Simulation and Classification of Spatial Disorientation in a Flight Use-Case Using Vestibular Stimulation

Jamilah Foucher, Anne-Claire Collet, Kevin Le Goff, Thomas Rakotomamonjy, Valerie Juppet, Thomas Descatoire, Jeremie Landrieu, Marielle Plat-Robain, Francois Denquin, Arthur J Grunwald, et al.

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 3 September 2022, accepted 20 September 2022, date of publication 28 September 2022, date of current version 7 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3210526

RESEARCH ARTICLE

Simulation and Classification of Spatial Disorientation in a Flight Use-Case Using Vestibular Stimulation

JAMILAH FOUCHER[1], (Member, IEEE), ANNE-CLAIRE COLLET[2], KEVIN LE GOFF[3], THOMAS RAKOTOMAMONJY[4], VALÉRIE JUPPET[5], THOMAS DESCATOIRE[5], JERÉMIE LANDRIEU[1], MARIELLE PLAT-ROBAIN[3], FRANÇOIS DENQUIN[1,4], ARTHUR J. GRUNWALD[6], JEAN-CHRISTOPHE SARRAZIN[4], AND BENOÎT G. BARDY[1]

[1]EuroMov Digital Health in Motion, University of Montpellier, 34090 Montpellier, France
[2]Human Design Group, 31000 Toulouse, France
[3]Airbus, 31000 Toulouse, France
[4]DTIS, ONERA, 13300 Salon-de-Provence, France
[5]Airbus Helicopters, 13700 Marignane, France
[6]Technion—Israel Institute of Technology, Halifa 32000, Israel

Corresponding author: Jamilah Foucher (j622amilah@gmail.com)

This work was supported in part by the French Department of Civil Aviation (DGAC), and in part by the European Union (Fonds Europeen de Developpement Regional (FEDER) iMOSE).

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the EuroMov Institutional Review Board (IRB) at the University of Montpellier under IRB-EM Rotational No. 1703B and IRB-EM Translational No. 1704B, and performed in line with the 1964 Declaration of Helsinki.

ABSTRACT A commonly used definition of spatial disorientation (SD) in aviation is ''an erroneous sense of one's position and motion relative to the plane of the earth's surface''. There exists a wide range of SD use-cases dictated by situational factors, therefore SD has been predominantly studied using reduced motion detection experimental contexts in isolation. The study of SD by use-case makes it difficult to understand general SD occurrence and thus provide viable solutions. To investigate SD in a generalized manner, a two-part Human Activity Recognition (HAR) study was performed. In Part I, a generalized SD perception dataset was created using whole-body experimental motion detection methods in a naturalistic flight context; joystick response was measured during rotational or translational vestibular stimulation. Results showed that SD occurred less for faster speeds than slower speeds, and specific orientations and axes were more difficult to detect motion. Part II evaluated supervised and unsupervised model parameters, including: model architecture, data use-case, feature-type, feature quantity, ground-truth labeling, unsupervised labeling. Long-Short Term Memory (LSTM), Random Forest (RF), and Transformer Encoder models most accurately predicted SD with mean accuracy of 0.84, 0.82, and 0.77 respectively. Using permutation importance (PIM), a dependency score for time, frequency, and time & frequency feature-types quantified the amount that each model architecture depended on a feature-type. The lenient ground-truth label best characterized features, and K-medoids clustering using position and velocity features most accurately replicated ground-truth labels.

INDEX TERMS Aircraft navigation, human computer interaction, machine learning, deep learning, unsupervised learning, motion detection, dead reckoning, activity recognition, supervised learning, joystick response, spatial disorientation.

I. INTRODUCTION

Spatial Disorientation (SD), in aviation, is the failure to perceive orientation, position, or movement. It is caused by multiple factors including environmental references and

The associate editor coordinating the review of this manuscript and approving it for publication was Aasia Khanum.

conditions, experience, and stress. There are diverse types of SD symptoms, ranging from confusion to physical sickness, and currently there is no proven method or solution to prevent it [1], [2], [3], [4], [5], [6]. International studies on the frequency and severity of SD accidents show that 6-32% of major accidents are due to SD, similarly 15-26% of fatal accidents are a result of SD [6]. Recovery from SD is strongly connected to the pilot's awareness of the situation, and his/her ability to perform corrective control to maintain aerodynamic stability despite disorientation; 80% and 20% of SD incidents are caused by unrecognized and recognized situations respectively [1], [7]. However, most importantly, there lacks general understanding of SD onset with respect to orientation, position, and speed using environmental references because it is difficult to label SD and non-SD time periods for time-series human activity measurements during real-world applications. Human Activity Recognition (HAR) is the research field in which time-series and/or image data are used with Machine Learning (ML) & Deep Learning (DL) algorithms to predict human activity in unconstrained real-world situations. HAR encompasses three main fields of study: gait monitoring, human pose estimation, and human activity recognition. HAR is the study of human behavior, including physical and long-term habits, using a wide variety of sensors such as accelerometers/gyroscopes, cameras, RFIDs, and environmental measures [8], [9], [10], [11], [12], [13], [14]. We hypothesize that SD occurrence can be predicted using HAR measurement and analysis methods. Therefore, in this study we propose a motion detection experiment where SD orientation or position, and speed situations are induced, similar to typical walking and running scenarios in HAR, such that human activity of one's perceived position or orientation are measured.

SD has been investigated in various scientific fields, including aviation, psychophysical human motion detection, control theory, neuroscience, and neuroergonomics. Depending on the scientific field, the approach to quantify human behavior during SD occurrence has been recasted in terms of each field's specialty. From an aviational approach, SD has been investigated by; categorizing physiological and environmental situations of SD occurrence, referred to as SD use-cases, using questionnaire-based methods with the goal of creating a behavioral instruction map to prevent SD occurrence; and pilot education and training of SD use-cases such that flight maneuvers do not cause physiological excitation to exceed human vestibular thresholds. For instance, 22+ SD use-cases were categorized with definitive names, like somatogravic and black-hole illusion, and human physicological vestibular thresholds were established such that flight maneuvers and/or speeds were restricted to prevent each of the SD use-cases [4], [5], [6]. The psychophysical human motion detection approach was to investigate behavioral response during varied situational stimuli for specific SD use-cases, with the goal of understanding a range of human response during isolated or mixed stimulus situations. Results assisted with clarifying the behavioral instruction map to prevent SD, and encouraged sensorial solutions to be developed such that human response during use-case exposure could be modified. For example, directional perception error was quantify in a realistic helicopter task where results could be used to improve behavioral instruction for preventing SD during abrupt landing [15]. Similarly, continuous heading detection perception was investigated using a compensatory task such that behavioral instruction for SD prevention could be improved for aircraft guidance [16]. Most recently, the individual and interactive influences of optical and gravito-inertial stimuli during simulated low-altitude flight demonstrated the importance of sensory integration effects on height perception using joystick response; results were intended to assist with sensorial solution selection for preventing SD [17]. The control theory approach to motion detection was to model typical human response during use-case stimuli and compare error between predicted and actual human response, making it possible to monitor motion detection in real-time [18]. Neuroscience and neuroergonomics approaches measured central nervous system mechanisms, including the brain and electrodermal activity, such that changes in physiological signals can assist in understanding neural mechanisms involved during SD. Insensate, unperceived, and perceived SD occurrence used different neural mechanisms, thus measuring these neural mechanisms would allow for SD detection [7]. These different perspectives of studying SD are useful and provide insightful information regarding human response in realistic contexts. However many of the mentioned studies research SD per use-case where a formalized ground-truth model or result is required, instead of trying to identify SD in a generalized manner. We believe that it is possible to measure general human activity during flight, and predict SD occurrence regardless of an SD use-case context using HAR modeling methods. Detection of aeronautical events and human activity using HAR methods has already been demonstrated. For example, disorientation was quantified and predicted using a joystick measure for an aerospace context [19]. Similarly, human movement activity was quantified by measuring the frequency and location of interacting agents [20]. Finally, from a human pose estimation approach, pilot activity was measured via hand, arm, and body positional movements' using a 3D camera [21].

In this study, we investigated SD human activity in two-parts; creation and validation of a generalized SD perception dataset using a motion detection inspired HAR experiment; identification of modeling parameters for reliable SD prediction using ML, DL, and unsupervised clustering methods. In Part I, SD was measured using a common piloting task with basic vestibular and visual stimuli, such that SD responses could be assumed or generalized for all SD use-cases. A whole-body rotational and translational vestibular stimulation task in darkness was inspired by motion detection experimentation, where automatic stimulation moved pilots around and along a three Cartesian coordinate frame axes respectively using a motion simulation system.

The HAR aspect required that pilots be loosely constrained in a natural piloting environment and actively compensate automatic stimulation motion, an act called compensatory tracking or dead-reckoning, using a joystick such that they remained stationary. Initial axial and directional joystick response was compared to automatic axial and directional motion stimuli, such that if initial response correctly counteracted the automatic stimuli, the trial-windowed dead-reckoning response period would be labeled initially correct implying non-SD; incorrect initial counteraction was labeled eventually or never correct implying non-SD or SD depending on the chosen ground-truth labeling convention. The main goal of the HAR experiments were to clearly label windowed time-series joystick data as non-SD or SD via initial compensatory response, thus creating an SD labeled dataset. Diverse use-cases for the labeled SD dataset were created by administering randomized combinations of three parameters that created the automatic angular or linear motion stimuli: axis, axis direction, and speed. Axis, axis direction, and speed influences on motion detection performance was performed using Exploratory Data Analysis (EDA) to confirm accurate recreation of naturalistic human response; gravitational influences on motion detection performance was investigated. In addition to automatically labeling time-series data, it was of interest to quantify the relationship between physical disorientation and motion detection performance because reports showed that physical health discomfort was another main cause of SD accidents [1], [4], [6]. Physical disorientation was measured before and after the piloting task, using the simulator sickness questionnaire (SSQ) disorientation subscale [22], [23]. We hypothesized that participants who initially detected correctly for the majority of the experimental trials, implying that the participant did not experience SD often, would have similar physical disorientation symptoms before and after the task. Implications for physical disorientation difference results are discussed. In summary, overall experimental motivation was not to identify vestibular thresholds and report motion detection behavior, like in controlled psychophysical motion detection experiments without a continuous task where choices and self-motion are limited. The goal of the Part I dataset creation study was to create a realistic and diverse labeled dataset for a continuous piloting task, measuring joystick dead-reckoning response with respect to SD occurrence, while identifying how to use physical discomfort measures and questionnaire data in an ML & DL modeling context. In Part II, endorsed HAR-literature model architectures, ML & DL supervised classification models, were tested using different data use-cases, feature-types, feature quantity, and ground-truth labeling to determine which modeling parameters promoted accurate SD prediction [8], [9], [12], [19], [24]. Selected ML & DL model architectures were: SVM, Long-Short Term Memory (LSTM), Multilayer Perceptron (MLP), Convolutional Neural Networks (CNN), LSTM-CNN, Transformer Encoder, and Random Forest (RF). Joystick dead-reckoning responses were used and transformed using derivative and spectral frequency methods to create a 27 column feature matrix of time, frequency, and time & frequency feature-types. The feature-type feature matrix and three ground-truth labels, derived from Part I initial detection performance categories, were used in three main model parameter selection studies: evaluation of model architecture and feature usage, ground-truth label comparisons, and unsupervised label comparisons with respect to ground-truth labels. The third main parameter study compared K-means, Gaussian Mixture Model (GMM), and K-medoids unsupervised methods with the three ground-truth labels via the rand score. Modeling parameter performance significance was discussed, and suggestions for future work and study limitations are mentioned.

II. RELATED WORKS

A. EXPERIMENTAL MOTION DETECTION

Vibration or motion, measured by the human vestibular system, conveys information about self orientation and position with respect to the environment. Motion detection is the act of discerning self-motion with respect to a reference in the environment [25]. Human motion detection and perception are quantified by stimulating the vestibular system systematically using different vibrational and motion experimental paradigms [26]. Early motion detection research before 2000 established human self-motion perceptual limitations; axial, axial direction, and speed/acceleration limits were referred to as vestibular thresholds or motion detection thresholds. Aeronautical applications required solutions for safe and efficient flight, thus early motion detection literature was strongly related to aeronautics; thresholds were reported in terms of acceleration instead of speed because flight instrumentation was in terms of acceleration. Earlier experimental paradigms were interested in self-motion perception during whole-body stimulation when; the axial motion stimuli trajectory had magnitude and/or frequency changes during fast or slow constant speed/acceleration; exposure time to motion stimuli lasted a long or short time, or was successively administered in a sequential manner; and head orientation was different or similar with self-motion [4], [27]. Modern motion detection research adopted robotic motion simulation, like the Moog 6-degree-of-freedom (DOF) motion platform; thresholds are reported in terms of speed because robotic motion planning is more reliable in terms of speed than acceleration [28], [29], [30], [31]. Both speed and acceleration motion detection thresholds are comparable because they are directly related with the derivative or integral function. Robotic motion simulation allowed for standardization of motion detection experimental design methods, and a systematic approach to progressively test relevant motion stimuli speed/frequency ranges. Modern experimental paradigms are interested in self-motion perception during whole-body stimulation, for context-driven situations concerning; non-constant speed for axial motion stimuli trajectories; vestibular dysfunction in comparison to healthy vestibular function; orientation and/or movement of the user's body during

exposure to stimuli; expertise in comparison to novice detection; and age [18], [28], [29], [31], [30]. Concerning SD, motion detection thresholds were used as an indicator of SD awareness [4], [5]. However, vestibular threshold values are not a continuously measured value with respect to a task and/or context, therefore they are less effective at identifying successive errors or trends that lead to SD, than a continuous physiological or human activity measurement. Similar to recent works, it was of interest to quantify SD occurrence with respect to continuous human activity measurements, because human activity has been shown to be a reliable predictive marker for real-world events like SD [19].

