft, approximately corresponding to the rotor hub height, must be maintained. As the helicopter nears the final wind turbine, it executes another left yaw to approach the target turbine. The target is identified among the idle turbines, of which there are two and of which the nearest idle turbine serves as the designated approach location.

Flight tolerances were derived from the Mission Task Elements (MTE), which provide a standardized reference for establishing operational boundaries. MTEs are widely used in military aviation to evaluate an aircraft's performance and handling qualities, making them a suitable basis for defining tolerances in this study.[**usa2000**]

Performance - Approach to Wind Turbine	**Flight Limits**
Maintain an airspeed of x knots throughout the course	100±10
Accomplish maneuver at an altitude of x ft	300±30

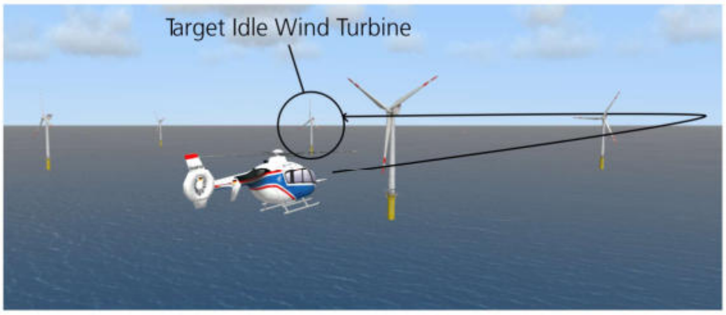


Figure 4.8.: **Approach to Wind Turbine Performance Table and Corresponding Flight Maneuver Illustration.**

Upon reaching the wind turbine, the pilot transitions into a stabilized hover adjacent to the hoist platform. This hover must be maintained for at least 30 seconds, adhering to an altitude of 330±15 ft and a heading of 275°±5°, as illustrated in Figure 4.9. This maneuver represents the operationally critical hoisting phase, during which the helicopter must remain sufficiently stable to transfer personnel or equipment. Maintaining this hover requires precise control, making it the presumed most challenging segment of the task.

The hover location is positioned next to the wind turbine rather than directly above the hoist platform to simplify pilot instructions, as no strict positional constraints were defined. All other limits and tolerances were established collaboratively with the pilot prior to the study.

After completing the hoist at the first turbine, the helicopter departs and accelerates to 60 ± 10 knots while transitioning toward the second, idle turbine as depicted in Figure 4.10. On arrival, the helicopter is decelerated and repositioned into a hover at 330 ± 15 ft with a heading of 275° ± 5°. This final maneuver replicates the approach and hover to the first turbine with the difference of adding rapid acceleration and deceleration segment based on a MTE to strain the pilot with further workload while repeating precision hover and positioning task.

Performance - Establish Hover Position next to Hoist Target	**Flight Limits**
Maintain a stabilized hover for at least x seconds	30
Maintain heading of x ft	330±15
Maintain heading of x degree	275±5

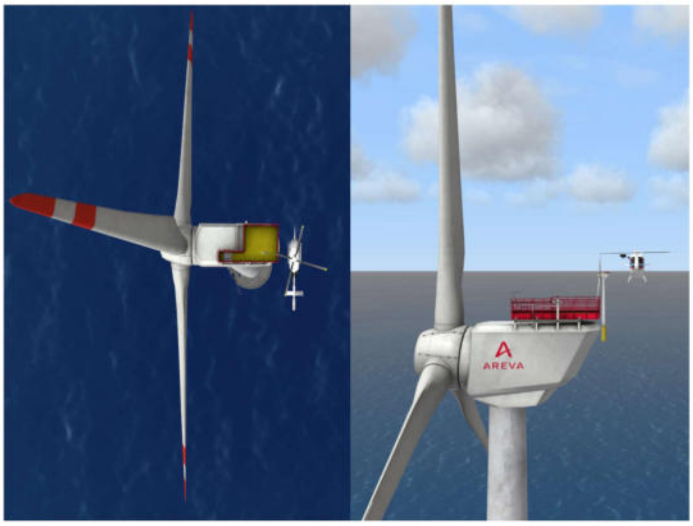


Figure 4.9.: **Stabilized Hover next to the Hoist Target Performance Table and Corresponding Flight Maneuver Illustration.**

4.2.3. Study Procedure

The study procedure follows the flow chart in Figure 4.11.

Before the simulator study starts, informed consent is obtained. The pilot is then briefed on the study objectives, the overall timeline and the sequence of flight tasks. Performance limits and completion criteria are reviewed (see Section 4.2.2).

Afterwards, the eye tracker is calibrated to the pilot's eyes.

Next, the test point is conducted. Successful execution of the test point is supported by allowing the pilot to become familiar with the simulation environment and the flight task using the "long-look" evaluation technique described in **dod1990**. Following this technique, two hot runs are guaranteed, although additional runs may be performed at the pilot's request. During each run, whether familiarization or evaluation, pilot feedback and comments are recorded. This helps reduce variance in performance due to unfamiliarity and enables thorough identification of deficiencies such as handling quality issues, undesirable aircraft behavior or discomfort with the eye tracker.

Performance - Depart and Arrive at Final Wind Turbine	**Flight Limits**
Accelerate after departure to x kts and decelerate when arriving at the other idle wind turbine	60±10
When arriving at final wind turbine:	
Maintain altitude of x ft	330±15
Maintain heading of x degree	275±5

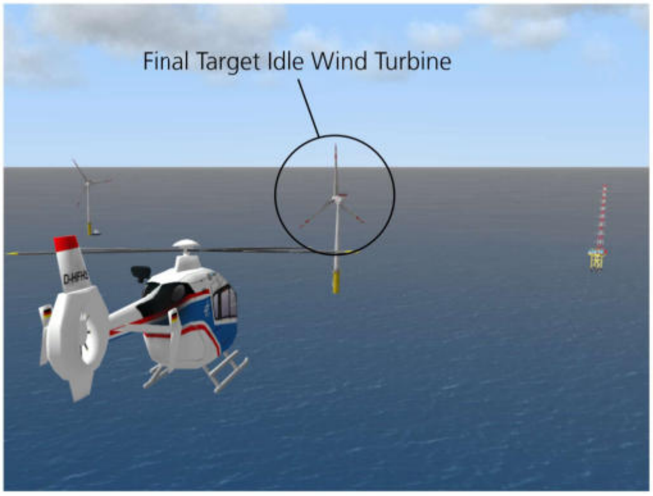


Figure 4.10.: **Departure and Arrival at Second Idle Turbine Performance Table and Corresponding Flight Maneuver Illustration.**

The predefined maneuvers are flown as described earlier, adhering to the stated performance limits. Eye-tracking data, simulator parameters, and pilot control inputs are recorded continuously. After each test point, a questionnaire is administered, which is described later in more detail. This process is repeated until all task points in the test matrix are completed. Finally, once all task points are finished, pilot data as specified in Table 4.2 are collected.

The test points flown depend on the evaluation conditions defined by the active flight control laws and the visual environment. Two distinct control system configurations are tested. The first employs only a basic Stability Augmentation System (SAS) in pitch, roll, and yaw (SAS 3AXES). The second employs a more advanced setup, with Attitude Control in pitch and roll, SAS only in yaw and Direct Control for the collective (ACpr SASy DIC) [**faa2019**].

Each control configuration is evaluated under two levels of visual environmental conditions, as shown in Figure 4.12, depicting the differences in external visibility. The Good Visual Environment (GVE, left in Figure 4.12) corresponds to a visibility range of 40,000 m. In contrast, the Degraded Visual Environment (DVE, right in Figure 4.12) corresponds to a visibility range of 2,500 m. The latter is expected to increase pilot workload and task difficulty, as the absence of a clear horizon complicates spatial orientation.

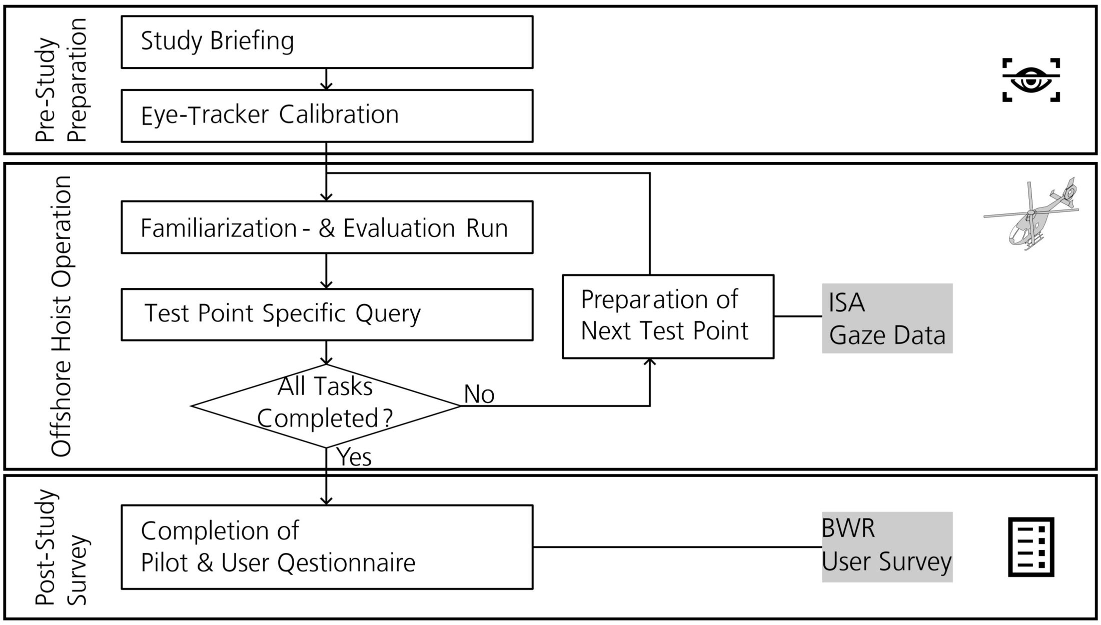


Figure 4.11.: **Overview of the Study Procedure.**

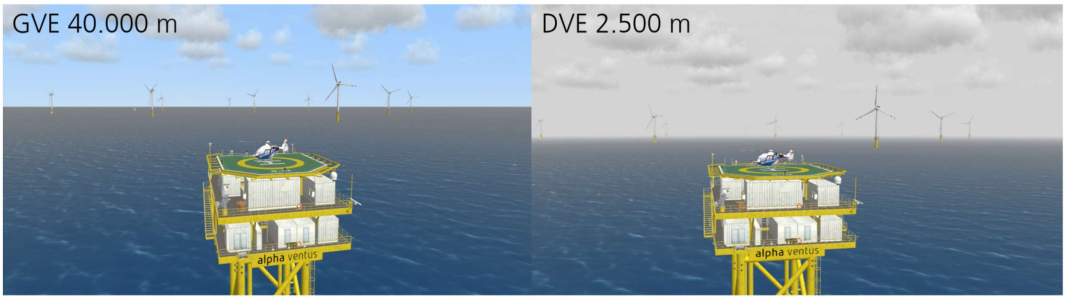


Figure 4.12.: **Visual Environments Used in the Study: GVE (40,000 m) on the left and DVE (2,500 m) on the right.**

This results in a 2 × 2 factorial matrix with four test points (TP01–TP04) presented in Table 4.3. A latin squared test matrix is not needed in this case as familiarization runs are conducted, which already eliminate variability due to the pilot's learning curve.