B. HUMAN ACTIVITY MEASUREMENTS

The force sensor, such as a joystick, is one of the first human activity measurements. A joystick is a stick-like input device that is omni-directional with respect to its supporting base, such that the angle and direction corresponds to motion control of an object. Joysticks are currently used for many applications, including applications in aviation, industry, military, and video gaming. Since the mid to late 1900s, human control using joysticks have been investigated in fields of human movement science in psychology and human-in-the-loop in automated control. Psychology and neuroscience fields were included in early HAR-like endeavors because the goal was to control a machine using a human activity measure like a joystick, thus the underlying mechanisms of human movement needed to be investigated with and without the usage of the human activity sensor. Psychophysical tracking experimentations, both pursuit and compensatory tracking, were proven ways to quantify human control performance using force sensors. Statistical analysis and automated control modeling of tracking behavior revealed that humans moved in a consistent manner, such that velocity and/or acceleration movements were modulated in order to perform a smooth position-based movement trajectory. Therefore, position, velocity, acceleration, and even the derivative of acceleration called jerk of human motion response were investigated to understand optimal human control of machines using joysticks [32]. A key realisation for joystick usage was that human operators could control positional outputs more smoothly and precisely when the velocity or acceleration of their angular and directional inputs were used; position, velocity, and acceleration controlled joysticks are used in many real-world applications like aviation. The intention of human activity measurements has changed with the advent of many types of affordable and portable sensors, that can be easily put in the environment and on humans [10]. Therefore, instead of studying human movement with respect to the sensor under different experimentally controlled scenarios, as was done in human movement science and human-in-the-loop control, researchers could investigate more real-world problems without sacrificing measurement accuracy. Thus, newly developed domains such as HAR and Neuroergonomics, stemming from traditional engineering & computer science and neuroscience fields respectively, have developed with purpose of quantifying and predicting human behavior in real-world settings using sensor fusion. Commonly used sensors in HAR are cameras, accelerometers, and gyroscopes; Inertial Measurement Units (IMUs), smartphones, video gaming technologies, and questionnaires are often used for data collection. Eventhough joysticks have been rigorously used and investigated as a human activity measure because they capture fine motor movements, IMUs are preferable in HAR because joystick devices capture 3D motion in a limited area bounded to the device base. Coupled IMU joystick devices used for gaming consoles capture both whole-body and small range limb/hand movements. IMUs can capture unbounded 3D motion, allowing them to be more suitable to capture movements over large distances and in areas with poor visibility and/or lighting. Activities such as walking, running, and stair climbing are often monitored using IMU devices. Despite the benefits of IMUs, IMUs can produce erroneous values caused by sensor drift, therefore sensor calibration before usage is necessary to minimize sensor error; sensor drift is the erroneous estimation of the Euler angles from the raw inertial data. In this study, we selected a joystick to measure human activity instead of IMU sensors and/or a camera because the joystick is an existing cockpit instrument that is an extension of the pilot; no addition sensors or tools would need to be installed or approved in real-world settings. Additionally, joystick usage was of interest because a joystick was shown to be an acceptable HAR measure that captured human disorientation behavior for an aeroespacial application [19].

C. ARTIFICIAL INTELLIGENCE METHODS FOR HAR

HAR data is typically in the form of time-series or images. Regarding time-series data, ML & DL models that accurately and sufficiently predict human activity are those that capture short to long range temporal dependencies. One of the best ML models proven to capture temporal causality, using raw time-series data, was SVM [8], [14]. SVM considers the entire feature space of the temporal data, thus temporal relationships are more likely to be found due to similar amplitudes. If time-series features are periodically repetitive, subspace models like RF are capable of capturing short-range temporal dependencies. DL algorithms improve prediction accuracy because they do not consider all of the data at one time, but they look at a window of temporal data only. Using precise windows of data these algorithms are able to better capture trends, both due to sequential order and amplitude, with respect to the given label. The best DL algorithms that capture causal information for HAR data, both raw time-series and transformed time-series data, are Recurrent Neural Network (RNN) such as LSTM, 1D and 2D CNN, and Transformer [9], [14], [24]. LSTM models were more effective than RNN because the cell architecture allows for past information within the specified window to be used for prediction, called a gating mechanism [33]. Moreover, 2D CNN has proven to have more predictive ability than the 1D CNN, due to the second dimensional space with respect to the neural network [12]. Depending on the feature,

LSTM maybe more effective at prediction than CNN, and vice versa. Finally, and most recently, Transformer models have been shown to reliably predict HAR activities. Transformers, like LSTM, window the time-series data thus produce predictions based on specific sequentially transformed pieces of data. Temporal, amplitude, and context similarity data aspects with respect to other features are evaluated, thus distinguishing data with respect to the corresponding label. Regarding image data, it has been shown that 2D CNN and Transformer architectures are more accurate than other architectures. Specifically for human pose estimation the Transformer model is able to decipher activity context with respect to previous frames better than 2D CNN [34]. Finally, hybrid architectures such as Transformer-CNN, CNN-Transformer, LSTM-CNN, and CNN-LSTM exploit both sequential and spatial aspects of the data. Regarding HAR accelerometer data, LSTM-CNN was shown to predict better than an LSTM [24]. Despite incremental improvements with hybrid model architectures, hybrid architectures are less desirable due to the unnecessary complexity of steps, and multi-processing architectures like Transformer are gaining popularity [9]. Previous works on prediction of human activity, specifically for IMU and image measures, have efficiently compared and reported model architecture performance with respect to feature-type/s [9]. However there is currently no systematic procedure for quantifying which feature-types contribute to reliable model architecture performance, for a less used HAR measure like the joystick. For example, joystick measures were low frequency measurements, where intentionally controlled human behavioral changes occur at time scales of 10Hz or less, in comparison to high frequency IMU measurements where human behavioral changes coupled with noise were far above 10Hz. It was uncertain whether reliable IMU model architectures, that needed high resolution features would also provide reliable predictions for smooth low resolution joystick features. Therefore, it was of interest to test a time, frequency, and time & frequency feature-type feature matrix on relevant model architectures, and quantify each model architecture's dependency on certain feature-types for a given time-series. In this work, we ensure successful SD prediction using low frequency joystick data by performing a systematic model parameter search to identify model architectures, feature-types, and other relevant model parameter attributes that allow for accurate SD prediction.

III. PART I: SD DATASET CREATION USING MOTION DETECTION EXPERIMENTATION

The goal of the Part I study was to create and validate a generalized SD occurrence dataset using a motion detection inspired HAR experiment and EDA respectively. The rotational and translational motion detection inspired HAR experiments were identically designed such that both experiments could be performed separately, while allowing for the final SD occurrence dataset to be in a standardized format.

A. EXPERIMENTAL DESIGN

Both rotational and translational experiments had a $3 \times 2 \times 2$ block design, where conditions were axis, axis direction, and speed respectively. The motion simulator system functioned using Cartesian coordinates, therefore experimental axis conditions for the rotational task were roll (RO), pitch (PI), and yaw (YA), and the translational task were left/right (LR), forward/backward (FB), and up/down (UD). In addition to the main axis stimuli, minuscule sinusoidal noise was added to non-stimulated axes because vibrational noise rendered a more challenging task by masking sound from the stimulated axis motor, while enabling a realistic vibrational aeronautical environment [25]. Sinusoidal noise had an amplitude and frequency of 1-2cm and >10Hz respectively. The axial direction experimental condition had two parameters: positive or negative direction. Figure 1 A depicts both the axis and axial direction conventions for both the rotational and translational experiments; the grey Cartesian coordinate frame represents the simulator cabin. The cabin could move in both rotation (RO, PI, YA) and translation

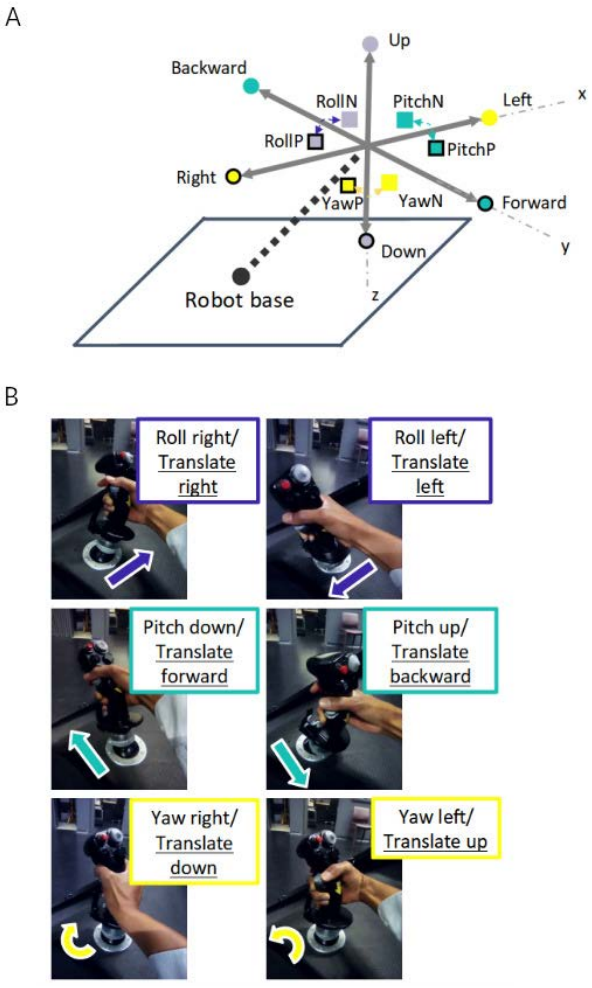


FIGURE 1. Axial and axial direction motion convention for cabin (A) and joystick (B).

(LR, FB, UD) via the input stimulus and/or participant control. The black outlined squares and circles in Figure 1A denote positive directional, rotational and translational movement (RollP, PitchP, YawP, Right, Forward, Down) respectively. Non-outlined squares and circles indicate negative directional movement (RollN, PitchN, YawN, Left, Backward, Up). Figure 1B shows the mapping of participants' joystick movements to the cabin movement. The speed experimental condition had two parameters; a slow ''near below-threshold'' (sub) speed where motion was difficult to detect and a fast ''above-threshold'' (sup) speed where motion was easier to detect. Sub speed motion detection was humanly possible, therefore this lower limit perceptual stimulation was emphasized to be at ''near below-threshold'' instead of at a specific below-threshold speed; below-threshold speeds are inhumanly possible to detect. Recent motion detection protocols were followed, such that robotic motion planning and motion stimuli were in terms of speed instead of acceleration [28], [29], [30], [31]. As mentioned in section II-A, speed parameters are referred to as motion detection thresholds and they are measured in terms of Hz, which is a frequency measure of deg/s or cm/s depending on whether the stimulus motion is in rotation or translation respectively. Rotational and translational, sub and sup speed selection was based on reported experimental design thresholds from motion detection literature that accommodated the motion constraints of the simulation system [28], [29], [30], [31], [27]. The Rotational and translation task sub & sup speeds were 0.5 Hz (deg/s) & 1.25 Hz (deg/s) and 3.75 Hz (cm/s) & 15 Hz (cm/s) respectively; implying that acceleration was constant at 0.5 deg/s^2 & 1.25 deg/s^2 and 3.75 cm/s^2 & 15 cm/s^2 respectively.

Figure 2A and 2B show a typical position trajectory when the participant did not respond and when the participant responded during phase a respectively, demonstrating that the experimental phases and trial length were dependent upon the participant's initial response. A single trial was composed of four different phases, as denoted by timeline B in Figure 2, in which participants were tasked to give feedback to specific visual and vestibular stimuli per phase. During phases a and b, participants could move the simulator using the joystick in any of the rotational or translational axes to counteract the perturbation. Joystick control was in terms of velocity control because it allowed for fast and smooth responses. Timeline A occurred when the participant did not respond in phase a, and it consisted of three phases: a Detection indicating motion stimulation of the cabin using a smoothed ramp forcing function, c Reinitialization denoting cabin reinitialization to the initial orientation or position, d Rest referring to the cabin and participant at rest. Timeline B occurred when the participant responded in phase a, the four phases consisted of: a Detection, b Active control referring to participant active control, c Reinitialization, d Rest. For both timeline A and B, visual and vestibular stimulation was given during each phase. The blue and red lines are position-based trajectories. The blue line denotes

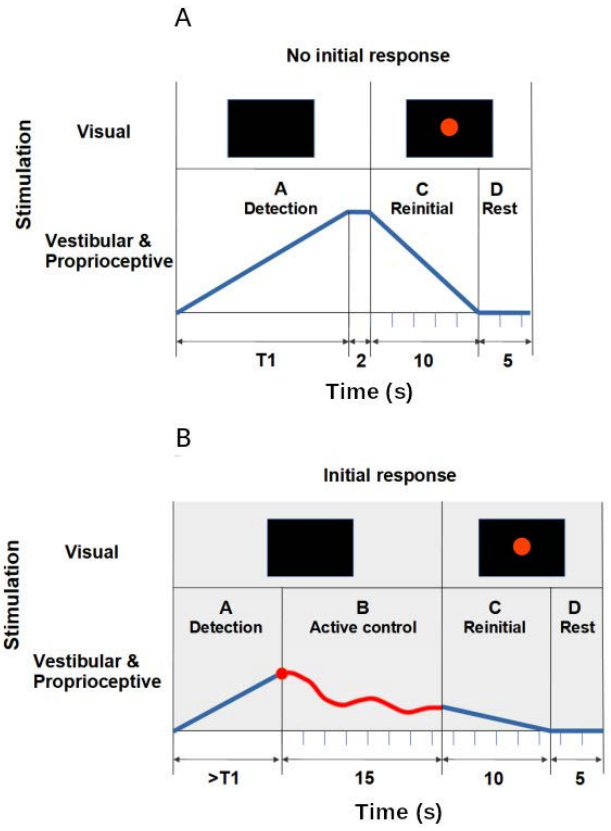


FIGURE 2. Experimental event timelines for when participants did not respond during phase a (Timeline A) and when participants did respond (Timeline B).

automatic robotic movement of the simulator cabin along one axis per trial, and the red line denotes the stimulus plus the participant's movements to compensate for the perturbation. T1 denotes the maximum allowed stimulation time per trial with respect to each axis and speed, if initial detection was not made within T1s the experimental phases followed as depicted in timeline A. If the joystick was moved within T1s, an initial response was registered and experimental phases occurred as depicted in timeline B.

- Phase a detection: A smoothed ramp-forcing function, where the rate of displacement was unknown to the participants, slowly and continuously perturbed one of the three rotational or translational axes of the simulator cabin at a sub or sup rate. Position trajectories are shown by the blue and red lines in Figure 2. During phase a participants were tasked to perform ''initial detection'', which consisted of identifying the axis and direction of the felt perturbation and manipulating a joystick replicating actual aircraft controls (Thrustmaster Hotas Warthog joystick), shown in Figure 1B, in the opposite direction of the stimulus. Participants had 15-20s to detect motion depending on the condition, denoted by T1 in Figure 2, which corresponded to the cabin reaching the maximum allowed cabin displacement for a particular axis and direction. T1 was different for every axis and experiment because sub and sup rates were

different for each experiment and the physical cabin displacement range was different for each axis. In particular, the rotational experiment had slightly longer stimulation times than the translational experiment because the sub and sup rates were slower and the available cabin displacements in the RO, PI, and YA orientations were larger than the available translational displacement ranges. If the participants did not respond within T1s during phase a, the cabin automatically displaced along one of the three axes as the ramp function increased until it reached T1s, where the ramp function maintained a zero slope causing the cabin to remain stationary for 2s.

- Phase b active control: If participants responded within T1s during phase a, phase b active control began and they had 15s to maintain the simulator orientation or position stably at the initial location by counteracting the perturbation; phase b was a vestibular dead-reckoning task. No visual stimulation was present; thus, the participants could rely only on vestibular and proprioceptive cues.
- Phase c reinitialization: A red dot appeared on the screen instructing participants to release the joystick and rest, while the cabin automatically returned to the initial starting location within 10s.
- Phase d rest: The cabin remained stationary at the starting location for 5s in order to avoid over-stimulation or after-effects.

In summary, the shortest and longest trials were approximately 32s and 50s respectively. The shortest trial length occurred when the participant immediately responded within 1-2s (2s + 15s + 10s + 5s) or did not respond such that T1 equaled 15s ((15s + 2s) + 10s + 5s), the longest trial length occurred when the participant responded just before T1 with T1 equaling 20s (19.9s + 15s + 10s + 5s). Both experiments administered 42 trials: 12 familiarization practice trials and 30 experimental trials. During the familiarization practice phase, unique experimental condition combinations were given, where each of the three axes was stimulated in negative or positive directions at sub or sup speeds. Similarly, the experimental phase consisted of 30 randomized trials, in which 15 trials with unique experimental conditions were repeated twice: five direction-speed conditions (negative sup, negative sub, no-movement, positive sup, positive sub) for each of the three axes (RO/LR, PI/FB, YA/UD). No-movement trials were included as sham trials to encourage the participants to remain active. Finally, in order to replicate a realistic flight scenario, the participants were free to move their head and body, looking and/or fixating where they wished, as long as it did not interfere with the task. The fact that the head was left unrestrained is considered undesirable, causing erroneous motion detection due to conflicting self-generated sensory information, and thus rarely performed in traditional motion perception experiments. However we considered it ecologically innovative and in alignment with HAR experimentation because it replicated human response under realistic flight circumstances, allowing for a more realistic SD occurrence dataset.

B. PARTICIPANTS

The EuroMov Institutional Review Board (IRB) at the University of Montpellier approved that the scientific objectives and organization of both experiments (IRB-EM rotational: 1703B, IRB-EM translational: 1704B) were safe and appropriate for human participation. The EuroMov IRB committee rules and regulations are in accordance with the 1964 Declaration of Helsinki and its later amendments. Eighteen and 14 healthy volunteers with normal or corrected vision gave informed consent before participating in the rotational and translational tasks respectively (males and females, 32 ± 10 years old); four of the 32 participants reported having novice time-limited piloting experiences lasting less than 40 hours. Four of the 18 rotational participants and four of the 14 translational participants were over the age of 40 years. The participants who performed the rotational experiment were not the same than those who performed the translational experiment, therefore, there was no confounds due to experimental ordering, learning, carryover, or fatigue effects. The same participant population, university students, and staff, were used for both experiments; therefore, it is likely that both experimental populations were similar.

C. EXPERIMENTAL PROTOCOL AND MOTION SIMULATION SYSTEM

The experiment took approximately 90 min and consisted of four sections (1) arrival, questionnaires, and instruction; (2) familiarization; (3) active control of rotational or translational stimulation; and (4) questionnaire and debriefing. Participants were instructed to maintain the cabin stationary at the initial trial position or orientation by compensating the axial stimulus. Axial stimulus dead-reckoning required that participants moderately move the joystick in the opposing axial direction as the felt stimulus; task performance strategies were encouraged. After describing the experimental task and completing the questionnaires, participants were securely installed using a safety harness and headphones for communication, as shown in Figure 3A. Once the participant was installed in the cabin, the cabin door was closed and all communication between the participant and experimenter was performed via a camera interface system that facilitated two-way auditory communication. The camera system also provided the experimenter visual feedback of the participant's upper body. The experimenter visually monitored the well-being of the participants, and confirmed participant's feelings of illness auditorily; the experiment ended if the participants reported physical illness. Two questionnaires were administered before the experimental phase: a claustrophobia assessment [35], [36] and SSQ [22], [23]. All questionnaires were administered in the native fluently spoken language of each participant (French or English). The claustrophobia questionnaire consisted of two sections: the first section measured fear of suffocation (14 questions) and the second section assessed fear of restriction (12 questions). The claustrophobia questionnaire was used as a screening method to assess whether participants could enter the

simulator and perform the task relatively stress-free; participants who scored 40 points or lower, indicating that they were not claustrophobic, were initially recruited, and participants scoring higher than 40 were recruited last. For both the rotational and translational experiments all participants scored ''non-claustrophobic'', rotational results were mean=10.94, max=38, min=0 and translational results were mean=8.77, max=9, min=0. The SSQ consisted of 16 questions and measured the participant's general physical state, evaluating nausea, ocular motor, and disorientation sub-scales. The SSQ was administered before and after the experiment to measure the effects of the experiment in terms of disorientation.

The motion simulation system that provided sensory stimulation, iMose, consisted of a 6DOF position-controlled KUKA-based motion simulator system (KR 500-3 MT adapted by BEC GmbH motion simulators, KUKA Roboter GmbH, Germany) and a local area network of three independent workstations [17], [37], [38]. Figures 3B and 33C show the interior and exterior of the simulation system, data was transferred between the simulator and workstations at 250 Hz using UDP. Workstations 1 and 3 were located in the experimenter control room; workstation 1 generated motion for the robot using a MATLAB/Simulink control interface program (MATLAB and Simulink Toolbox Release 2009, The MathWorks, Inc., Natick, Massachusetts, USA). Workstation 2 was fixed to the simulator cabin, and it administered the red dot or black visual screen and recorded participant joystick responses. Workstation 3, using Labview, served as the experimenter's user control interface to start and stop the experiment and collect experimental data without causing information delays between the workstations.

D. ANALYSIS

The analysis methodologies for dataset creation were to: 1) verify the correctness of experimental design execution using data standardization, 2) perform response categorization to label the joystick response data, 3) perform motion detection EDA to identify speed and axis influences on motion detection performance, 4) perform physical disorientation EDA to confirm whether other possible measures besides joystick response could convey markers for SD occurrence, and 5) summarize dataset performance by identifying behavioral trends. Python was used for all analyses, using numpy, pandas, scipy, seaborn, plotly, and matplotlib (Python 3.9, Python Software Foundation, Fredericksburg, Virginia, USA).

1) VERIFICATION OF SIMULATION DATASET

All trials, familiarization and experimental trials, were used in data analysis to maximize data usage. Automatic motion stimuli and participant joystick responses were downsampled from 250 Hz to 10 Hz for data analyses, such that only relevant human motor movements were considered; literature has shown that human hand and arm movements do not exceed frequencies of 10 Hz [32]. Data standardization pre-processing analysis was performed, using two-steps, to ensure that the data was properly collected. In the first step, correct experimental function was verified for three main items, using joystick response and cabin motion trajectories per trial; axis, axial direction, and speed labeling; joystick and cabin directional control convention, ensuring that joystick response opposed cabin motion; minimal joystick motion required to command cabin motion referred to as the deadzone. In the second step, the motion and timing of cabin motion with respect to joystick response were checked for correctness. The robotic simulator performed motion stimulation in real-time using a real-time Linux kernel, with a MATLAB/Simulink input layer, to capture responses with minimal delay. Despite the advantage of rapid response synchrony, real-time systems are prone to having system delays that can influence functional timing and communication between tasks; real-time functioning refers to the order in which numerical tasks are executed using the available computer resources. Therefore, the rotational and translational experiments had trials where system delays caused certain sequential events, like joystick and cabin response, to be executed in the wrong sequential order. Due to these slight processing and thus execution errors that are due to the real-time functionality of the motion simulator, it was necessary to remove all trials that had frequency or joystick-cabin related defects such that experimental defects were not confounded with participant response. The following defects, ordered from most to least prevalent, were checked in the second step of data standardization:

- insufficient trial length, trials where phases a and/or b were shorter than the minimal expected trial length of 17s, denoting the system sampling frequency was faster than desired,
- delays greater than 5s that prevented rapid cabin movement with respect to joystick response, or incorrect axial and/or directional cabin motion with respect to joystick response,
- trials where joystick motion was sufficient but the cabin insufficiently moved,
- temporal gaps in data.

In total, 40% and 50% of rotational and translational trial data was removed from the analysis, respectively. Functional and execution errors were expected because the system was a new experimental test platform, where many computers needed to operate in synchrony. Data standardization was the only step that removed trial data, trials that passed data standardization were used in data analysis.

2) RESPONSE CATEGORIZATION

Detection of correct stimuli was categorized into ten possible categories based on the selection of axis and axial direction. Figure 4 depicts a flowchart and possible participant choices based on response movements. The blue squares indicate the experimental trial type: the presence of motion stimuli denoted by ''Movement'' and no presence of motion stimuli denoted by ''Sham''. For ''Movement'' activity, the green squares indicate participant response activity such

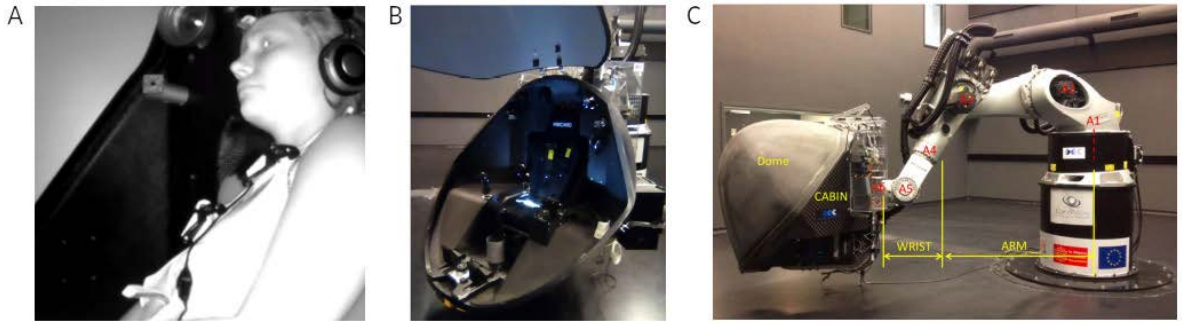


FIGURE 3. Motion simulator apparatus and installation; A and B show the experimental simulator cabin with and without a seated participant respectively. C shows an exterior view of the six-axis iMose motion simulator, consisting of the participant cabin and the robotic arm.

that ''1-3 axes'' means that the participant moved the joystick on one or more axes and ''No axes'' means that the participant did not move the joystick. The yellow diamonds denote the decision process based on the question asked within the diamond. For example, for ''Movement'' activity where the participant responded using one or more joystick movements, the following question is posed: ''Is the stimulus axis the same as the axis in which the participant initially moved the joystick?''. If yes, the axis was noted as correct, and the initial direction was confirmed in a similar manner. For example, ''Did the participant initially move in the opposite direction of the stimulus direction?''. The red numbers indicate the total number of possible categories based on the logical progression of performing the task correctly, first finding the correct axis and then finding the correct direction to counteract the vestibular stimulus. The ten detection performance categories were reduced to four categories:

- Initially Correct axis and direction: trials in which the first response was with the correct axis and direction (IC: Category 1),
- Eventually Correct axis or direction: trials where the first response was with an incorrect axis or direction but the correct axis and direction was found (EC: Category 2, 4, and 5),
- Never Correct: trials where participants acted on the joystick but never found the correct axis and/or direction (NC: Category 3, 6, and 7),
- No response: trials in which participants did not respond (NR: Category 9).

Categories 8 and 10 corresponded to the no-movement sham trials and were not used in the analysis.