Table 4.3.: **Test Matrix for Offshore Hoist Operation**

Visual Environment	SAS 3AXES	ACpr–SASy–DIC
GVE (40,000 m)	TP01	TP02
DVE (2,500 m)	TP03	TP04

4.2.4. Evaluation of the Online Algorithms

To verify the applicability of the online eye movement detection algorithm in a flight simulator environment, an evaluation framework is designed. This chapter defines the metrics and data sources used to assess both the feasibility and performance of the algorithm.

The evaluation method comprises subjective- and objective measures. Subjective measures consist of pilot workload ratings, task-specific questionnaire items and the post-study user survey. Objective measures include eye movement metrics derived from the online detection algorithms as well as output data from both the scripts. The following sections provide a detailed description of each measurement method.

User Survey

The user survey is developed to specifically assess the usability, clarity and practical applicability of the online eye movement event detection scripts. The purpose of this questionnaire is to collect qualitative feedback from the user regarding the system's output format, interpretability and configuration options. This survey will be filled out by an DLR internal expert on gaze behavior analysis, who will operate the algorithms during the study, so that the survey contributes to the evaluation of whether the detection tool is suitable for a (near) real-time use in a flight simulator. The survey is appended in Appendix A and consists of five sections.

In the first section it is evaluated how easily the console output can be read in real time, whether metrics are self-explanatory and whether visual aids such as sliding windows, color coding, or symbols improve interpretability. After that, the user's perception of system latency and performance, including whether any delays interfered with real-time monitoring and under which circumstances the algorithm might misclassify events are surveyed. The next section examines the accessibility of adjustable settings (e.g., bar width, history length, post-processing toggle) and whether alternative configuration approaches such as command-line parameters or configuration files would be preferred.

Then, the tool's applicability in live flight testing, including whether synchronization with live scene video playback would be needed for future use cases are addressed. Finally, the user assesses the overall experience, which provides space for open comments on confusing aspects, preferred features and suggestions for improvements to usability.

Pilot Questionnaire

To assess pilot workload, a structured questionnaire is administered during the simulator campaign. The questionnaire combines standardized workload rating scales with task-specific evaluation items, thereby capturing both quantitative and qualitative aspects of pilot performance and perception. The complete test card, including the questionnaire, is provided in Appendix B. Each test point includes an Instantaneous Self-Assessment (ISA) [**kirwan1997**], Bedford Workload Rating (BWR) [**roscoe1990**] and task specific evaluation.

For the ISA, the pilot provides workload ratings at fixed 15-second intervals using the ISA scale from **kirwan1997** (see Appendix B). In this approach, ratings range from 1 to 5, where 1 corresponds to the lowest workload demand and 5 corresponds to the highest workload demand. This interval-based method allows the construction of a workload profile over time, enabling analysis of workload dynamics across test points. The ISA data can be compared to the workload classifications based on the online algorithms to evaluate whether algorithm-derived workload estimates correspond to pilot-reported workload at specific moments.

The Bedford Workload Rating Scale serves as a second subjective measure and is illustrated in the flight test card (Appendix B). Ratings follow the scale developed by **roscoe1990**, ranging from 1 to 10 and are divided into four categories: Satisfactory (1–3), Tolerable (4–6), High (7–9), and Intolerable (10) workload. The pilot assigns these ratings after each task for both the approach to the wind turbine phase and the flight phase perceived as most critical. These two flight phases provide complementary insights. The approach phase, assumed to be the least demanding, helps identify potential learning effects across test points, while the most critical phase highlights the workload peak and allows direct comparison with gaze-derived workload estimates.

Unlike ISA, which focuses on perceived intensity, Bedford ratings explicitly account for pilot spare capacity and the operational feasibility of continuing the task. Thus, the Bedford scale complements ISA by adding an operational perspective to workload assessment. Moreover, it was selected over the NASA Task Load Index, as the Bedford scale is explicitly recommended for verification tasks [**nasa2019**].

Additional questions address practical aspects of the scenario, which are the visibility during the approach, manageability of the flight control settings and potential interference of the eye tracker with visual perception, and are answered using a 1–5 Likert scale. An additional space for comments allows the pilot to report deficiencies, difficulties or other observations that might not be reflected in the rating scales. Finally, after completion of all flight test points, a post-study questionnaire collects pilot demographic data, as these can also influence flight behavior, workload and consequently gaze behavior during the experiment [**haslbeck2014**, **greiwe2023a**].

Algorithm Performance- and Eye Movement Metrics

For the final evaluation of both algorithms, performance metrics are defined. In particular, eye movement metrics are used to evaluate whether the online event detectors yield results consistent with the subjective ratings obtained from the pilot questionnaire.

To determine the real-time capability of both algorithms, the mean output latency is recorded across all test points. As the scripts already compute and display mean latency upon termination (see Section 4.1), these values are systematically noted after each run.

A further metric addresses the share of samples that are not classified as eye movements. The percentage of “N/A” classifications in the algorithm output is compared against the percentage of samples where the eye tracker provides no gaze direction and only monocular gaze. Ideally, the no gaze direction and algorithm N/A percentages should be equal, ensuring that “N/A” classifications are only assigned when the eye tracker itself provides insufficient gaze data. Contrary, If the N/A percentages are worse than the percentages of monocular gaze, the algorithm is deemed unsuitable for reliable eye movement detection.

Another important factor is data integrity, which examines whether all incoming eye tracker samples are processed and output by the algorithms. If the incoming data rate exceeds the processing rate, buffer overflow occurs, leading to dropped samples. This can also happen in case of CPU overload, where computational demand exceeds hardware capacity. To check for possible data loss, the number of samples recorded by the Tobii API logged on the SD card is compared to the number of samples processed by the algorithms. Since Tobii API data use different timestamps and are not given in UTC, exact synchronization of start and end samples is not feasible. Instead, mean sampling frequency is calculated for both data sources. Matching mean frequencies indicate that no data loss occurs in the algorithm output despite potential timing misalignments.

Finally, algorithm accuracy is evaluated using eye movement metrics displayed in the real-time terminal output. Prior literature indicates that mean fixation duration typically ranges from 200–300 ms and that saccade rate averages 3–5 per second (see Section 2.2). From this, two verification criteria are derived. First, fixation durations and fixation rates reported by both algorithms should in general approximate these reference ranges. Large deviations would indicate unreliable classification. Second, for the most cognitively demanding flight phases, mean fixation duration should increase and fixation rate should decrease, reflecting the well-established relationship between higher cognitive workload resulting in longer fixations and lower fixation rates (see Section 2.3.2).

Because the algorithms are designed for online application, both metrics are calculated using a moving time window. Following previous studies described in Section 2.3.2, a 10-second window is chosen to balance temporal resolution with classification reliability. Within this window, fixation duration and fixation rate are computed and compared against subjective workload ratings from the pilot questionnaire. This cross-validation of subjective and objective measures allows verification of whether the algorithms are sufficiently accurate for real-time eye movement event detection.

Specifically, for the BWR, comparisons are made qualitatively, while for the ISA ratings a standardized quantitative approach is applied. For the quantitative approach, the mean fixation duration and fixation rate values are extracted every 15 seconds and aligned with the ISA ratings provided at the same temporal resolution. Correlation analyses are then performed using scatter plots with linear regression lines and Pearson's r values. The scatter plots help visualize trends between ISA scores and eye movement metrics, which are further highlighted by the regression lines.s. Additonally, the Pearson's r quantifies the strength and direction of a linear relationship between two variables, ranging from -1 (perfect negative correlation) to $+1$ (perfect positive correlation), with values near 0 indicating no linear relationship [**duchowski2017**].

4.3. Results of the Helicopter Flight Simulator Study

The findings are structured along the evaluation framework defined in Section 4.2.4. First, subjective measures are reported, including the responses to the task-specific questionnaire and user survey as well as pilot workload ratings from the ISA and Bedford scales. Second, objective results from the online event detection algorithms are presented, which covers latency, data integrity and eye movement metrics such as fixation duration and fixation rate.

4.3.1. Survey Results and Pilot Feedback

The user survey indicates that the online algorithm application is generally well-received, with the user finding the console output clear, metrics intuitive and the sliding window effective for identifying trends. The latency display and UTC timestamps were rated as helpful while particularly the tool's sliding window output in the terminal was highly valued. However, the survey also highlighted key areas for improvement. The user noted the need to synchronize the output with live scene video playback for future applications. Additionally, the user also found the current configuration settings difficult to locate and modify. Overall, the tool was considered suitable for potential use in live flight tests.

The pilot feedback summarizing the pilot's ratings of visibility, control manageability and eye tracker comfort across the four test points is displayed in Figure 4.13.

Visibility was consistently rated high across all test points, showing that external visual cues remained sufficient while being slightly lower under DVE conditions. Control manageability was rated lower at 3 in TP01 and TP02, but improved to 4 in TP03 and TP04, indicating greater ease of handling with task repetition.

Eye tracker comfort was rated lowest at 3 in TP01, where the device was initially worn without a headband and tended to slip during maneuvers. After the headband was introduced, comfort improved to 4 and remained stable for TP02–TP04.

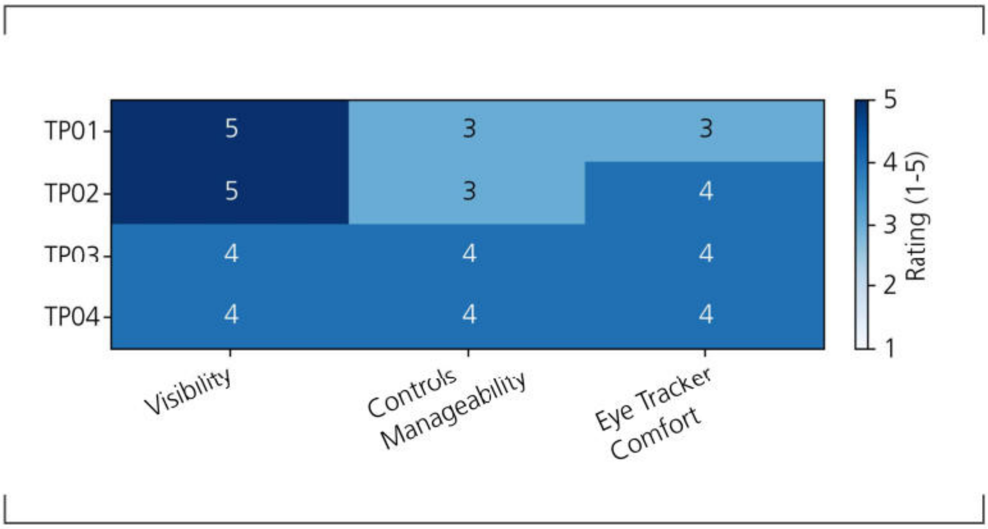


Figure 4.13.: **Pilot Ratings of Visibility, Control Manageability and Eye Tracker Comfort for Each Test Point (TP01–TP04).**

4.3.2. Subjective Workload Assessment

This section presents the pilot's ISA and BWR subjective workload assessments during the simulator study.