3) MOTION DETECTION EDA

Motion detection EDA investigated three aspects, including speed, axis, and gravitational influences on motion detection performance. The normalized response count and Reaction Time (RT) per detection performance category were quantified for each axis and speed condition. The normalized response count was the adjusted count per response category, with respect to the given number of trials multiplied

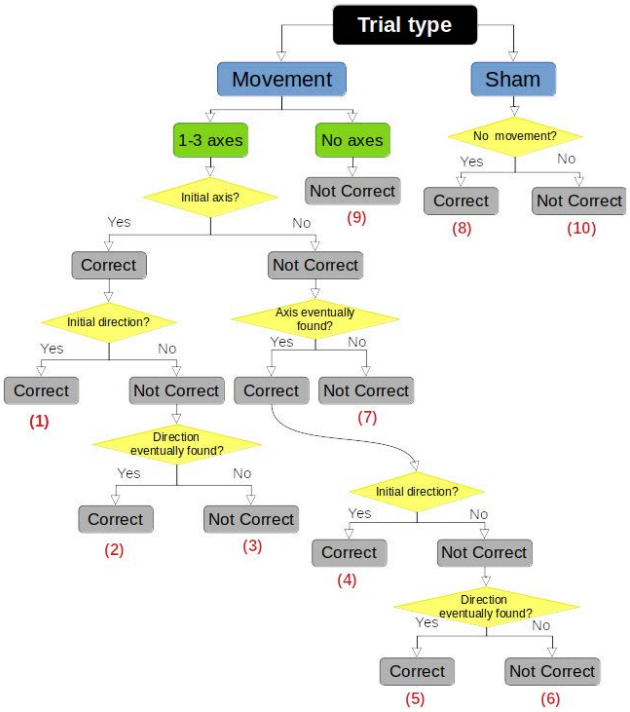


FIGURE 4. Flowchart of selection process for detection performance categories, where correct response categories 1, 2, 4, and 5 denote non-SD occurrence and wrong response categories 3, 6, 7, and 9 denote SD occurrence.

by participants; the total trial count per participant was 36, excluding sham trials. The total trial count per participant was adjusted to 36, such that the interpretation of results would be consistent with the experimental design. The total number of trials per participant was less than 36 because trials that did not follow the experimental design were removed during the data standardization step mentioned in Section III-D1. RT was the time that the participant used to find the correct axis and direction. The 95% confidence interval per axis was calculated to determine which detection performance categories were significant. Detection performance categories above the lower confidence interval were evaluated further. Significant and corresponding detection

performance categories were compared for the speed and axis. The Kolmogorov-Smirnov (KS) test was used to evaluate whether to use a parametric or non-parametric two-sample comparison test for within-axis and across-axis comparisons. All test evaluations resulted in non-parametric distributions; therefore, only non-parametric tests were used. Two non-parametric tests were used to evaluate comparisons: Wilcoxon signed-rank distribution test and Wilcoxon rank-sum distribution test [39]. Uneven two-sample non-parametric test data vectors were compared using the Wilcoxon rank-sum test. However, the Wilcoxon signed-rank test required that equal length vectors be compared, thus shorter length vectors were padded with NaN values to preserve the equivalent number of samples with respect to the longer vector and the distribution of the shorter length vector. Statistical p-values are reported using the following standardized significance levels: the Bonferroni required value of 0.0167 for two test comparisons, 0.05 for single test comparisons, and 0.001 for strongly significant one or two test comparisons.

4) PHYSICAL DISORIENTATION EDA

Detection performance categories were related to only the SSQ disorientation sub-scale, not the combined SSQ score, because the task was related to disorientation with respect to motion detection [22], [23]. Physical disorientation was monitored before and after the experiment using the SSQ disorientation sub-scale, such that the difference in before and after measures were attributed to the experienced task; SSQ disorientation difference equaled the disorientation score before the experiment, minus the score after the experiment. Negative SSQ disorientation difference meant that the task made the participant disoriented (e.g., they felt better before), and positive SSQ disorientation difference meant that the task rendered the participant less disoriented (e.g., they felt better after). Physical disorientation for accurate and non-accurate motion detection performers were compared, to quantify whether physical disorientation report could also be a marker for SD, like dead-reckoning joystick response. Again, Wilcoxon signed-rank or rank-sum non-parametric distribution tests were used to evaluate comparisons, as the KS test only found non-parametric distributions. The mentioned statistical p-value reporting convention was used.

5) PERFORMANCE SUMMARY

A participant detection performance rank score was created to compare overall participant detection performance with perfect performance. The performance rank score was calculated per subject across trials, per experiment, where

$$Rank\ score = 2 \cdot (IC\ count) + (EC\ count) \tag{1}$$

The rank score equation weights were arbitrarily chosen such that the equation formulation was most simplistic; RT was not considered in the rank score because rotational and translational experimental stimulation timings were different and thus non-comparable. IC performance was the desired behavior for the task so a weight of two was given to each IC trial. EC was also desired task behavior because participants were able to eventually find the correct axis and direction, however mistakes were made, thus a weight of one was given to each EC trial. NC and NR performance trials were not the desired task behavior so they were given no credit. Thus a rank score of 72 corresponded with perfect performance, where IC detection was performed for all 36 motion stimuli trials. Finally, the rank score was used to divide participants into three final categories in order to summarize performance with respect to each experiment. Mean and standard deviation of participants' rank score per experiment were calculated, such that participants were divided into best, average, and worst categories if their rank score was greater, within, and lower than one standard deviation from the experimental participant mean respectively.

E. PART I RESULTS

1) MOTION DETECTION EDA

For both rotational and translational experiments, EDA for motion detection behavior was quantified using count and RT per detection performance category, axis, and speed; no significant differences were found between positive and negative axial directions, thus axial direction differences were not considered. As mentioned in II-A, axial, axial direction, and speed/acceleration limits were referred to as motion detection thresholds. A motion detection threshold is registered from a self-report that motion was felt along a specific axis and axis direction for a specific motion stimulus frequency. Results are typically displayed in terms of mean detection count across or per subject for many stimulus motion frequencies, where count results are grouped by successful and unsuccessful detection [29], [30], [31]. We performed the same analysis presentation for sub and sup speeds/frequencies, displaying results in terms of count, mean count, and RT across participants; count and RT results were grouped by detection performance categories. Figure 5 shows the normalized summed count (top row), normalized mean count (middle row), and mean RT (bottom row) per detection performance category, across participants for RO, PI, YA, LR, FB, and UD axes and sub & sup speed conditions. The top row shows the normalized summed count per detection performance category for each axis and speed condition. For rotation, the summed bars in the top row are equal to 648, which corresponds to the 18 participants multiplied by 36 trials. The top row represents the distribution of total trial responses per response category. Similarly, for translation, the summed bars in the top row are equal to 504 which corresponds to the 14 participants multiplied by 36 trials. The middle row shows the mean count of the same normalized count data across participants. The mean count represents the frequency of selecting a response category across participants. It was necessary to show both mean and total selection count because average axis selection count could not be clearly understood from the total axis selection count; the total count gave information about overall

participant response and the average count gave information about participant tendencies. The bottom row displays the mean RT taken to detect correctly, thus only IC and EC response categories are shown. Bars without error bars indicate a single sample value, or several participants had the same count value. Single-sample bar values may exist due to data elimination during the rigorous standardization process. Wilcoxon signed-rank test and rank-sum tests were used to determine significance such that significant and slightly significant relationships were represented by (* within axis comparison of sub and sup, ** across axes comparison). Bonferroni correction: $p < 0.0167$ was used as the significance threshold. Detection performance categories above the lower confidence interval, denoted by the solid red line, were considered for statistical comparison across subjects for categories within (e.g.; sub vs. sup) and across (e.g.; RO sub vs. PI sub) axis conditions.

Motion detection speed performance was evaluated by statistically comparing sup and sub speed counts and RT, per axis and detection performance category, for both rotational and translational experiments. The most counted detection performance category for RO and PI axes, EC and IC respectively, had sup speed counts that were higher than sub speed counts; the blue stars in the top row of Figure 5 depict slight statistical significance for sub and sup differences (RO count EC sup vs sub: KS: non-normal distribution, signed-rank: $p < 0.001$, rank-sum: $p < 0.026$, n=13; PI count IC sup vs sub: KS: non-normal distribution, signed-rank: $p < 0.08$, n=18). There was a similar trend for the YA axis, where the most counted detection performance category, IC, had a higher sup count than sub count. No statistically significant differences between sub and sup speeds were found in the translational experiment. However, there was a trend for all axes where the most counted detection performance category for LR, FB, and UD axes, corresponding to EC, EC, and IC respectively, had sup speed counts that were higher than sub speed counts. Translational motion sub and sup speed differences were less apparent than in rotational motion due to inner-ear stimulation differences. Reduced speed detection in translational motion were likely attributed to less semi-circular stimulation and delayed otolith signaling in comparison to rotational motion [26]. Therefore, we suspect that more data was needed for differences to become statistically significant. Finally, regarding RT differences for the rotational experiment, some detection performance categories had significantly lower RT for the sup than the sub speed condition. The most counted detection performance category for RO and PI axes, EC and IC respectively, had lower RT for sup speed in comparison to sub speed (RO RT EC sup vs sub: KS: non-normal distribution, signed-rank: $p < 0.001$, rank-sum: $p < 0.001$, n=66; PI RT IC sup vs sub: KS: non-normal distribution, signed-rank: $p < 0.001$, rank-sum: $p < 0.001$, n=65); the blue stars in the bottom row of Figure 5 depict slight statistical significance for sub and sup differences. In summary, we demonstrated that faster sup motion caused more accurate and faster motion detection than slower sub motion. This result has been reported in motion detection literature, thus confirming that the experiments were performed correctly and that the dataset accurately represented human response [29], [31].

Motion detection axes performance was evaluated by statistically comparing counts and RT across axes, per speed and detection performance category, for both rotational and translational experiments. In particular, successful detection performance categories denoted by IC and EC, per speed condition, were compared across axes in order to determine which axis was more difficult than another; and thus demonstrate that SD dataset responses were in alignment with psychophysical motion detection findings. Two whole-body motion detection literature sources showed that RO and PI are easier to detect than YA and translational motion. In the first source, RO and PI detection thresholds were shown to be statistically similar for novices and experts [29]. Additionally, RO was reported to be easier to detect than LR, UD, and YA in both non-vestibular and vestibular dysfunction participants in the second source [31]. Table 1 depicts significant differences in response count and mean RT for speed and axis condition categories 1 and 2. Listing significant differences allowed us to rank axis conditions, with respect to motion detection ease and difficulty, and then compare the ranked list with literature reports to confirm correctness of experimental stimuli. Table 1 shows that, in alignment with literature reports, we similarly found that RO, PI, and FB axial motions were easier to detect than YA, with dependence on speed when considering only correct responses. In particular, successful response category counts for both RO and PI at sup speed were significantly higher than those for YA, and for sub speed FB and PI had significantly higher counts than YA. Moreover, our results showed functional differences between the RO, LR, & FB and PI, YA, & UD tasks. During PI participants mostly initially detected correctly (IC), and rarely when they did not initially detect correctly, they eventually or never detect correctly. Again in YA, participants often initially detected correctly (IC), but when they did not initially detect correctly they did not feel any motion and did not respond (NR). Similarly in UD, participants often initially detected correctly (IC), and when they did not

TABLE 1. Count and RT comparisons for combined IC & EC response.

	Speed & axis		Significance
	Category 1 (high count or fast RT)	Category 2 (low count or slow RT)	(KS: non-normal, signed-rank, rank-sum)
Counts	sup RO	sup YA	$p < 0.001$, 0.0167, n=20
	sup PI	sup YA	$p < 0.001$, 0.0167, n=20
	sub FB	sub YA	$p < 0.001$, 0.0167, n=22
	sub PI	sub YA	$p < 0.001$, 0.001, n=22
Mean RT	sup RO	sup YA	$p < 0.001$, 0.001, n=67
	sub PI	sub YA	$p < 0.001$, 0.001, n=67
	sub FB	sub UD	$p < 0.001$, 0.001, n=41

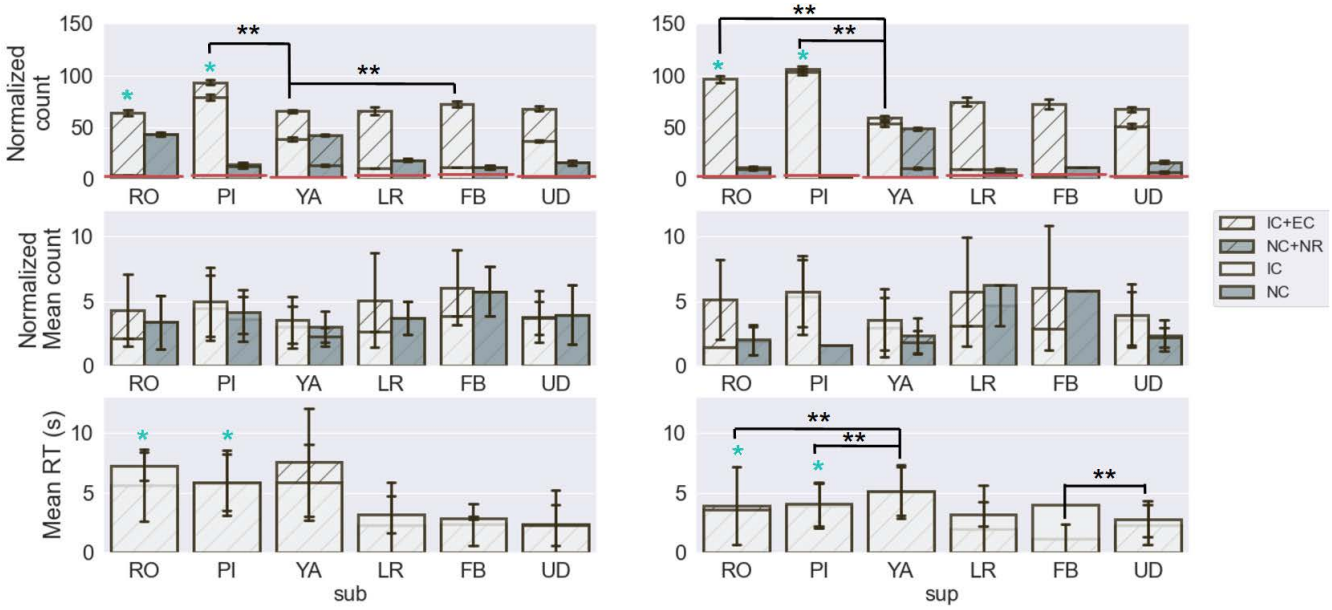


FIGURE 5. **Normalized summed count (top row), normalized mean count (middle row), and mean RT in seconds (bottom row) per detection performance category, axis, and speed for rotational and translational stimulation.**

initially detect correctly they often eventually corrected; IC was more prevalent when speed was fast. Whereas in RO, LR, and FB, participants could not initially detect the correct axis and/or axis direction, but they could eventually find the correct axis after several mistakes. Lastly, task difficulty for within rotational and translational stimulation appeared to be correlated with longer RT. As mentioned in Section III-A, participants were stimulated slower in the rotational task than in the translational task; thus, RT was different for the rotational and translation tasks and were not compared. The second portion in Table 1 labeled RT, shows the significant within experiment comparison across axes for correct IC and EC responses. For the rotational task at sup speed, RO and PI had faster RT than YA indicating that participants needed less time to detect motion for RO and PI. Similarly, for the translational task at sup speed, FB had significantly faster RT than UD. In summary, initial motion detection was least to most difficult for PI, RO, FB, LR, UD, and YA axis; this result was in alignment with motion detection literature. Contrary to conventional thinking, there was no significant sensory advantage for UD detection due to gravity because UD detection was challenging; it was likely that the vestibular system compensated additional gravitational information [31]. EC and IC initial detection response differences for RO, LR, & FB and PI, YA, & UD axes respectively, were likely due to the HAR experimental aspect that participants were loosely constrained and free to naturally move, thus causing ambiguous sensorial interpretation.

2) PHYSICAL DISORIENTATION EDA

Twenty of the 31 participants did not feel any difference in terms of physical disorientation during the entire experiment. Considering the performance rank score mentioned in Section III-D5, approximately 1/3 of the average detectors, 1/3 of the best detectors, and 2/3 of the worst detectors experienced physical disorientation. The 2/3 worst detection ratio is reported for completeness; however this measure is disregarded because it is based on only three participants. Thus, 1/3 of the population felt physical disorientation regardless of performance. To investigate whether there was a relationship between physical disorientation and detection performance, the detection performance of the 12 participants who reported physical disorientation was evaluated; see Table 2 for a percentage of their summed trial performance per category per SSQ difference report. For instance, a participant who reported a before and after SSQ score of six and four respectively would have their trial performance category counts, of eight EC and six IC trials, associated with an SSQ disorientation difference score of negative two. Table 2 shows the motion detection response category per reported SSQ disorientation sub-scale difference for both the rotational and translational experiments. Performance category percentage values across SSQ scores sum to 100%. Negative and positive SSQ values denote that the participant felt better before and after the task respectively. The bold percentages corresponding to negative SSQ values for categories EC and NC highlight that more negative physical disorientation was present in unsuccessful initial attempts to detect motion. Table 2 demonstrates that more negative physical disorientation was observed for unsuccessful initial detection response categories EC and NC, than for IC successful initial detection response or no response. The negative and positive SSQ disorientation differences per response category were summed respectively, to evaluate significance between IC negative and NC or EC negative. Physically disoriented best performers (IC) did not report significantly less physical disorientation

TABLE 2. SSQ disorientation sub-scale per motion detection performance category.

	SSQ (%)						
Category	-5	-3	-2	-1	1	2	4
IC (1)	5.4	8.4	17.8	23.6	24.5	8.9	11.5
EC (2,4,5)	10.1	12.0	29.0	26.6	14.1	2.8	5.4
NC (3,6,7)	10.3	2.7	38.5	26.3	5.2	11.5	5.5
NR (9)	0	3.4	20.8	18.7	26.0	21.8	9.4

than poor performers. In summary, no significant relationship between physical disorientation and motion detection was found. There was only a trend that EC and NC performers, who felt physical disorientation, felt better before the task than after. This implies that participants became fatigued while trying to perform the task, when detection was not easy for them. For IC performers who experienced physical disorientation, there was no trend in terms of feeling better before or after. Implying that participants who could detect easily, felt discomfort for other reasons not related to the experiment. There was a slight trend for NR performers that felt physical disorientation, such that they felt better after the task than before. Showing that participants who did not respond, became comfortable and relaxed in the dark experimental setting.

3) MOTION DETECTION PERFORMANCE RANK

Including both rotational and translational tasks, the highest rank score was 55 and the lowest score was 11. On average, participants received a rank score of 37. Therefore, the best performer, regardless of rotation or translation, achieved $(55/72) \cdot 100 = 76.3\%$ accuracy for the task. The average performer was only able to achieve $(37/72) \cdot 100 = 51.38\%$ accuracy for the task. The same task accuracy statistic was calculated for sub and sup conditions individually, for both rotation and translation experiments, and similar results were found, as shown in Table 3. Table 3 shows the experimental performance accuracy per speed condition using the performance rank measure. All percentages were calculated by dividing by 36 trials. These rank statistics showed that the detection task was challenging for the average person, regardless of the experimental conditions, but it was not impossible to perform with reasonable success. The participant distribution count for the rotational experiment was five best performers, 11 average performers, and two worst performers. Similarly, the participant distribution count for the translational task was as follows: two best performers, 11 average performers, and one worst performer. The rotational and translational participant distribution counts for best, average, and worst performance were similar, showing that both tasks were similarly challenging in terms of motion detection. Therefore, translational detection may not be more difficult than rotational detection in realistic environments.

TABLE 3. Detection performance rank per speed condition.

Rank	Rot sub	Trans sub	Rot sup	Trans sup
Best	83.3%	75%	80.5%	77.7%
Average	47.2%	55.5%	55.5%	58.3%
Worst	11.1%	0%	30.5%	25%

IV. PART II: SD PREDICTION USING ML, DL, AND CLUSTERING METHODS

The goal of the Part II study was to identify modeling parameters for reliable SD occurrence prediction using ML, DL, and unsupervised clustering methods, using the SD occurrence dataset that was experimentally created in Part I.

TABLE 4. Rotation and translation SD dataset organization.

Column attribute	Column number
Subject	0
Trial	1
Speed, Axis stimulus	2, 3
Data point count, **Time**	4, 5
Response Type (10 detection performance categories)	6
Commanded cabin position (RO/LR, PI/FB, YA/UD)	7,8,9
Actual cabin position (RO/LR, PI/FB, YA/UD)	10,11,12
Joystick (RO/LR, PI/FB, YA/UD)	13,14,15
Vibrational noise (RO/LR, PI/FB, YA/UD)	16,17,18

A. DATASET DESCRIPTION

The rotation and translation SD datasets contained 19 columns where the columns contained time-series data per trial; scalar values per trial were repeated for each corresponding trial. Table 4 shows column attributes with respect to column number; highlighted attributes were used for SD prediction. The subject, trial, speed, axis, time, response type, and joystick columns were pre-processed using the data preprocessing pipeline. Three ground-truth labels and feature-types were created; feature-types included time, frequency, and time & frequency features and human movement science inspired features.

B. ANALYSIS PIPELINE

The SD dataset was analysed using the following ML, DL, and unsupervised classification analysis methodologies: 1) supervised model architecture and feature selection & evaluation, investigating unique data use-cases and feature-types, 2) ground-truth label selection & evaluation, and 3) unsupervised model architecture and feature selection & evaluation. Python was used for the analysis pipeline, using numpy, pandas, scipy, pywt, tensorflow, scikit-learn, seaborn, plotly, and matplotlib (Python 3.9, Python Software Foundation, Fredericksburg, Virginia, USA). Modeling analysis was performed using jupyter-notebook with the PyPy3 Just-in-Time Compilation kernel for faster computational performance. Tensorflow models used standard parameter settings, such as early stopping, He initialization, and Adam optimisation; models were fit using 100 epochs and a batch size of 32 was used for all models, except for the Transformer Encoder model that used a batch size of 64.

1) SUPERVISED MODEL ARCHITECTURE AND FEATURE SELECTION

SVM, LSTM, MLP, CNN, LSTM-CNN, Transformer Encoder, and RF model architectures were identified as HAR-relevant models for predicting SD occurrence using dead-reckoning joystick response features. Each of the seven models were selected because they distinctively use feature data, as explained in subsection II-C. For instance, SVM and LSTM exploit temporal aspects of feature data. MLP is a fundamental modeling architecture for DL methods, and thus merited comparison for bench-marking reasons. CNN and LSTM-CNN models were selected because they exploit spatial aspects of feature data; similarly the Transformer Encoder architecture exploits both temporal and spatial components of feature data. Finally, RF was selected because it efficiently uses feature space to organize feature data. The seven model architectures are defined and discussed per paragraph; reasons for model selection are given with respect to HAR literature and used model parameter tuning are reported for result replication. The SVM model architecture is an ML method that distinguishes two or more classes by finding a bisecting line in feature space that separates the classes maximally. The feature vectors x_i and class labels y are known, such that the slope of the separable line w and y-intercept b are calculated using an iterative approach, where a cost function construction of w is minimized [8], [40]. SVM can distinguish unique classes of time-series data well because unique types of time-series are likely to have connected areas in feature space, thus boundaries can be found around these areas to distinguish certain temporal patterns from other temporal patterns. Despite numerous confirmations in literature that DL methods outperform SVM using time-series features, SVM was selected as a comparison method to survey its performance with respect to DL models for low-frequency joystick data and decomposed feature-types [14], [24]. The scikit-learn SVC model was used, where parameters C and gamma were automatically hyperparmeter tuned in an adaptive manner using accuracy, across batches of data points restricted to length of 70000; use-case data was assumed to be homogeneous. For example, C was initialized to the default value of 1 and gamma was initialized to a decimal value, the ratio of the number training features. If the current batch accuracy was lower than the previous batch accuracy, C and gamma would be increased and decreased by an incremental decimal value respectively; the incremental decimal value was 10 percent of the initialized values. Batch models were tested on randomly selected portions of test data, and the best predicting SVM batch model was selected to represent the data use-case. All data use-cases used the batch model hyperparameter tuning, with the exception of four use-cases where there was less data due to data removal standardization step. Due to the fact that the majority of models were adaptively trained, the accuracy and ROC-AUC values were average to best predictive representations of SVM. Restriction of feature space to 70000 points was not only motivated by hyperparameter tuning, but used such that SVC could reliably compute the result without excess usage of computational memory.

LSTM is a RNN DL method that uses a window of data points across feature samples to make an output estimate $\hat{y}$; a window is referred to as a batch and the number of data points in a window are called timesteps. Unlike an RNN, LSTM is able to learn long-term temporal dependencies using learned associations of past features, while avoiding the vanishing gradient problem via it's cell gating structure. An LSTM model consists of cells, also called nodes, where the number of cells are the number of timesteps that are sequentially connected from left to right. The first leftmost cell takes in the first timestep per batch and outputs three values: the cell memory state c which is a matrix containing the forward propagation values, the hidden activation state a which is the cell memory state transformed by the tanh activation function multiplied by a constant, and the output estimate $\hat{y}$. During the computation of each batch, the LSTM learns temporal information from cell-to-cell by passing a and c to the next LSTM cell to the right as an initialization, while using a new input timestep; thus a prediction per batch depends upon all of the timesteps [33], [41]. The LSTM architecture was selected because numerous HAR-literature reports indicated that LSTM captured causal information for HAR data well, as mentioned in section II-C. In the analysis, one Tensorflow LSTM model was used where the return state and return sequences were set to False, indicating that we wished a prediction per batch using timestep a and c estimates passed from cell-to-cell. The hidden state size called n_a was tuned in advance and 40 was found to produce the highest prediction accuracy results across data use-cases, for rotational and translational data.

A Neural Network (NN), also known as a MLP, estimates a probabilistic output $\hat{y}$ by solving the gradient descent linear/nonlinear optimization problem in a layered/nested manner such that at each layer, parameters called weights w and biases b are estimated from known individual inputs X and outputs y. A NN with more than two non-output layers is referred to as a DL model [40]. Specifically, known input features X are passed to a chosen number of nodes that each solve gradient descent in three steps:

- Forward propagation: $Z = wX + b$, $\hat{y} = A = f(Z)$ where w and b are initialized with selected random values. The activation function f truncates and bounds Z matrix values to a desired range, such that cost function $J(w, b)$ is likely to decrease. Relu and tanh are common activation functions used between non-output layers. The probabilistic estimated output $\hat{y}$ is computed by transforming Z using a sigmoid or softmax function for desired binary or multi-category classes respectively.
- Backward propagation: using the cost function $J(w, b)$ such that, four partial derivatives are computed $\frac{\partial J}{\partial A}, \frac{\partial J}{\partial Z}, \frac{\partial J}{\partial w}, \frac{\partial J}{\partial b}$,
- Update parameters: $w = w - \alpha \frac{\partial J}{\partial w}$, $b = b - \alpha \frac{\partial J}{\partial b}$ where α is known as the learning rate.

The number of layers and nodes per layer are chosen in a systematic search manner, called hyperparameter search, such that the output $\hat{y}$ is closest to y [41]. A basic NN was selected such that performance differences between a less complex DL model architecture could be compared with more complex DL HAR proven architectures. Concerning the performed analysis, two to eight layer Tensorflow MLP models with randomly selected nodes per layer were tested, such that the best performing MLP model was used for each data use case. Nodes per layer were selected as a function of the number of prediction classes, such that nodes decreased, increased, decreased then increased, or increased then decreased across layers. In general, 4 to 6 layer models with a decreasing node size per layer predicted best; Tensorflow Dense layers were employed for NN models.