Figure 4.14 shows the results of the pilot's ISA workload ratings over time for all four test points. Across all test points, the workload profile shows considerable fluctuations, which reflect varying task demands during different flight phases.

For TP01, ratings at T/O phase start at a moderately high level (4), decrease to lower levels (2) during the mid-phase and then rise sharply to peak values of 5 during the first hovering phase next to the wind turbine around 4:00–4:15 minutes. After dropping again, workload briefly increases to 5 once more when stabilizing the helicopter to the final hover condition on the last wind turbine at approximately 5:30 minutes.

In TP02, high workload ratings (4–5) are reported within the first 45 seconds of T/O but quickly fall to around 1–2 during the approach to the wind turbine. In the later stages, at around 4:00 minutes, workload rises again to 4–5 during hovering next to the first wind turbine and later increases from 2 to 4 when stabilizing the helicopter at the final turbine at about 5:45 minutes.

The degraded visual environment is first flown in TP03 and produces an initial short-term high workload of 4 at T/O, which sharply drops to 1 and then fluctuates between 2 and 3 during the approach to the wind turbine. At 3:45 minutes, workload reaches its peak value of 5. In the final phases, ratings gradually rise again, reaching 4 at 6:15 minutes.

The final test point, TP04, the T/O begins with high workload ratings (4–5) during the first 30 seconds before decreasing sharply to the lowest level (1) during the mid-flight segments. Toward the later stages, ratings gradually increase again to 2–3 during the hovering phase next to the first wind turbine, drop briefly at around 5:00 minutes and finally peak at 4 at approximately 6:00 minutes during the final stabilization phase at the last wind turbine.

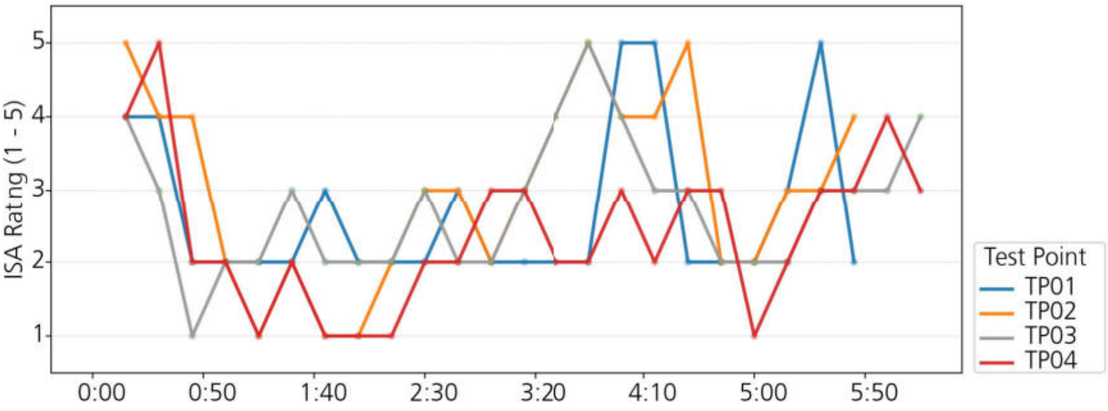


Figure 4.14.: **ISA Ratings over 15 s Intervals for Each Test Point (TP01–TP04).**

Figure 4.15 summarizes the Bedford ratings for each test point, reported separately for the approach to the wind turbine and for the most critical phase of the task, defined as the stabilization phase in which the pilot attempts to establish the hover within the predefined conditions next to the first wind turbine.

For TP01, the approach is rated 3, indicating low to moderate workload with ample spare capacity. The most critical phase is rated 6, reflecting high workload with limited spare capacity.

In TP02, the approach receives the lowest rating of 1, denoting very low workload and substantial spare capacity. By contrast, the most critical phase is rated 7, corresponding to very high workload with only minimal spare capacity remaining.

For TP03, the approach is rated 2, consistent with low workload, while the most critical phase is rated 6, indicating high workload comparable to TP01.

Finally, TP04 shows the lowest approach rating same as TP02 (1), yet the most critical phase reaches 8, which is the highest workload rating observed across all conditions, suggesting performance close to the limits of manageability.

Overall, all test points fall within the satisfactory workload range (1–3) during the approach phase. In contrast, the most critical stabilization phase differentiates between conditions. TP01 and TP03 remain within the tolerable workload bracket (4–6), while TP02 and TP04 escalate into the lower bracket (7–9), where task completion is possible but workload is intolerable.

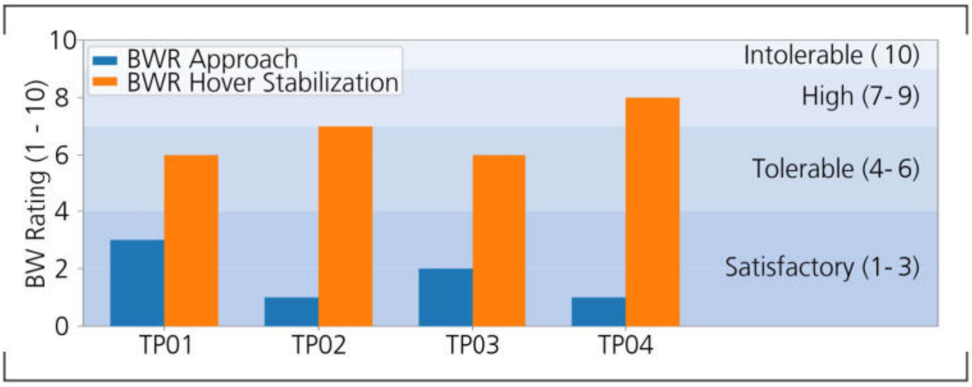


Figure 4.15.: **Bedford Ratings per Test Point (TP01–TP04) for Approach- and Critical Phase.**

4.3.3. Algorithm Performance- and Eye Movement Metrics

First, the algorithm performance metrics are presented, which quantify the proportion of missing data, system latency and unclassified eye-tracking samples. In the following section, the eye movement metrics that relate to workload demand will be reported.

Algorithm Performance Metrics

Figure 4.16 summarizes the results for the two evaluated algorithms, I-VT (orange) and I-KF (blue), as well as the raw eye-tracking data (green). All values are averaged across all four test points.

The first bar group displays the average percentage of samples in which data from both eyes were missing, while the second bar group shows the percentage of samples where data from either the left or right eye was unavailable. Both algorithms exhibit an equal share of unclassified data samples (3.8%), which is close to the 2.3% of fully missing samples in the raw data. This contrasts with the much higher 16.8% of single-eye data loss observed in the raw recordings.

The third bar group reports the effective sampling frequency. Both algorithms reproduce the raw sampling rate of 49.91 Hz, which is consistent with the Tobii eye tracker's nominal sampling rate of 50 Hz.

The latency results, shown in the last bar group, represent the only metric where a clear difference between the two algorithms was observed. I-VT yielded an average latency of 140.3 ms, whereas I-KF achieved a lower latency of 120.2 ms. Thus, I-KF not only provides reduced overall latency but also exhibits a shorter computational processing time, with 0.2 ms compared to 0.3 ms for I-VT.

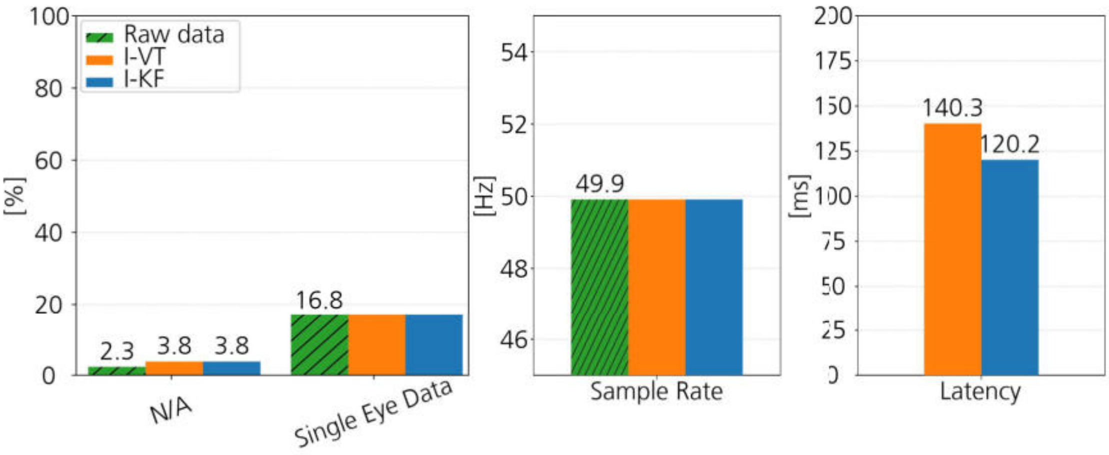


Figure 4.16.: **Comparison of Raw Data, I-VT, and I-KF across Performance Metrics.**

Eye Movement Metrics

The eye movement metrics are presented in Figure 4.17. The graphs depict the results for the four test points (TP01–TP04) over the flight test period, expressed in minutes and seconds. One figure displays the results derived from the I-VT event detection method, while the other shows the corresponding I-KF filter results. In each figure, the top graph illustrates the mean fixation duration (ms), and the bottom graph illustrates the fixation rate (fix/s). Both metrics were calculated using a 10-second sliding window that advanced with every new sample timestamp, providing a continuous time series of these metrics rather than being based on fixed 10-second interval computations.

Both figures reveal broadly similar trends across the four test points. Mean fixation duration remained relatively stable during cruise phases, typically ranging between 200 and 400 ms from approximately 0:50 to 2:40 min. Distinct mean fixation duration peaks were observed during hover and final approach, around 2:40–5:00 min and 5:10–6:25 min, respectively, which were consistent across test points. Similarly, fixation rate tended to slightly decrease during these phases. In particular, a prominent outlier was identified in the final approach, where the maximum fixation duration reached 2.0 s (I-KF) and 1.6 s (I-VT) and fixation rates 0.5 and 0.7 fix/s,respectively. In contrast, during takeoff, the mean fixation duration showed less pronounced deviations from the cruise baseline for the first 50 seconds compared to hover and final approach, although it exhibited higher fluctuations. The fixation rate, however, remained very low, close to zero, and increased steadily over time toward the rates observed during the cruise phase.