CNN is a DL NN method that uses 2D spatial input information to estimate a probabilistic output $\hat{y}$ using the NN framework of forward and backward propagation. Stacked 2D matrix or stacked 3D image inputs are manipulated value-by-value or pixel-by-pixel using successive convolution, pooling, and dropout to enhance spatial patterns before and/or after NN optimisation is performed [40], [41]. Well-known shorter CNN architectures include Max Pooling CNN (MPCNN), Encoder-Decoder, whereas longer deep architectures include ResNet, Xception, and Inception. CNN was selected as a comparative model because HAR literature reported that CNN was a reliable method for predicting HAR activity; CNN architecture selection was inspired by existing architectures used for HAR [12], [24]. In the analysis, we tune the CNN Tensorflow representative model by performing both MPCNN and encoder-decoder architectures, and selecting the architecture with the highest accuracy for each data use-case and label. Feature signals were stacked on top of each other, and then reshaped into a 3D image of size 64×64. The encoder-decoder architecture consisted of: 64 filters 5×5 Conv2D with stride (2,2), LeakyReLU, Dropout 0.3, 128 filters 5×5 Conv2D with stride (2,2), Flatten, Dense layer output dimension with softmax or sigmoid activation for multi-class or binary prediction respectively. The following MPCNN architecture was used: 32 filters 5×5 Conv2D with stride (1,1), ReLU, MaxPooling2D pool size (2,2) with stride (2,2), 64 filters 5×5 Conv2D with stride (1,1), ReLU, MaxPooling2D pool size (2,2) with stride (2,2), Flatten, Dense layer 1000 with relu, Dense layer output dimension with softmax or sigmoid activation for multi-class or binary prediction respectively. The padding option called same was used for all models.

LSTM-CNN is a hybrid DL method that first performs LSTM to reduce inputs to binary patterns, such that similar sequential behavior is grouped, and then applies CNN on reduced spatial features. LSTM-CNN was selected because HAR literature reported that this modeling architecture was a competitive alternative to CNN [24]. Additionally, it was hypothesized that spatial binary groups of SD and non-SD, like a QR code, maybe more efficient as inputs than the raw time-series or transformed time-series features. LSTM and CNN modelization were similarly performed as previously mentioned individual models. After LSTM was performed, the reduced estimated output was reshaped into a 64×64 3D image, such that CNN was used to find spatial trends in the binary behavioral image.

The Transformer Encoder model consists of three main steps: positional encoding, multi-head attention, feedforward NN. In the positional encoding step, the feature matrix is added to a positional encoding matrix; the positional encoding matrix is created from unique feature point values that are mapped to shifted sinusoids. Next, Multi-Head Attention compares the positionally enhanced data points from a selected feature called query, with the other enhanced feature data points called keys. The similarity of query data points with respect to key data points are computed via Self-Attention, the relationship between data points via their value and position is referred to as the context. The feedforward NN then learns how the context corresponds to the output y, thus allowing for reliable estimations for y. Multi-head attention and feedforward are typically repeated six times, such that different combinations of query, keys, and values are selected, thus allowing for learning of different context representations with respect to the output y [33], [41]. Due to the fact that the Transformer Encoder model quantifies the value and positional relationship of each feature with respect to other features, it is widely used in applications where feature order is important for meaning such as text-processing, speech recognition, and image classification [34]. The Transformer Encoder architecture was of interest as a spatial comparison method to the CNN architecture because it compares information in parallel. In the analysis, the maximum number of unique encoded positions was set to the number of timesteps, the encoding/embedding dimension size was the number of features; input vocabulary size or unique data points was set to 10. The Standard Transformer Encoder modeling architecture was employed; Tensorflow functions Embedding, MultiHeadAttention, FullyConnected, LayerNormalization, and Dropout were used.

RF is an ensemble bagging method that combines the result of many weaker decision tree models into a single framework, by a process called voting where the mode prediction class is found per sample across all decision tree models. The decision tree method systematically divides the feature space into two subspaces at every decision criteria evaluation, such that the values in each subspace become more homogeneous. The goal of the decision criterion is to split the subspace at a location boundary where the difference between certain neighboring points are the largest. The stopping criteria typically consists of the three situations: the right and/or left subspace have less than a minimum number of data points, the maximum depth has been achieved, the data points in the subspace have similar/homogeneous values. RF is effective if decision tree models are not correlated, meaning that each model prediction is different such that voting results are not biased. Correlated models can be prevented by tuning the number of trees and using a reliable decision criteria like

entropy [40], [42]. HAR literature has shown that RF, similar to SVM, is a reliable ML method for predicting human activity. However, unlike many DL methods that use time-series features, RF requires feature transformation [24]. RF was selected as a comparison method to document responses for low-frequency joystick data and decomposed feature-types. Regarding the modeling analysis, default scikit-learn parameter selection was used for the RF model.

Feature matrix construction involved selection of three aspects; relevant experimental data referred to as use-cases; feature quantity; and feature-types. Data use-case, feature quantity, and feature-type were selected because they strongly influence model predictive ability with respect to data variability, feature matrix explainability, and feature characteristics in alignment with model function. Regarding data use-cases, six data use-cases per experiment, rotational and translational, were constructed from the speed and axis experimental parameters:

- general use-case: all speed and axes conditions,
- speed use-case: sup speed and all axes conditions,
- speed use-case: sub speed and all axes conditions,
- axis use-case: all speed and ax0 RO/LR axes conditions,
- axis use-case: all speed and ax1 PI/FB axes conditions,
- axis use-case: all speed and ax2 YA/UD axes conditions.

Speed and axis data use-cases served as a way to mimic SD use-cases in order to numerically determine whether currently practiced SD use-case evaluation is an effective strategy for SD prediction. Next, feature quantity construction consisted of testing predictive feature explainability using one, two, three, or all features; all total features consisted of 9 or 27 features depending on the feature-type. Feature explainability, also referred to as feature importance, quantifies the predictive relationship between feature/s and the target output. Finally, feature-type construction was motivated by two factors:

- exploitation of human movement science domain knowledge that humans regulate velocity and acceleration to perform position-based motions,
- investigation of time and/or frequency signal influences on model architecture.

HAR pose estimation typically uses position or image features, and HAR IMU accelerometer features are acceleration-only representations [10], [24]. However, the human movement science domain has proven that the brain requires derivative information in order for the body to generate smooth position trajectories [32]. Thus, implying that position, velocity, and acceleration motion trajectories convey unique and important temporal information about human motion. It was of interest to understand whether the additional derivative information would improve human activity predictions. Thus, first and second derivatives of the joystick signal were calculated; a third-order low pass filter with a cutoff frequency of 10Hz was applied to the second derivative. The position, velocity, and acceleration joystick time-series features are used in combination as opposed to position-only or acceleration-only features, that are typically used in HAR. In addition, we compare model performance using only position trajectories in comparison to position, velocity, and acceleration trajectories, to determine whether the additional derivative information would result in better model performance. As mentioned in subsection II-C, feature selection was also motivated by the need to categorize feature characteristic dependency with respect to model architecture, such that the feature and model selection process for SD prediction would be fast, easy, and well documented. Feature characteristics consisted of time, frequency, and time & frequency representations, for consistency feature characteristics are referred to as feature-types. The three feature-types were:

- time-only: joystick time-series signals in temporal order,
- frequency-only: five frequency pattern sublevels of the DWT using the symlets 5 mother wavelet [12],
- time & frequency: flattened 2D spectrogram formally called the short-time Fast Fourier transform (FFT) and the flattened 2D continuous wavelet transform (CWT) using the Mexican hat mother wavelet as reported in [12].

Moreover, position, velocity, and acceleration joystick features were transformed into time, frequency, and time & frequency feature decompositions, thus creating a total of 27 features; position only joystick features had 9 total features. Feature construction of the three aspects consisted of several pre-processing steps, called the pipeline, where time-series joystick dead-reckoning data was transformed into respective feature matrices. Joystick signals were baseline shifted to zero and constant zero joystick response trials were removed. All features and labels were linearly interpolated or downsampled such that all trials were 400 data points, equivalent to 40s long. As previously mentioned in section III-A, data trial length was different for each trial because participants initially responded when they perceived motion; shortest and longest trials were approximately 32s and 50s respectively. Forty seconds was chosen as an appropriate length because the average trial length was approximately 45s and total trial length needed to be divisible by multiples of 10, 20, 40, 100, 200, 400 for the data length evaluation analysis. Finally, the features were scaled using standardization such that values were scaled appropriately [40]. Once the feature matrix and labels were created for each data use-case, class balance oversampling was used for each of the three ground-truth labels. Classes with less label data were padded with respective class samples that were randomly selected in non-repeating order, such that each class had an equivalent and diverse representation of data samples.

Five main analysis were performed to test both the selected model architectures and feature constructions: model architecture performance evaluation using data use-cases, model architecture performance evaluation using feature quantity, model architecture performance evaluation using permutation importance (PIM) feature-type, model architecture performance evaluation using human movement science feature-type, and LSTM required data quantity.

Model architecture and feature usage evaluations investigated which ML & DL model architectures had better prediction accuracy and/or ROC-AUC for different data use-cases, feature quantity, and feature-type. Concerning the feature-type evaluations, a feature-type dependency scored was created from PIM such that feature-type dependency was quantified for each model architecture. Using the feature-type dependency score we confirm that model architectures are designed to process certain types of feature data. For example, sequential models are designed to identify trends in time-based features and CNN-based model are designed to identify spatial features [33], [40], [41]. In addition, the dependency score provided information about non-intentionally designed feature-types with respect to model architectures, thus allowing for better fundamentally understanding of model architecture function. After identifying the best performing model architecture, data length usage was investigated such that the ideal data quantity that best predicted SD was found.

2) SUPERVISED PERFORMANCE METRICS

Supervised classification analysis used model accuracy and ROC-AUC metrics to evaluate model performance, feature performance was evaluated via the permutation importance (PIM) metric. Individual metric comparisons, of the three metrics, were evaluated using the Wilcoxon signed-rank or rank-sum tests where $p < 0.05$ and $p < 0.001$ were considered significant and strongly significant respectively; only non-parametric tests were used because the KS test reported non-parametric distributions. Accuracy measured the true positive (TP) and true negative (TN) counts over the total number of samples; a value of 1 and 0 correspond to 100% and 0% correct prediction. Accuracy only gives information about how well the model approves data, but not about how well the model rejects data.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

where FP and FN correspond to false positive and negative counts, respectively. The ROC-AUC measure was used to evaluate both classification acceptance and rejection performance. ROC-AUC is the area under the false positive rate (FPR), shown in equation (3), versus the True Positive Rate (TPR), shown in equation (4),

$$FPR = \frac{FP}{FP + TN} \tag{3}$$

$$TPR = \frac{TP}{TP + FN} \tag{4}$$

An ROC-AUC score of one indicates perfect prediction of all labeled classes, whereas a score of 0.5 or lower indicates that prediction of all labeled classes was poor with chance level performance or lower. ROC-AUC was needed in addition to accuracy to determine if FP values were balanced with TP values, ensuring that the SD model could accurately reject and accept the data [40]. AUC and roc auc score metrics in TensorFlow and scikit-learn were used respectively. The accuracy metric in both TensorFlow and scikit-learn were used. Average 5-fold cross validation test prediction accuracy and ROC-AUC measures were used to evaluate model performance.

Feature importance was of interest because each feature contained distinct information about disorientation. It was of interest to understand which feature/s could convey the most informative information about the occurrence of perceptual disorientation. Feature importance was calculated such that each feature was shuffled individually and model accuracy was calculated for each shuffled feature. Unshuffled model prediction accuracy was subtracted with each of the shuffled feature prediction accuracy scores. The change in prediction accuracy for each shuffled feature was ranked, such that the feature with the largest change in prediction accuracy was considered the most important feature. The scikit-learn PIM function was used for SVM and RF models, PIM was calculated manually for the MLP, LSTM, Transformer, CNN, and LSTM-CNN Tensorflow models. A dependency score was constructed to evaluate each model architecture's dependence on three feature-types, the score was calculated by first obtaining the minimum number of used feature-types. For example there were three, 18, and six time, frequency, and time & frequency features, therefore three was selected as the minimum number of feature-types and only the best three features from each group were used. A ratio was constructed such that the minimum number of feature-types, which was 3, was divided by the sum of the first three permutation important ordered features, per feature-type.

$$dependency\ score = \frac{3}{\sum_{rank=0}^{2} PIM\ rank}, \tag{5}$$

where the dependency score was calculated for each feature-type and the PIM rank corresponded to the first three features for each respective feature-type. A value of 1 signified that the permutation important feature-types changed model predictions during PIM thus the model was strongly dependent on these features-types. A value closer to zero indicates that the feature-type did not strongly change model prediction during PIM, and thus the model weakly depends on these features-types.

3) GROUND-TRUTH LABEL SELECTION

The data pre-processing pipeline prepared features and labels per data use-case such that they could be used for SD supervised and unsupervised classification analyses. There was uncertain about how to quantitatively define SD, therefore it was of interest to investigate several plausible quantitative definitions of SD, called SD ground-truth labels. Three ground-truth labels identifying SD occurrence were created, based on the identified performance categories IC, EC, NC, and NR in Part I. The three SD ground-truth labels were:

- Lenient: a binary label denoting SD for NC and NR performance categories and non-SD for IC and

EC categories, implying that small occasional mistakes did not signify SD whereas successive errors signified SD,
- Strict: a binary label denoting SD for EC, NC, and NR performance categories and non-SD for the IC category, implying that small occasional mistakes and successive errors are likely to be SD and only non-SD occurred when performance was perfect,
- Complex: a multi-category label depicting SD via NC and NR responses, mild-SD using EC responses, and non-SD using only IC responses.

The purpose of testing different labels was to understand how to best define SD from the intrinsic organization of the data; better predicting models using a certain label implies that the data is best structured for that label. We compare our data-driven definition of SD with the current functional definition of SD [6]. SD identification label effectiveness was evaluated using model mean accuracy, where high mean accuracy signified that the label described the data well. Ground-truth label identification demonstrated that a numerically derived definition of SD was possible. Each ground-truth label was ranked from most to least appropriate based on highest to lowest model mean accuracy respectively.

4) UNSUPERVISED MODEL ARCHITECTURE AND FEATURE SELECTION

K-means, K-medoids, and GMM unsupervised clustering methods were used to investigate SD. K-means is a recursive method that groups feature data X into k chosen number of clusters using the minimum Euclidean distance, from each centroid to each X sample, as a measure to assign samples of X to each group; centroids can be randomly assigned [40]. K-medoids, known as Partitioning Around Medoids (PAM), is also a recursive method that groups feature data X into k chosen number of clusters using the minimum sum of pairwise dissimilarities, as a measure to assign samples of X to each cluster object called a medoids; centroids must be assigned to X sample points [43]. Unlike hard clustering methods, like K-means and K-medoids where each example is assigned to a unique cluster, GMM is a soft method that gives each example a membership score quantifying a probabilistic associated with each cluster [40]. K-means, K-medoids, and GMM were selected because they are reliable methods that use unique sample assignment strategies. Data with different feature space organization and scaling, like the rotational & translational task and even expert pilot trajectories, can be reliably clustered. Multiple unsupervised methods were tested in order to improve the possibility of finding an unsupervised label similar to each ground-truth label. Unsupervised method labels were compared with SD ground-truth labels, using six different feature matrix combinations:

- joystick position
- joystick position & velocity
- joystick position, velocity, & acceleration
- joystick position, & two joystick position DWTs
- joystick position, & six joystick position DWTs
- three PCA components generated from joystick position, velocity, & acceleration

These six combinations were selected because they combined position joystick trajectories with unique representations of joystick behavior. Position combined with derivative representations were motivated by human movement science findings where position movements depend on velocity and acceleration information. Combinations of position and the DWT were used in order to create a feature space with both important temporal and frequency representations of joystick behavior. Finally, PCA was used on position and derivative representations in order to create a feature space with important temporal representations of joystick behavior. PCA is a dimensionality reduction technique where data is transformed into a new coordinate system where data is organized from highest to lowest data variance [40]. Scikit-learn was used to calculate K-means, GMM, and PCA methods; the scikit-learn-extra package was used to compute K-medoids. The rand score performance metric, also referred to as the rand index, was used to evaluate similarity between unsupervised labels and ground-truth labels. The rand score was selected because it is a simple and reliable method for comparing labels with the same number of clusters, regardless of cluster ordering assignment. The rand score is the ratio of the number of agreeing label pairs with respect to the number of label pairs. The scikit-learn rand score function was used for all unsupervised label comparisons.

C. PART II RESULTS

1) SUPERVISED MODEL ARCHITECTURE AND FEATURE EVALUATION

Model architecture predictive ability was evaluated using classification accuracy and ROC-AUC score on a test dataset. Figure 6 shows test dataset prediction accuracy for the selected model architectures, using the 27 position, velocity, & acceleration feature set. Colored points depict model accuracy values for each data use-case, ground-truth label, and feature quantity. Each model architecture and data use-case per ground-truth label tested four different feature quantities, thus corresponding to four points per model on each line. The mean model test accuracy from greatest to smallest was LSTM, RF, Trans, SVM, MLP, CNN, and LSTM-CNN with mean accuracy of 0.84, 0.82, 0.77, 0.67, 0.58, 0.45, and 0.44 respectively; the average for each model was computed across experiments, data use-cases, ground-truth labels, feature-types, and feature quantities. The following five paragraphs quantify relationships between model architecture, data use-case, feature quantity, PIM and human movement science feature-type, and LSTM data quantity usage in detail. For all analysis the 27 position, velocity, & acceleration feature set was used; the 9 position feature set was compared with the 27 feature set for human movement science evaluations.

SD prediction accuracy and ROC-AUC was evaluated for speed and axis data use-cases in comparison to using all the data, to determine whether SD modeling was more effective

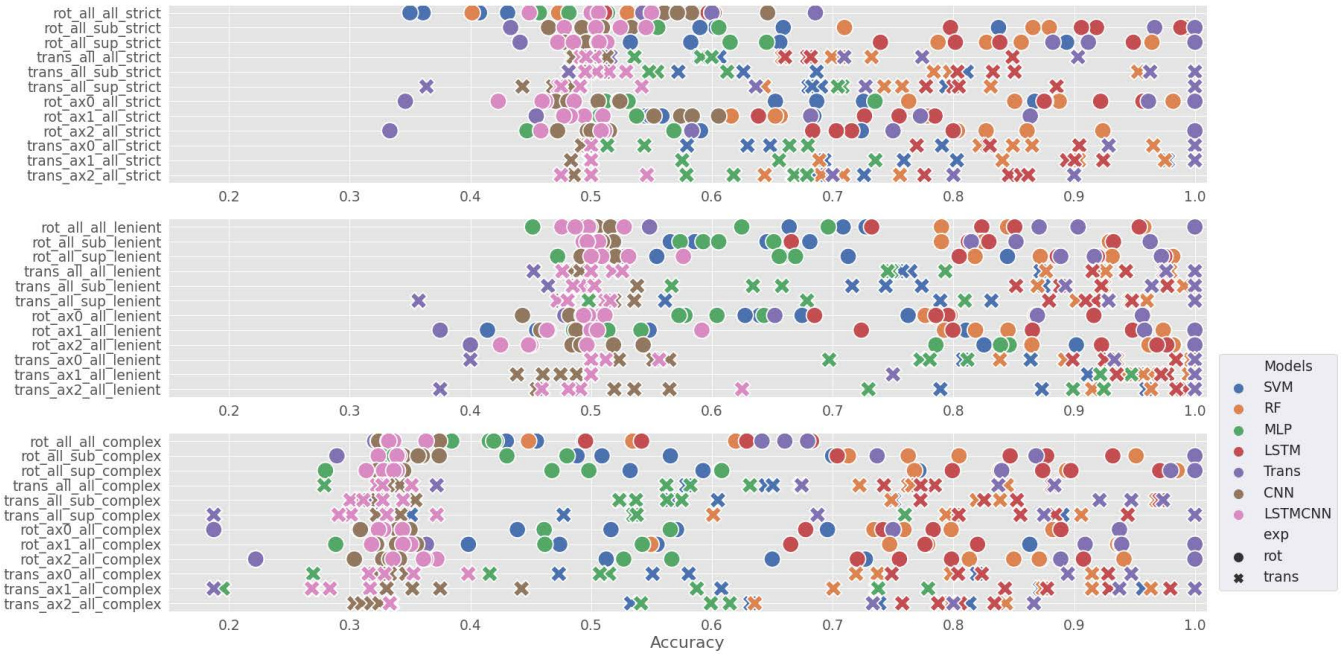


FIGURE 6. Test prediction accuracy for the selected model architectures using the position, velocity, & acceleration feature set for different: data use-cases, ground-truth labels (strict, lenient, complex), and feature quantity (one, two, three, or all features) used. For visual ease, circles and crosses correspond to the rotation and translation experiment types, respectively. Blue, orange, green, red, purple, brown, and magenta correspond to the SVM, RF, MLP, LSTM, Transformer, CNN, and LSTM-CNN models. Regarding notation, ax0, ax1, ax2 indicate RO, PI, YA for the rotational task and LR, FB, UD for the translational task respectively; 'all' after the first underscore refers to all axis trials. Additionally, sup and sub denote sup and sub speed stimulation trials respectively; 'all' after the second underscore refers to all speed trials.

by use-case. Model performance, accuracy & ROC-AUC, for the speed use-case and general data usage were statistically compared across feature quantity and model architectures, no significant differences in accuracy and ROC-AUC were found for each ground-truth label. Similarly, model performance for the axis use-case and general data usage were statistically compared across feature quantity and model architectures. No significant differences in ROC-AUC were found for each ground-truth label, however accuracy was significantly higher for the translational lenient label use-case (mean accuracy: 0.84) than general data usage (mean accuracy: 0.79), regardless of model architecture (translational lenient, axis vs all data: KS: non-normal distribution, signed-rank: $p < 0.05$, n=7). These results show that if several model architectures are used for SD prediction, better prediction results for data use-cases depend on the label and the use-case. Model performance for speed or axis data use-cases were statistically compared with general data usage model performance, across feature quantity and ground-truth labels; no significant differences in accuracy and ROC-AUC were found for model architectures. This result shows that regardless of the SD label tested, each model architecture could predict use-cases and the general use-case similarly. Therefore, using a special use-case is not an effective strategy for SD prediction because there was no significant prediction performance improvement with respect to using all of the data. In summary, constructing a general model including all use-cases will create a reliable SD predictive model. However for certain use-cases with appropriate label selection, SD prediction can be significantly improved by modeling with use-case data only.

The goal of this analysis was to investigate whether specific model architectures required a certain number of features to reliably predict SD. Initially, PIM was calculated for each model architecture, data use-case, and ground-truth label, to identify which features were most important. Next, the minimum number of needed features per model architecture was evaluated using two methods: PIM and comparison of accuracy for 'all', 'top3', 'top2', and 'top1' models. Models constructed with 27, three, two, and one feature/s were referred to as 'all', 'top3', 'top2', and 'top1' models, respectively. Figure 7 shows each model architecture, data use-case, and ground-truth label with respect to the three most important features. The mean and standard deviation of PIM per model architecture was calculated, regardless of data use-case and ground-truth label, and features above the mean plus one standard deviation were counted as a required feature. The mean required feature count per model architecture showed that Transformer, MLP, SVM and RF required at least 6, 4, 4, and 2 features in order to have reliable SD prediction. LSTM, CNN, and LSTM-CNN had minuscule fluctuating differences from baseline measures when columns were individually permuted during PIM, therefore it was unclear how many features were required by these models; features were ordered from most important to least important but values with respect to baseline were too small for comparison.

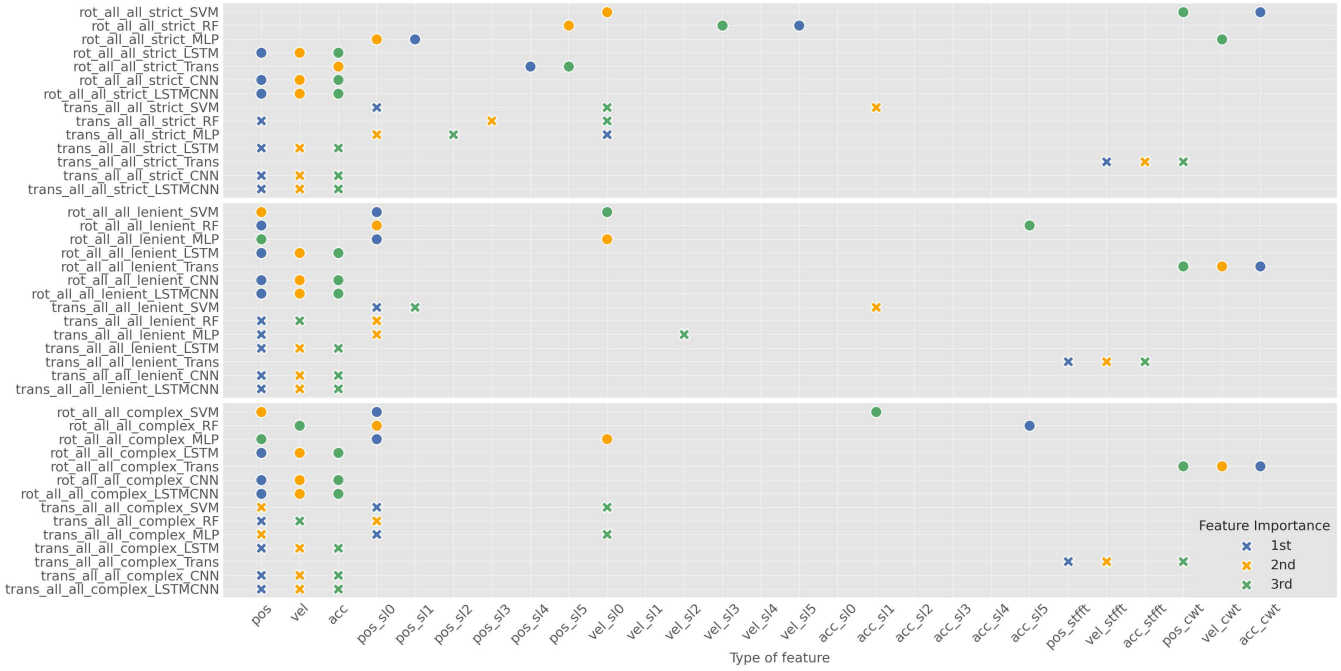


FIGURE 7. Feature order importance based on PIM for each model architecture, data use-case, and ground-truth label. Blue, orange and green indicate the first, second, and third most important feature respectively. The circles and crosses indicate rotation and translation experiment types, respectively.

Due to the fact that all models could not be compared with PIM, a second previously mentioned method was used to evaluate minimal required feature quantity. Prediction accuracy was compared for model architectures using more to less features, 'top3' models reported the highest accuracy for all model architectures. For the rotational task, the model architectures that were most to least accurate were Transformer, LSTM, RF, SVM, MLP, CNN, and LSTM-CNN where accuracy was 0.76, 0.74, 0.66, 0.55, 0.53, 0.5, and 0.47 respectively. Similarly for the translational task, the model architectures that were most to least accurate, were Transformer, LSTM, RF, SVM, MLP, LSTM-CNN, and CNN where accuracy was 0.93, 0.9, 0.85, 0.71, 0.66, 0.46, and 0.44 respectively. This result shows that three features are likely to be sufficient for reliable prediction of SD. Moreover, Transformer, LSTM, and RF models are more accurate at predicting SD with three features than other model architectures.