4.3.4. Correlation of Eye Movement Metrics and Workload Ratings

First, the relationship between the ISA workload ratings and eye movement metrics are presented, followed by description of qualitative relation between the BWR and eye movement metrics. Notably, the derived correlations between eye movement metrics and subjective workload ratings are based on a single pilot and therefore reflect only this individual's data, rather than generalizable trends across pilots.

The scatter plots in Figure 4.18 illustrate the quantitative correlation between eye movement metrics derived from the I-KF and I-VT algorithms and ISA workload ratings across all four test points (TP01–TP04), along with their respective r values. In the plots, blue dots represent I-KF data points and orange dots represent I-VT data points. Regression lines are included for both algorithms, with the same color as for the data points.

The top four scatter plots show the correlations between mean fixation duration and ISA ratings. For nearly all test points, except TP02 in both algorithms, mean fixation duration exhibits a positive correlation with ISA ratings. The strongest positive correlations occur at TP01 and TP03, with r values approaching 0.5. When averaged across test points, both algorithms show the same positive correlation of approximately r=0.2.

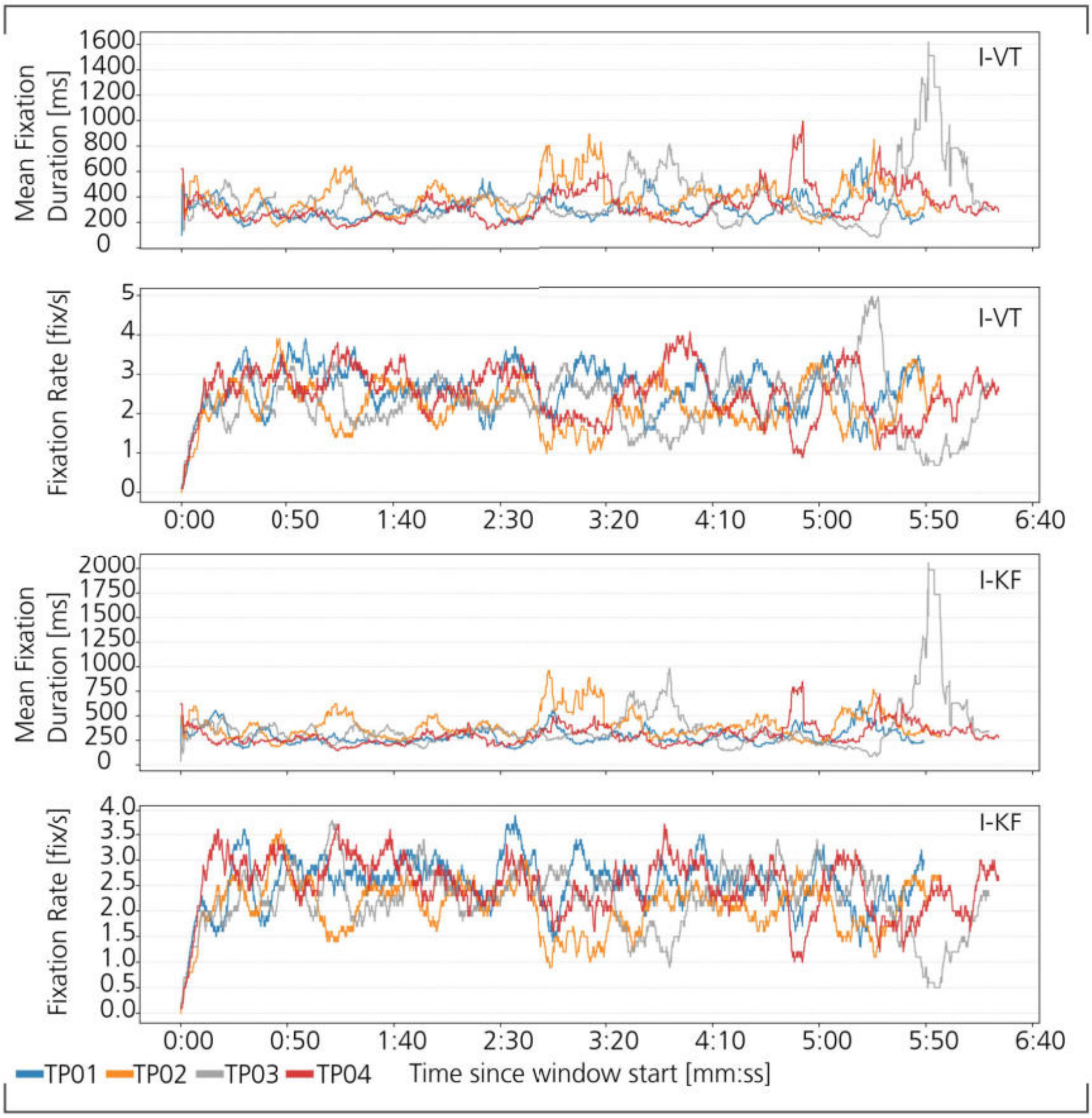


Figure 4.17.: **Mean Fixation Duration and Fixation Rate over a 10 Second Time Window for I-VT and I-KF Across all Test Points (TP01-TP04).**

Conversely, the fixation rate plotted in the lower four scatter plots exhibit generally negative correlations with ISA ratings for all test points except TP02 in both algorithms. The strongest negative correlation appears at TP03, with r values around -0.4. Overall, the I-VT algorithm shows slightly stronger negative correlations (average r ≈ -0.15) compared to I-KF (average r ≈ -0.1).

In total, correlations between mean fixation duration and ISA ratings are generally stronger than those observed for fixation rate.

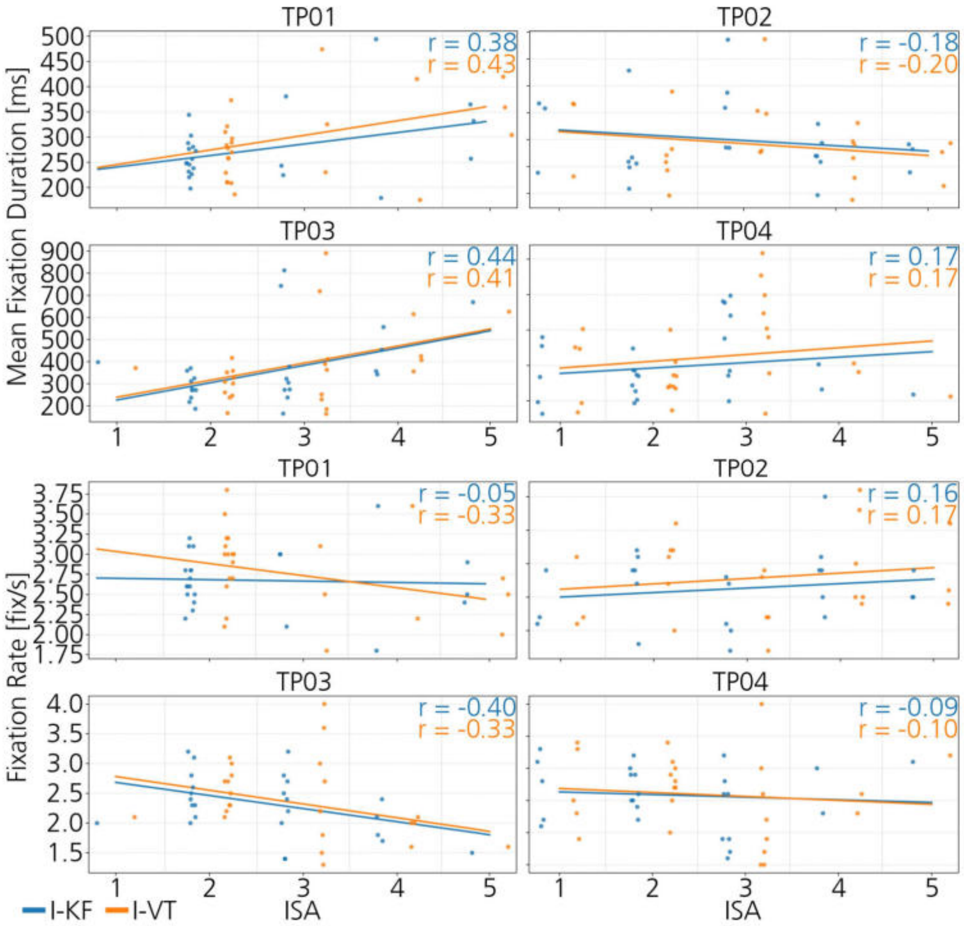


Figure 4.18.: **Correlation of Eye Movement Metrics with ISA Workload Ratings.**

BWR Ratings versus Eye Movement Metrics

The BWR ratings demonstrate a broad consistency with the eye movement metrics during the approach to the wind turbine and the stabilization phase to achieve hover conditions. Specifically, the higher BWR ratings given for the hover phase align with an observed increase in mean fixation duration and a corresponding decrease in fixation rate relative to the cruise phase. This pattern reflects the same relationship between workload and eye movement metrics as seen in the comparison with the ISA ratings.

However, similar to the comparison with ISA ratings, this alignment does not hold consistently when examined in more detail. For instance, in TP03, the peak in fixation duration, which is the highest across all test points, occurs during the final approach phase. In contrast, the highest BWR rating was neither assigned to TP03 nor to its final approach phase.

5. Discussion on Online Eye Movement Event Detection in a Helicopter Simulator

In the following, the verification of the developed online eye movement event detection algorithms applied in the helicopter simulator study is discussed. The aim is to assess their suitability for real-time use and their potential for supporting pilot monitoring and workload assessment. The verification is structured around three core aspects. The usability of the detectors, their capability to operate in real time and their event classification accuracy. Building on these findings, the feasibility of applying such algorithms to detect helicopter pilots eye movements is evaluated. In this context, the research questions stated in Section 1.1, concerning how online eye-tracking algorithms can be developed to effectively classify helicopter pilots' gaze points and to what extent their feasibility can be demonstrated in a preliminary full-flight simulator study, are addressed. Finally, the limitations of the eye movement detection setup are examined, outlining technical and methodological constraints that affect both the generalizability and quality of the results. These insights will help identifying areas of improvement and guiding future research toward practical implementation in helicopter flight simulators.

5.1. Verification of Algorithms for Helicopter Simulator Application

The verification for a helicopter simulation application is structured around three core aspects. The usability of the detectors, their capability to operate in real time and their event classification accuracy. Together, these factors verify the reliability and practical applicability of the eye movement event detection algorithms.

5.1.1. Usability of Eye Movement Event Detectors

Since the head-mounted eye tracker of the eye-tracking setup can connect via Wi-Fi, the system can be integrated into the AVES flight simulator as well as other flight simulator platforms or laptops equipped with the required .NET framework for C#. The script has been successfully tested on an off-the-shelf laptop, meaning no special hardware requirements are imposed on either the laptop or the experimental system. Additionally, the head-mounted eye tracker allows free head movement, without significantly impairing overall vision. To prevent slippage, the eye tracker can be secured with the provided headband, which the pilot noted to be effective. The pilot reported no notable discomfort or visual impairment from the device. However, this may be partly explained by their regular use of glasses for short-sight correction and may have therefore increased their tolerance to discomfort or minor obstructions, such as those introduced by the eye tracker's frame.

All in all, the eye-tracking setup is highly flexible and can be deployed across different flight simulators and potentially even in real helicopter flights, with only minimal impact on pilot comfort.

Furthermore, by organizing the algorithms into distinct modules for preprocessing, classification, post-processing, and the main pipeline, their maintainability and adaptability are significantly improved. This structure allows for future modifications or extensions to be integrated with minimal complexity. The online storage of preprocessed velocity and average gaze position makes the detection logic traceable, which is a critical requirement for validating their event detections and post-hoc analysis. Besides, the expert user's intuitive understanding of the metrics and the live terminal output from the sliding window indicates that the system requires minimal training. Clear notifications during initialization and termination further enhance the ease of operation.

The results also suggest that the eye movement metrics displayed in the online terminal can provide useful insights into ongoing behavioral patterns. Figure 4.17 illustrates variations in these metrics for different flight phases using the same 10-second time window as applied in the terminal display. Peaks observed during hover and final approach indicate that the metrics respond to flight mission demands, underscoring their potential for real-time analysis. Additionally, the correlations between ISA workload ratings and the online eye movement metrics further suggest that the observed variations are not purely by chance.

Finally, robustness to gaze data loss of the live display is demonstrated under conditions such as taking off the eye tracker. The system's ability to output "unknown" in the absence of valid events ensures that misleading classifications are avoided, which is essential for reliable online use.

5.1.2. Real-Time Performance Capability

The proportion of classified eye-tracking samples was close to optimal, with both algorithms showing only around 1.5% more N/A classifications than the theoretical minimum. This deviation arises because velocity calculation requires both the current and previous gaze sample. Consequently, following an N/A event, at least two valid gaze samples must be available before velocity and thus an eye movement event can be computed. The results show that the implemented approach of classifying data as soon as a single valid gaze sample is available significantly improves performance as without it, the proportion of N/A classifications would have been at least 16.8%, which is more than four times greater than the implemented approach.

Both algorithms also demonstrated stable sampling performance, with no evidence of data loss. The achieved sampling rate of around 49.91 Hz matched the recorded rate while being close to the nominal eye tracker rate of 50 Hz, indicating that samples were retrieved and processed with consistent approximate 20 ms spacing. Consequently, the buffer size of the algorithms proved sufficient to operate without imposing special demands on the CPU, even when running on a standard off-the-shelf laptop.

In terms of processing efficiency, the I-KF algorithm holds an advantage of about 20 ms in processing time, matching the expected latency difference. This is attributable to the I-VT's additional one-sample delay caused by the moving median smoothing filter. Preprocessing and event detection contributed only marginally to overall latency, with post-processing representing the largest component. This raises the question of whether post-processing should be reduced or omitted in applications where minimal latency is prioritized over maximum detection accuracy. Importantly, the algorithms required no special processing power and stable runtimes were achieved even on standard hardware. The inclusion of live terminal output and simultaneous online data storage only increased processing times minimally, with total execution remaining within 2–3 ms. This indicates that both algorithms are computationally efficient and capable of stable performance under online application.

5.1.3. Classification Accuracy of Eye Movement Events

The reliable accuracy of eye movement event detection was first assessed during offline algorithm development. Both the I-VT and I-KF algorithms outperformed the baseline I-TobiiVT in detecting events. Smooth pursuit performance was slightly lower in the moving point task, but this was mitigated by selecting slightly more conservative values for the final parameters of both algorithms.

The combined scatter plots from the AVES verification study showed a generally consistent relationship between ISA ratings and eye movement metrics. Specifically, mean fixation duration tended to increase, while fixation rate tended to decrease with higher subjective workload, except for TP02. For this test point, a negative correlation was observed in the r values, which can be attributed to the eye movement metric peaks occurring before the hover, whereas the ISA ratings peaked afterward. The BWR ratings suggest that these eye movement metric peaks are likely genuine, as the BWR identifies the stabilization-to-hover phase as the most critical, corresponding to the observed peaks.

For the other test points, r values showed positive correlations, confirming the expected workload-related trends. Strong positive correlations were observed primarily for TP01 and TP03, while the correlation for TP04 was relatively small. Overall ISA ratings decreased from test point to test point, suggesting a training effect. In contrast, the BWR classified TP04 as having the highest critical workload, likely due to the combination of DVE conditions and the more challenging ACPR SASy DIC flight control law. This indicates that for TP04, ISA ratings alone may not be reliable, as the eye movement metrics also showed the second-highest extreme value for this test point, corresponding more closely to the BWR rating.

For TP01 and TP03, general trends were consistent between ISA ratings and eye movement metrics, likely contributing to the higher positive correlations. However, pilot feedback and BWR ratings suggested that maintaining hover was less demanding than stabilizing into it, whereas the ISA scale and eye movement metrics reflect workload more generally across the entire stabilization and hovering phase, without capturing the distinct increase in workload during the stabilization phase alone.

One peculiarity between ISA ratings and eye movement metrics is the highly variable mean fixation durations and very low fixation rates observed during takeoff at each test point, compared to the ISA ratings, which generally identified takeoff as the second-highest workload phase. This ISA rating is plausible, given the takeoff setup. The helicopter started in hover in ground effect above the platform, requiring immediate stabilization, which significantly increased workload, as reported by the pilot. In contrast, the eye movement metrics during this period are affected by the first 10 seconds, delivering inaccurate results for the takeoff phase, as can be seen in Figure 4.17. During this interval, the sliding window covers less than 10 seconds, causing inaccuracies in both mean fixation duration and fixation rate. Specifically, mean fixation duration is computed over a shorter interval, while fixation rate is normalized to the predefined 10-second window regardless of its actual length. This leads to large initial fluctuations in mean fixation duration and a sharp increase in fixation rate during the first 10 seconds, making the metrics in this time frame unreliable and rendering the takeoff phase incomparable with the subjective workload ratings.

Another peculiarity is that the highest extreme value across all test points occurred in TP03 during the final approach phase, which neither the BWR nor ISA ratings identified as the highest workload for that test point or overall. This extreme value may, however, be plausible. The gaze data recording suggests that the unusually long fixation was due to the pilot using the platform directly in front of them as a visual reference. Notably, this visual reference was used only in TP03, likely because it was the first test point conducted under DVE conditions. In TP04, the pilot did not use this visual cue, likely due to increased familiarity with the environment, as indicated by the lower ISA and BWR ratings for the approach phase compared to TP03, although differences in the flight control law may also have contributed.

Comparing the two online algorithms, I-VT and I-KF, revealed generally consistent trends in mean fixation duration and fixation rate across flight phases. The I-KF method showed slightly higher peak values for both metrics, particularly during the final approach, where maximum fixation durations reached 2.0 seconds for I-KF and 1.6 seconds for I-VT. Although fixation durations of several seconds are uncommon, they can occur under specific task conditions, as in this case, where the pilot used the platform directly in front of their view as a visual reference, making both values plausible.

The results show that, while phase-specific discrepancies were observed between the workload ratings and the eye movement metrics, overall trends were identifiable, and the metrics fell within the range of expected mean fixation durations and rates. Furthermore, although the optimization was performed in a different flight simulator, the algorithms also demonstrated high accuracy in the fixation-saccade invocation task study. Finally, the comparison between subjective workload and eye movement metrics was conducted solely to verify reliable event detection and not to determine event detection accuracy. Given that both algorithms demonstrated reliable performance and that optimization results did not identify a superior event detection method, it cannot be concluded that either algorithm outperforms the other in terms of event detection performance.

5.2. Feasibility and Limitation of the Eye Movement Event Detection

In the following, the feasibility and limitations of the developed online eye movement event detection algorithms, as well as their setup and verification, are discussed. Building on the verification results presented earlier, the discussion is divided into two parts.

The first part revisits the two research questions posed at the beginning of the thesis, evaluating the feasibility of the I-VT and I-KF algorithms for online application and their development approach. The second part addresses the methodological limitations of the verification study and the technical constraints of the event detection setup.

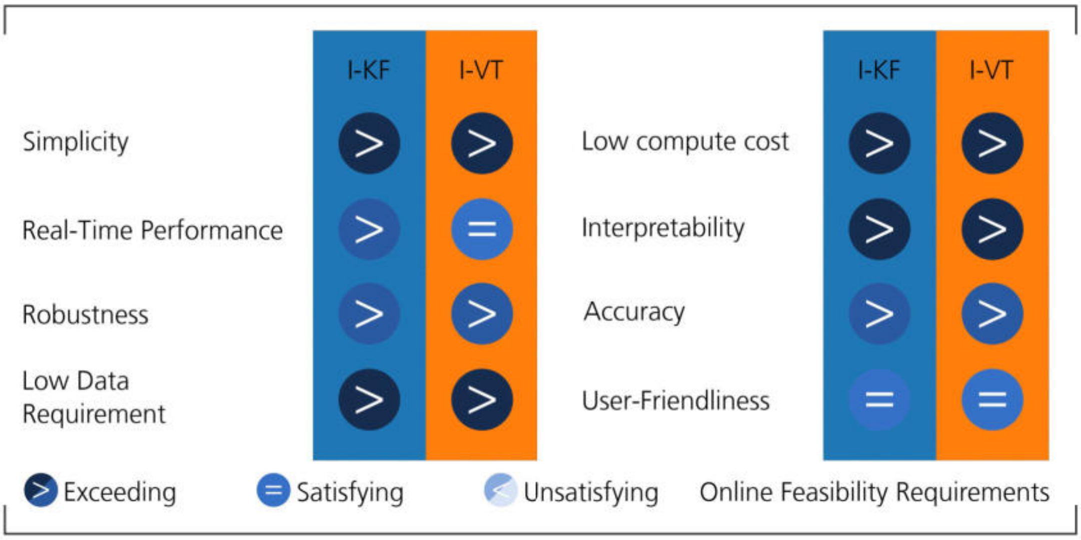
5.2.1. Feasibility of the Eye Movement Event Detection Algorithms

Finally after discussing the verification of the algorithms the two research question stated at the beginning of the thesis can be reflected on.

The first research question addressed how an online eye-tracking algorithm can be developed and optimized to effectively classify real-time gaze points of helicopter pilots. The findings demonstrate that such an algorithm can indeed be developed using a relatively small, synthetically derived dataset from a fixation–saccade invocation task. Since the elicited eye movement patterns resemble those typically observed in helicopter flight simulators, the algorithms could be optimized for this specific use case. From this process, two implementations, namely the I-VT and the I-KF, were adapted for online applicability and verified in the AVES flight simulator. Importantly, both algorithms maintained reliable detection accuracy in both AVES and 2PASD, suggesting that they can be applied across simulator environments and provide a valid approach for developing online algorithms suitable for helicopter pilot eye movement event detection.

The second research question concerned the feasibility of these algorithms in real-time application, which was assessed across multiple criteria. These criteria allow for a critical appraisal of the algorithms' suitability for online eye movement event detection of helicopter pilots and their assessment is summarized in a qualitative evaluation depicted in Figure 5.1. The chart illustrates both algorithms and their performance for each criteria on a scale ranging from unsatisfying to exceeding online feasibility requirements. Both online algorithms exceed performance expectations in terms of simplicity, interpretability, low computational cost and minimal data requirements for parameter optimization. Although accuracy and robustness exceeded satisfactory levels, they fell short of the highest rating, as deep learning models, as discussed in the literature review, outperform both algorithms in these criteria. Real-time performance, on the other hand, reached only a satisfactory level, with both algorithms achieving latencies of around 100 ms, and the I-KF showing a slight advantage. User-friendliness also remains an aspect with considerable room for improvement and was therefore rated as only satisfactory.

Overall, both algorithms can be considered feasible for online application. As a final recommendation, the I-VT filter is suggested, as its only drawback compared to the I-KF is slightly higher latency, while its event detection method is based on the I-TobiiVT and allows researchers in the field to cross-validate and reproduce results by adjusting I-TobiiVT's velocity threshold to the personalized threshold of the I-VT.



	I-KF	I-VT		I-KF	I-VT
Simplicity	>	>	Low compute cost	>	>
Real-Time Performance	>	=	Interpretability	>	>
Robustness	>	>	Accuracy	>	>
Low Data Requirement	>	>	User-Friendliness	=	=

> Exceeding = Satisfying < Unsatisfying Online Feasibility Requirements

Figure 5.1.: **Evaluation of I-VT and I-KF Online Algorithm's Overall Performance.**

5.2.2. Limitations of the Eye Movement Event Detection Setup and Verification

Despite these strengths, several limitations remain regarding the event detection setup and its verification.

First, while the head-mounted eye tracker allows free head movement, it obstructs peripheral vision due to the glasses' frames. Wearing the glasses may also cause discomfort and slippage is possible during longer sessions. Nevertheless, these issues appear to be minor, as the pilot reported no significant impairment of vision and comfort was rated high when the Tobii's eye tracker headband was used to prevent slippage.

Second, since the primary focus of this thesis was on achieving reliable eye movement event detection, the development of a user-friendly graphical interface received less attention and thus represents the area with the greatest potential for improvement. Currently, the configuration of live output and event detection parameters is hardcoded in the script, making it difficult to adjust settings such as the investigation period, sliding window length, or detection thresholds. Furthermore, the system does not provide a synchronized live video stream, which would enable direct linking of gaze classifications to the visual scene.

Third, although the verification study demonstrated reliable event detection across simulator setups, it cannot be ruled out that slightly different optimal parameter values occur if optimization were performed directly in the AVES simulator. This is particularly relevant because the optimal configuration depends on factors such as the distance from pilot seat to instrument panels and primary vision system.

Finally, the verification was conducted with a single participant and the study was primarily designed to test event detection rather than workload or situational awareness measures. While correlations between eye movement measures and workload provide preliminary evidence of validity, larger and more diverse studies are required to reduce variability and strengthen generalizability. Furthermore, the observed correlations between eye movement metrics and workload appear minor, however, they may underestimate the true relationship. This is partly because workload was measured using the ISA ratings, which are inherently subjective and may not fully capture the pilot's actual cognitive load. Addtionally, the temporal resolution of the measurements introduces further limitations. ISA ratings were intended to be recorded at 15-second intervals, but in practice they were not always provided exactly on schedule, while eye movement metrics were computed continuously using 10-second intervals. This mismatch in both interval length and timing may have reduced their correlation, giving the appearance of a weak association between workload and eye movement metrics.

6. Conclusion and Outlook

This thesis investigated how online eye-tracking algorithms can be developed to effectively classify the gaze points of helicopter pilots and to what degree their feasibility can be demonstrated in a preliminary study in a full-flight simulator. Building on a structured development and verification process, the work provided methodological contributions to the design of online eye movement event detection algorithms and two deployable implementations.

The thesis began with a literature review to establish the foundations for algorithm development. First, an overview of common eye trackers was conducted, with particular focus on head-mounted trackers and the PCCR method, as the eye-tracking setup had been predetermined to use this approach. Following this, the eye movement typologies relevant to event detection were examined and their potential application to future use cases such as online workload and situational awareness assessment or adaptive pilot support systems was explored to derive the algorithm requirements. Next, the main algorithmic approaches that enables these future use cases, which are threshold based, probability based and deep learning models, were studied, followed by a review of suitable evaluation metrics at the event-statistic, sample and event level.

Based on this groundwork, several algorithms were optimized for helicopter pilot use. Optimization was conducted using the Optuna framework with gaze data and ground truth derived from a study based on fixation-saccade invocation tasks, which replicated typical eye movement patterns observed in helicopter flight. Four algorithms (I-VT, I-DT, I-KF, and I-HMM) were evaluated, while deep learning models were excluded due to the limited available dataset size and their limited interpretability. After parameter optimization, additional analysis of smooth pursuit handling and parameter validation on a test dataset, the I-VT and I-KF algorithms were identified to outperform the baseline I-TobiiVT and were selected for online adaptation.

To verify their applicability, a study was conducted in the AVES flight simulator. The designed mission simulated offshore wind farm operations under varying visual environments and flight controls to induce different workload demands. Results showed that both I-VT and I-KF maintained reliable detection accuracy in AVES and 2PASD simulators, indicating that the optimized parameters generalize across different simulation environments. Both algorithms proved feasible for online application, with the I-VT offering the additional benefit of easier reproducibility and comparability to the existing I-TobiiVT.

Beyond these two implementations that have been proven to be feasible for helicopter pilot eye movement event detection, the thesis provides several broader contributions. The fixation–saccade invocation task study generated a synthetic dataset that can be reused to optimize or validate algorithm parameters for helicopter specific applications. Furthermore, a novel optimization pipeline was developed, combining IoU-based event matching, a customized penalization strategy for mismatches and intelligent parameter search with the Optuna framework. Finally, the verification demonstrated robustness and reliable event detection across AVES and 2PASD, strengthening the case for generalizability.

While this thesis provides important contributions and demonstrates the feasibility of online eye movement event detection for helicopter pilots, certain limitations remain. The verification was conducted with a single participant, which restricts the comparability and generalizability of workload related results. Furthermore, aspects such as hardcoded algorithm configuration, the absence of synchronized video feedback, and the potential influence of the head-mounted tracker on comfort and peripheral vision point to areas where usability could be enhanced. Finally, optimal parameter settings may vary depending on pilot–instrument distances, primary vision system configurations, or cockpit layouts, which were beyond the scope of this work.

Building on these identified constraints of the thesis, several recommendations for future development and research on eye movement event detection and its setup for online application in helicopter environments are proposed.

First, if head position data from head-mounted eye trackers can be reliably retrieved in real time, it should be incorporated into the preprocessing stage. Accounting for head movement would substantially reduce noise and enable reliable detection of smooth pursuit eye movements. With head movement corrected, postprocessing may no longer be required, as the majority of fixation fragmentation stems from head movement noise and smooth pursuits. Removing this step from the algorithm would in turn significantly reduce latency. To further reduce slippage-induced noise, the use of a headband is recommended to stabilize the eye tracker.

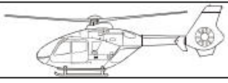
Second, future implementations should also consider alternative detection methods. In particular, deep learning models may become viable once larger ground-truth datasets are available, offering potentially higher accuracy and robustness.

Third, while the current PowerShell based online eye movement event metrics and sliding window display was highly valued by the user, usability enhancements should be implemented for broader adoption. Configuration should be made more accessible by shifting parameter adjustments to a configuration file or command-line interface, thereby eliminating the need for source code modifications. User-friendliness could be further enhanced through the development of a graphical or web interface. Furthermore, synchronizing the live video stream with event classifications would further enhance interpretability by linking gaze events directly to the visual scene. To address the live video stream synchronization, the tool could leverage the clock synchronization capabilities of the Tobii Pro SDK. The modular design of the current implementation provides a solid basis for extending such functionalities.

Fourth, for future workload and situational awareness assessments based on eye movement, the use of the modal fixation duration instead of the mean should be examined, given the skewed distribution of fixation durations reported in the literature. Furthermore, incorporating additional physiological and behavioral measures, such as pupil diameter, electrodermal activity (EDA), and electrocardiogram (ECG), alongside eye movement metrics could enhance online workload assessment by providing complementary indicators

Finally, future research should also extend verification to larger studies in different operational settings and more pilots with also diverse pilot backgrounds. The algorithms could potentially be verified in a real helicopter flight as well. The ISA rating proved to be a useful benchmark for comparison with eye movement metrics and might be able to be adapted for shorter 10-second intervals if implemented with a button press, allowing pilots to input ratings themselves without the need for verbal communication. Finally, beyond event detection verification, a validation on its application as real-time workload or situational awareness assessment or adaptive pilot support systems should also be considered.

A. Appendix A

		MA Verification Study - 2025	
Date	Simulator AVES	Offshore Hoist Operation	Participant ID

User Survey Questionnaire

This questionnaire is designed to gather feedback on the usability, clarity and performance of the online eye movement event detections scripts. Please answer the questions below.

Output & Interpretation

Is the console output layout easy to read in real time?

Do you understand what each metric means without additional explanation?

Does the use of the sliding window help you identify trends in the data?

Would you prefer a different format for displaying event history?

Would color coding or alternative symbols make the output easier to interpret?

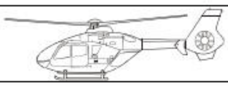
Is the latency display and UTC Time helpful for your use case?

Performance & Responsiveness

Did the script feel responsive (enough) during your session?

Did you experience any delays that interfered with real-time monitoring?

Can you think of scenarios where the algorithm might misclassify events?

DLR		MA Verification Study - 2025	
Date	Simulator AVES	Offshore Hoist Operation	Participant ID

Configuration & Flexibility

Are the adjustable settings (e.g., bar width, history lengths, post-processing toggle) easy to locate and modify?

Would you prefer a configuration file or command-line options instead of editing the source code?

Are there features you would like to toggle on/off while the script is running?

Are there any additional metrics you think should be included for real-time analysis?

Practical Use Case

Could you imagine using this tool in a live flight test scenario?

For future applications, would you need the script's output stream to be synchronized with live scene video playback?

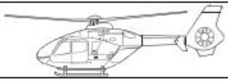
Overall Experience

Was there any confusing part of running the script?

What did you like most about the script's interface?

What would you change to improve usability?

B. Appendix B



MA Verification Study - 2025

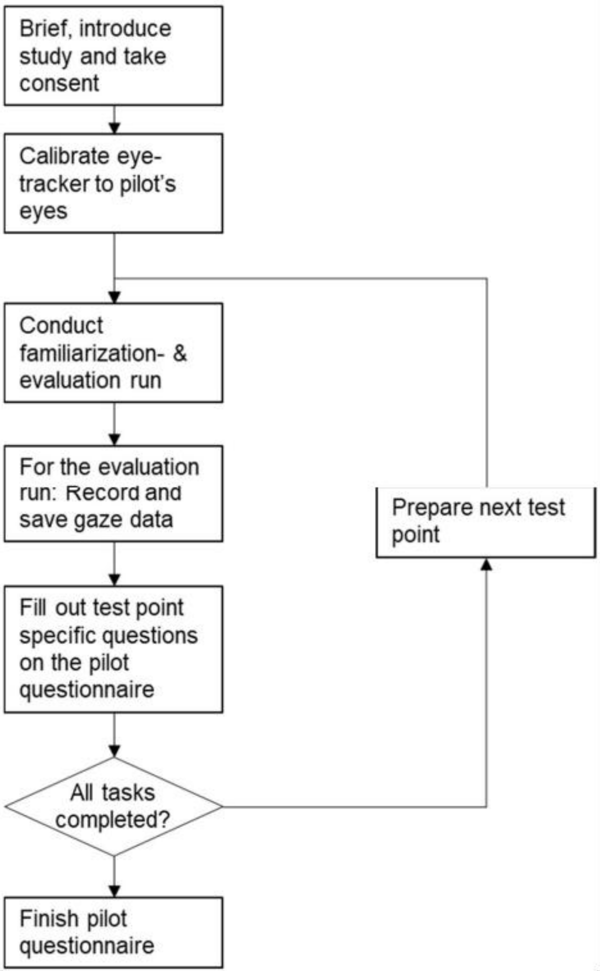
Date	Simulator AVES	Offshore Hoist Operation	Participant ID

Study Goal

The goal is to verify the online applicability of an online eye movement event detector in a flight simulator. By introducing flight tasks with varying flight conditions but a consistent flight scenario, a range of cognitive workloads are induced to verify the algorithm's performance.

Procedure

Perform familiarization- & evaluation run:
Familiarization ⇒ 2x or according to pilot's decision
Evaluation run ⇒ pilot's evaluation, questionnaire and ratings



Brief, introduce study and take consent

Calibrate eye-tracker to pilot's eyes

Conduct familiarization- & evaluation run

For the evaluation run: Record and save gaze data

Fill out test point specific questions on the pilot questionnaire

All tasks completed?

Prepare next test point

Finish pilot questionnaire

Flight Scenario and Test Matrix

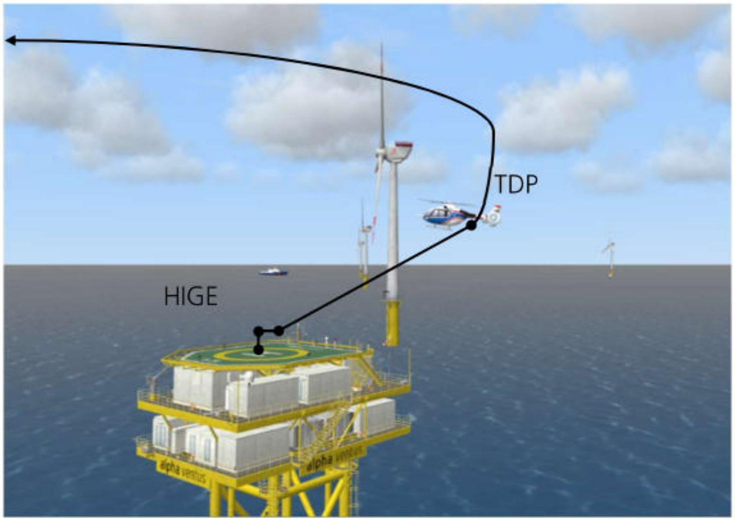
Execution according to test matrix and flight scenario description below.
Recording of pilot's gaze movements with Tobii Pro Glasses 3.

Test Matrix: Offshore Hoist Operation	**2SAS 3AXES**	**5AC$_{PR}$ SAS$_{Y}$ DIC**
GVE 40.000	TP01	TP02
DVE 2.500	TP03	TP04

		MA Verification Study - 2025	
Date	Simulator AVES	Offshore Hoist Operation	Participant ID

Flight Scenario Description

Performance – Helipad T/O	Flight Limits
• Hover with HIGE at x ft height and x m rearward movement	4+15; 3+6
• Fly backward to TDP of x ft AHE	120±30
• Transition to forward flight and accelerate to X kts	65±20

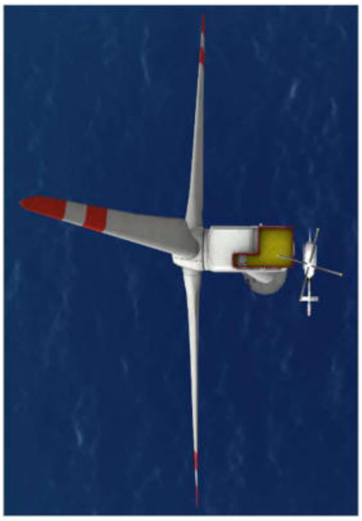


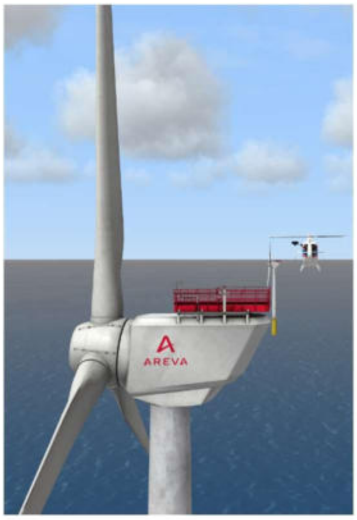
Performance – Approach to Wind Turbine	Flight Limits
• Maintain an airspeed of X knots throughout the course	100±10
• Accomplish maneuver at an altitude of X ft	300±30

Target Idle Wind Turbine

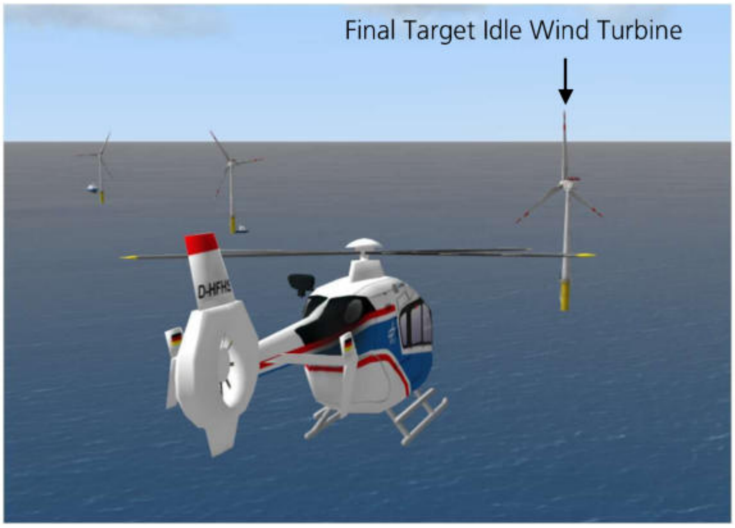
DLR		MA Verification Study - 2025	
Date	Simulator AVES	Offshore Hoist Operation	Participant ID

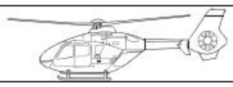
Performance – Establish Hover Position over Hoist Target	Flight Limits
• Maintain a stabilized hover for at least X seconds	30
• Maintain altitude of X ft	330±15
• Maintain heading of X degree	275±5





Performance – Depart and Arrive at Final Wind Turbine	Flight Limits
• Accelerate after departure to X kts and decelerate when arriving at the other idle wind turbine	60±10
When arriving at final wind turbine:	
• Maintain altitude of X ft	330±15
• Maintain heading of X degree	275±5



DLR		MA Verification Study - 2025	
Date	Simulator AVES	Offshore Hoist Operation	Participant ID

Questionnaire

PLEASE ANSWER THE QUESTIONS BELOW.

FOR A SCHEMATIC VIEW OF THE BEDFORD WORKLOAD RATING SCALE (BR) AND THE INSTANTANEOUS SELF-ASSESSMENT (ISA), REFER TO THE DESIGNATED PAGES.

WHAT IS WORKLOAD?

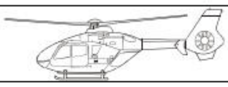
"WORKLOAD IS THE INTEGRATED MENTAL AND PHYSICAL EFFORT REQUIRED TO SATISFY THE PERCEIVED DEMANDS OF A SPECIFIED FLIGHT TASK" (ROSCOE, 1987)

For each task, please:

1. *Assign a workload rating using the Bedford and ISA scales (refer to the provided scale sections).*

2. *Answer the task-specific difficulty questions based on your experience during the task.*

After all tasks are completed, please fill out the Post-Study Questionnaire.

DLR		MA Verification Study - 2025	
Date	Simulator AVES	Offshore Hoist Operation	Participant ID

TP 01

No. of Hot Runs:

Annotations regarding hot runs:

ISA Ratings:

0:15:	0:30:	0:45:	1:00:	1:15:	1:30:	1:45:	2:00:
2:15:	2:30:	2:45:	3:00:	3:15:	3:30:	3:45:	4:00:
4:15:	4:30:	4:45:	5:00:	5:15:	5:30:	5:45:	6:00:

Bedford Workload Ratings:

Bedford Workload Rating for “Approach to Wind Turbine” flight phase:

1	2	3	4	5	6	7	8	9	10

Bedford Workload Rating for most critical flight phase:

1	2	3	4	5	6	7	8	9	10

which flight phase do you considered the most critical?

How would you rate the visibility during the approach to the wind turbine?

1	2	3	4	5

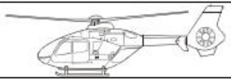
Were the flight control settings manageable?

1	2	3	4	5

Did the eye tracker cause any visual impairment?

1	2	3	4	5

Comments:



MA Verification Study - 2025

Date	Simulator AVES	Offshore Hoist Operation	Participant ID

TP 02

No. of Hot Runs:

Annotations regarding hot runs:

ISA Ratings:

0:15:	0:30:	0:45:	1:00:	1:15:	1:30:	1:45:	2:00:
2:15:	2:30:	2:45:	3:00:	3:15:	3:30:	3:45:	4:00:
4:15:	4:30:	4:45:	5:00:	5:15:	5:30:	5:45:	6:00:

Bedford Workload Ratings:

Bedford Workload Rating for “Approach to Wind Turbine” flight phase:

1	2	3	4	5	6	7	8	9	10

Bedford Workload Rating for most critical flight phase:

1	2	3	4	5	6	7	8	9	10

which flight phase do you considered the most critical?

How would you rate the visibility during the approach to the wind turbine?

1	2	3	4	5

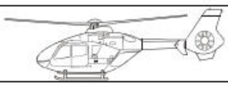
Were the flight control settings manageable?

1	2	3	4	5

Did the eye tracker cause any visual impairment?

1	2	3	4	5

Comments:



MA Verification Study - 2025

Date	Simulator AVES	Offshore Hoist Operation	Participant ID

No. of Hot Runs:

Annotations regarding hot runs:

ISA Ratings:

0:15:	0:30:	0:45:	1:00:	1:15:	1:30:	1:45:	2:00:
2:15:	2:30:	2:45:	3:00:	3:15:	3:30:	3:45:	4:00:
4:15:	4:30:	4:45:	5:00:	5:15:	5:30:	5:45:	6:00:

Bedford Workload Ratings:

Bedford Workload Rating for “Approach to Wind Turbine” flight phase:

1	2	3	4	5	6	7	8	9	10

Bedford Workload Rating for most critical flight phase:

1	2	3	4	5	6	7	8	9	10

which flight phase do you considered the most critical?

TP 03

How would you rate the visibility during the approach to the wind turbine?

1	2	3	4	5

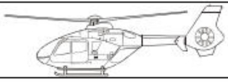
Were the flight control settings manageable?

1	2	3	4	5

Did the eye tracker cause any visual impairment?

1	2	3	4	5

Comments:



		MA Verification Study - 2025	
Date	Simulator AVES	Offshore Hoist Operation	Participant ID

TP 04

No. of Hot Runs:

Annotations regarding hot runs:

ISA Ratings:

0:15:	0:30:	0:45:	1:00:	1:15:	1:30:	1:45:	2:00:
2:15:	2:30:	2:45:	3:00:	3:15:	3:30:	3:45:	4:00:
4:15:	4:30:	4:45:	5:00:	5:15:	5:30:	5:45:	6:00:

Bedford Workload Ratings:

Bedford Workload Rating for “Approach to Wind Turbine” flight phase:

1	2	3	4	5	6	7	8	9	10

Bedford Workload Rating for most critical flight phase:

1	2	3	4	5	6	7	8	9	10

which flight phase do you considered the most critical?

How would you rate the visibility during the approach to the wind turbine?

1	2	3	4	5

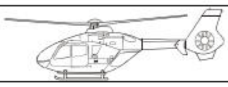
Were the flight control settings manageable?

1	2	3	4	5

Did the eye tracker cause any visual impairment?

1	2	3	4	5

Comments:

		MA Verification Study - 2025	
Date	Simulator AVES	Offshore Hoist Operation	Participant ID

Post-Study Questionnaire	
No	*Please provide your background information and qualitative impressions related to the simulator session.*
01	What is your age?
	___ years
02	What is your eye color?
03	Do you currently use any vision aids (e.g., glasses, contact lenses)?
04	How many years have you held a valid pilot license?
	___ years
05	Approximately how many flight hours do you have?
	Total flight hours: _______ Flight hours on EC135: _______
06	Are you familiar with research flight tests?
	□ Yes □ Somewhat □ No
07	Do you have any additional comments or feedback regarding the study or tasks?

DLR		MA Verification Study - 2025	
Date	Simulator AVES	Offshore Hoist Operation	Participant ID

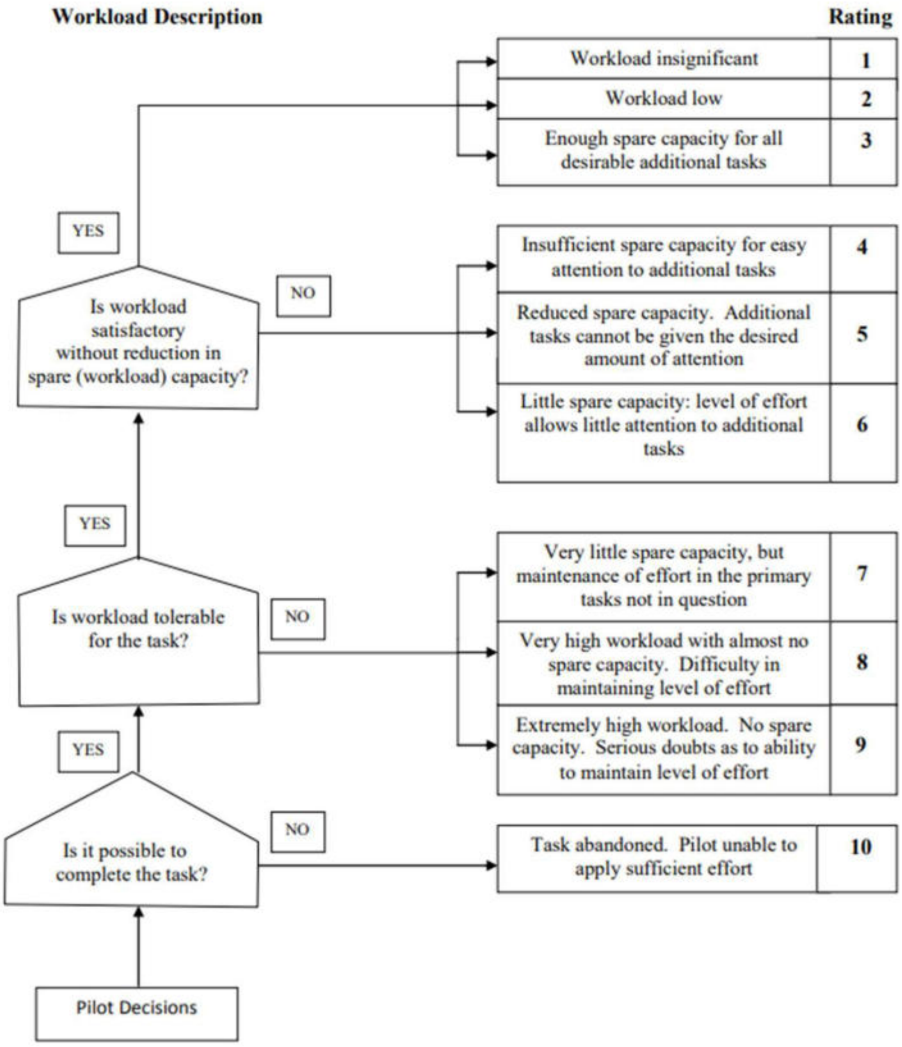

Workload Description
Rating
Workload insignificant
1
Workload low
2
Enough spare capacity for all desirable additional tasks
3
YES
Insufficient spare capacity for easy attention to additional tasks
4
NO
Is workload satisfactory without reduction in spare (workload) capacity?
Reduced spare capacity. Additional tasks cannot be given the desired amount of attention
5
Little spare capacity: level of effort allows little attention to additional tasks
6
YES
Very little spare capacity, but maintenance of effort in the primary tasks not in question
7
Is workload tolerable for the task?
NO
Very high workload with almost no spare capacity. Difficulty in maintaining level of effort
8
YES
Extremely high workload. No spare capacity. Serious doubts as to ability to maintain level of effort
9
NO
Is it possible to complete the task?
Task abandoned. Pilot unable to apply sufficient effort
10
Pilot Decisions

Figure 1. Bedford Workload Rating Scale
(Source: Roscoe, A. H., & Ellis, G. A. (1990). *A subjective rating scale for assessing pilot workload in flight: A decade of practical use* (Technical Report No. 90019). Bedford, UK: Royal Aerospace Establishment, Ministry of Defence.)

Level	Workload Heading	Spare Capacity	Description
5	Excessive	None	Behind on tasks; losing track of the full picture.
4	High	Very Little	Non-essential tasks suffering. Could not work at this level very long.
3	Comfortable Busy Pace	Some	All tasks well in hand. Busy but stimulating pace. Could keep going continuously at this level.
2	Relaxed	Ample	More than enough time for all tasks. Active on task less than 50% of the time.
1	Underutilised	Very Much	Nothing to do. Rather boring.

Figure 2. ISA workload scale
(Source: Kirwan, B., Evans, A., Donohoe, L., Kilner, A., Lamoureux, T., Atkinson, T., & MacKendrick, H. (1997). Human factors in the ATM system design life cycle. In *Proceedings of the FAA/Eurocontrol ATM R&D Seminar* (pp. 16–20). Paris, France.)

C. Appendix C

Questionnaire P-ID:

Please fill in the ratings in section 1 after each task completion. If you encounter a task for a second time and wish to change your previous rating, please cross out the new score and encircle it. The tasks listed below may not be in the same order as you experience them in the simulator.

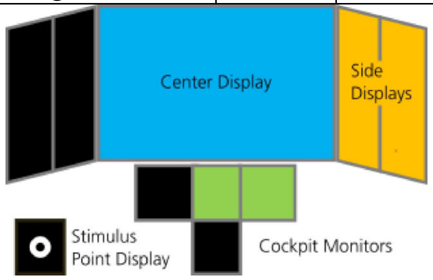
Section 1: Task-Specific Feedback

For each task you complete, please rate the perceived difficulty to trace the stimulus point:

	Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
The Jumping Point on the Center Display was difficult to trace.					
The Jumping Point on the Cockpit Monitors was difficult to trace.					
The Jumping Point on the Side Displays was difficult to trace.					
The Jumping Point on the Cockpit Monitors and Center Display was difficult to trace.					
The **Clustered** Jumping Point on the Cockpit Monitors and Center Display was difficult to trace.					
The **Moving** Point on the Side Displays was difficult to trace.					

For each task, how mentally demanding did you find it?

	Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
The Jumping Point Task on the Center Display was mentally demanding.					
The Jumping Point Task on the Cockpit Monitors was mentally demanding.					
The Jumping Point Task on the Side Displays was mentally demanding.					
The Jumping Point Task on the Cockpit Monitors and Center Display was mentally demanding.					
The **Clustered** Jumping Point Task on the Cockpit Monitors and Center Display was mentally demanding.					
The **Moving** Point Task on the Side Displays was mentally demanding.					



Questionnaire P-ID:

Section 2: Participant Background and Qualitative Feedback

What is your age? ____

What is your gender?____

What is your eye color?____

Do you use any vision aids (contact lenses)? __________________________________

Do you hold a helicopter pilot license? (Yes/No)

Do you have experience flying helicopters in flight simulators? (Yes/No)

Were there any aspects of the tasks or simulation that felt confusing?

____

____

____

____

Do you have any other comments or feedback about the study or the tasks?

____

____

____

____