For both rotational and translational experiments using all the use-case data, certain model architectures consistently had PIM ranking where certain feature-types were ranked as most important, therefore feature characteristic dependency was quantified and categorized per model architecture. The amount that model architectures depended on time, frequency, and time & frequency feature-types was quantified via the PIM-based dependency score. Table 5 shows the time, frequency, and time & frequency mean feature dependency score and mean accuracy for each model architecture across ground-truth labels; rotational results are shown above translational experiment results and all the data for speed and axis conditions was used. SVM, RF, and MLP models largely depended on DWT frequency-only features, with a dependency score ranging from 0.5 to 0.75. LSTM, CNN, and LSTM-CNN models strongly depended on time-only features, with the maximum dependency score of 1. The Transformer architecture depended on time & frequency features, the short-time FFT and CWT, with a dependency score of 0.66. It is likely that SVM, RF, and MLP model architectures depended on frequency features because repeated groupings in feature space caused by periodic wavelets were likely to be easier to distinguish with respect to the label for pure gradient descent dependent architectures. Time features with unique values were likely to facilitate more accurate predictions for LSTM, CNN, and LSTM-CNN because the LSTM gating structure requires distinctive data to distinguish long-term dependencies, and CNN smoothing methods like convolution and pooling are only effective on detailed or finer resolution data. The Transformer model was likely to depend on

TABLE 5. Model architecture dependency on time, frequency, and time & frequency features.

Model	feature-type	time	freq	tf	mean acc
SVM	freq	0.11	**0.5**	0.26	0.55
RF	freq	0.16	**0.7**	0.07	0.66
MLP	freq	0.1	**0.75**	0.1	0.53
LSTM	time	**1**	0.25	0.04	0.74
CNN	time	**1**	0.25	0.04	0.49
LSTM-CNN	time	**1**	0.25	0.04	0.44
Trans	tf	0.09	0.31	**0.69**	0.76
SVM	freq	0.11	**0.87**	0.06	0.71
RF	freq	0.16	**0.42**	0.05	0.85
MLP	freq	0.2	**0.7**	0.06	0.63
LSTM	time	**1**	0.25	0.04	0.9
CNN	time	**1**	0.25	0.04	0.44
LSTM-CNN	time	**1**	0.25	0.04	0.46
Trans	tf	0.05	0.21	**0.67**	0.93

TABLE 6. Ground-truth label comparison.

	Model Accuracy						
Label	LSTM	Trans	RF	SVM	MLP	CNN	LCNN
Rot L	**0.8**	**0.8**	**0.9**	**0.7**	**0.6**	**0.5**	**0.5**
Rot S	0.6	0.6	0.5	0.4	0.5	0.6	0.5
Rot C	0.6	0.6	0.5	0.4	0.4	0.3	0.3
Trans L	**0.9**	**0.9**	**0.9**	**0.8**	**0.7**	**0.5**	**0.5**
Trans S	0.7	0.7	0.7	0.6	0.6	0.5	0.5
Trans C	0.8	0.7	0.8	0.6	0.5	0.3	0.3

time & frequency features because they had sparse simplistic representations with minute differences, thus allowing for the model to learn ordering and value information with respect to the label.

Model performance, accuracy and ROC-AUC, for models composed of 27 position, velocity, and acceleration features were compared with models composed of 9 position only features, in order to determine whether derivative components of joystick would improve prediction performance. Accuracy was not statistically greater when position, velocity, and acceleration features were used in comparison to position only features; there was slight significance for rotational Transformer and translational RF models (accuracy for position, velocity, & acceleration vs position only : KS: non-normal distribution, signed-rank: $p < 0.05$, n=12). Additionally, ROC-AUC was statistically greater when position, velocity, and acceleration features were used in comparison to position only features for rotational MLP models (ROC-AUC for position, velocity, & acceleration vs position only : KS: non-normal distribution, Bonferroni required value of signed-rank: $p < 0.0167$, sum-rank: $p < 0.0167$, n=12). These results show that including derivative position features are likely to improve predictions, however improvements are not likely to be strongly significant.

The best performing model architecture, LSTM, was used to investigate the ideal quantity of data per prediction, called timesteps, required to reliably predict SD. One, two, four, 10, 20, and 40 second data timesteps were tested for each ground-truth label and feature-quantity using general data usage, such that the ideal timestep was selected based on highest LSTM prediction accuracy. Tested timesteps were counted across ground-truth label and feature-quantity, the most counted timestep was 20 and 4 seconds for the rotational and translational tasks respectively. Mean LSTM accuracy for 20 and 4 second timestep models were 0.7 and 0.85 respectively, demonstrating that the entire 40 second trial was not needed to obtain reliable SD prediction.

2) GROUND-TRUTH LABEL EVALUATION

Table 6 shows the mean accuracy per model architecture and ground-truth label, for general use-case data and regardless of feature quanity. Rot and Trans refer to the rotational and translational experiments, and L, S, and C denote the lenient, strict, and complex ground-truth labels respectively. The lenient ground-truth label, in bold, produced the highest mean model accuracy results regardless of model architecture for both rotational and translational experiments. Considering the lenient and strict binary ground-truth labels, the lenient label was hypothesized to produce better predictive results than the strict label because the lenient label allowed for more samples to be labeled 'non-SD' than the strict label. In particular, the lenient label allowed for initial mistakes to be made, thus facilitating balanced class selection that allowed for a better numerical representation of both 'non-SD' and SD trials. The strict ground-truth label had less 'non-SD' samples because the novice participants had difficulty detecting motion; as mentioned the best performer could only accomplish 71% of the task correctly. The strict ground-truth label convention was inspired by the setting where pilots are required to make minimal to no mistakes, if participants had more expertise the number of 'non-SD' trials maybe similar to 'SD' trials like the lenient label. Finally, model accuracy for complex ground-truth labels would probably be higher if more data was available, multi-class labels require more training data than binary classification [41]. These results show that the lenient ground-truth label best identified SD for novice participants.

3) UNSUPERVISED MODEL ARCHITECTURE AND FEATURE EVALUATION

K-means, GMM, and K-medoids unsupervised clustering methods were used with six different feature matrices, for comparison with each ground-truth label. The feature matrices were: position, position/velocity, position/velocity/acceleration, position/two position Discrete Wavelet Transforms (DWTs), position/six position DWTs, Principle Component Analysis (PCA) components of position/velocity/acceleration. Figure 8 shows three heatmaps displaying the rand score for the K-means, GMM, and K-medoids clustering methods, shown from top to bottom; maximum values are displayed in black font. Each unsupervised label was compared with each of the three ground-truth labels, to quantify how well each clustering method and feature matrix could replicate each experimental ground-truth label. Maximum achieved rand score regardless of the feature-type matrix and ground-truth label evaluated clustering method performance; from best to worst performance K-medoids, K-means, and GMM clustering methods had scores of 0.77, 0.75, and 0.7 for the rotational task and 0.82, 0.69, 0.66 for the translational task respectively. The lenient, strict, and complex ground-truth labels were best to worst replicated, using unsupervised clustering for both rotational and translational tasks. The maximum achieved rand scores regardless of the clustering method and feature matrix for lenient, strict, and complex ground-truth labels were 0.77, 0.72, and 0.66 for the rotational task and 0.82, 0.73, 0.74 for the translational task respectively. Regarding the best feature matrices per unsupervised method, K-medoids achieved the highest rand scores using position and velocity features while being compared with the lenient ground-truth label, for both rotational and translational tasks. K-means achieved the highest rand scores using position and all DWT features with respect to the lenient label for the rotational task. However,

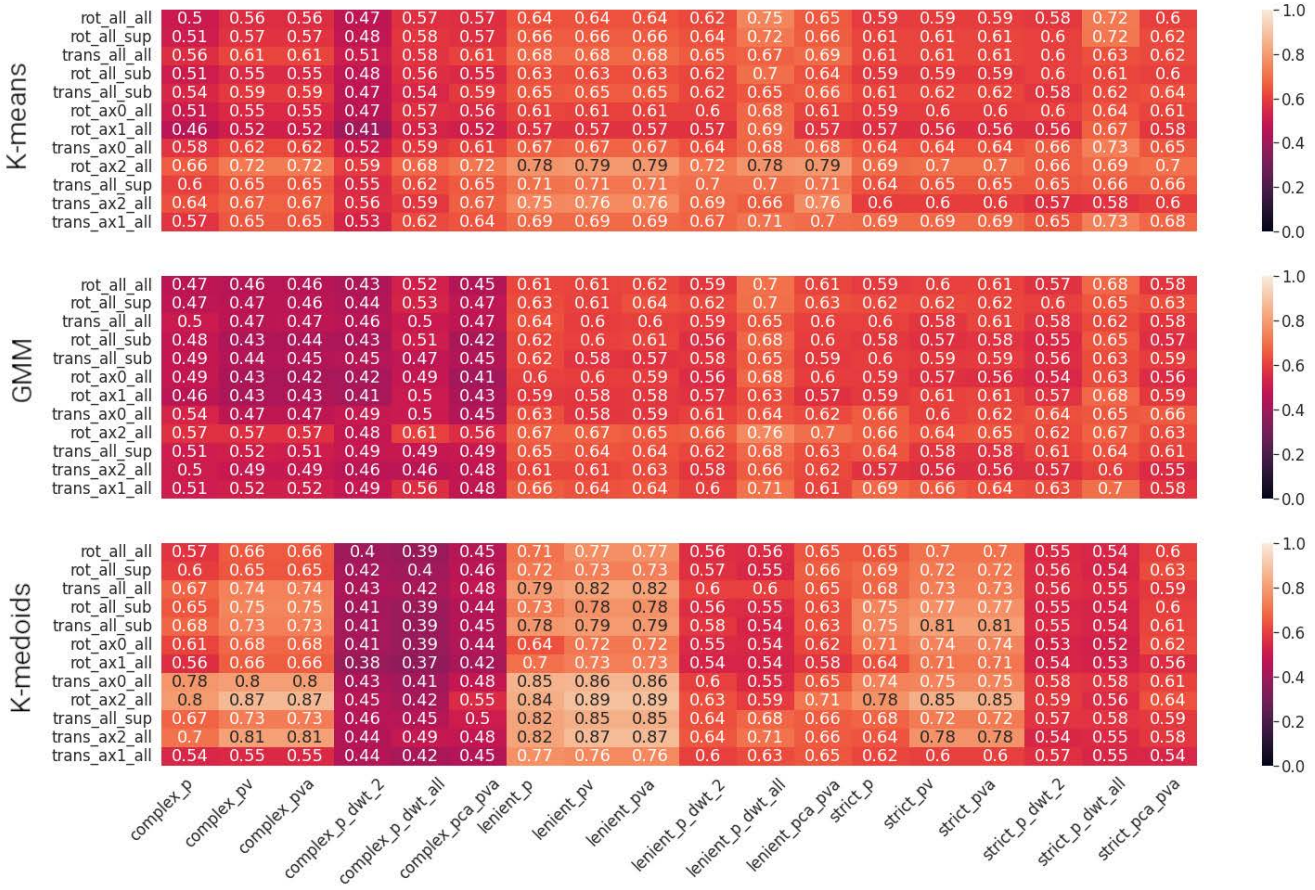


FIGURE 8. **Clustering labels compared with ground-truth labels via the rand score.**

for the translational task, K-means performed best using PCA features of position, velocity, and acceleration while being compared with the lenient ground-truth label. GMM achieved the highest rand scores using position and all DWT features with respect to the lenient ground-truth label for both rotational and translational tasks. These results showed that K-medoids using position and velocity features best replicated the ground-truth labels, specifically K-medoids best replicated the lenient ground-truth label.

V. DISCUSSION

This comprehensive two-part SD study showed that it was possible to isolate, simulate, and recreate realistic aspects of a vestibular feedback dead-reckoning piloting task and create an SD occurrence dataset. Joystick features and detection performance ground-truth labels were constructed from the SD occurrence dataset, supervised classification methods were used to build and identify best predicting models. Unsupervised classification methods identified the best clustering method and joystick features that had the highest rand score with respect to detection performance ground-truth labels. In Part I, rotational and translational motion detection dead-reckoning experiments were used to create an SD dataset, where the whole-body was the tracked object that was perturbed by vestibular and proprioceptive stimulation. The experiment type was selected because whole-body motion detection and dead-reckoning are commonly required flight tasks, and vestibular stimulation is the most basic sensory information in the flight environment. Measuring SD using these basic flight environmental attributes allowed for the results to be generalized to all flight situations and SD use-cases, or in other words these responses described general SD regardless of the flight manoeuvre and SD use-case [4], [6]. Joystick responses, that captured motion detection perception, were measured during vestibular and proprioceptive stimulation with different axial, axial direction, and speed stimuli. Before analysing motion detection response results, a crucial data standardization step was used to verify that the simulator system correctly performed the experimental design; trials with delays and erroneous motion were removed. Motion detection responses were quantified and statistically compared for two speed (sub, sup) and three axis (RO, PI, YA or LR, FB, UD) conditions per experiment type, whereupon four main results were compared with motion detection literature: 1) sup speed stimulation induced more accurate and faster responses than sub speed stimulation, 2) PI, RO, FB, LR, UD, and YA were the least to most difficult axis tasks, 3) axis task ranking confirmed

that there was no sensory advantage for UD detection due to gravity because the vestibular system compensated for gravity, 4) longer reaction times corresponded with task difficulty [29], [30], [31]. The four main results were in alignment with reported literature, thus the SD dataset was approved to capture known human motion detection behavior and the real-time motion simulation environment was considered fidel despite functional timing delays [44]. Approving the SD dataset with respect to literature results was necessary because the entire motion simulation environment including joystick control, speed, and automatic whole-body control were programmed, and thus subject to unexpected functional and unrealistic environmental planning errors. These types of errors can cause the simulation to be administered or saved in an unrealistic or incorrect manner. In addition to the four main results that replicated findings from controlled psychophysical experimentation, an additional result was observed due to the uncontrolled nature of the experiment. Perceptual response to axis motion regardless of speed was observed and quantified using the most counted detection performance response per axis; the most counted response for PI, YA, & UD and RO, LR, & FB axes were IC and EC respectively. The high EC count implied that RO, LR, & FB axes caused less upright posture and/or more self-motion, thus preventing clear interpretation of sensory-cues and resulting in initial mis-detection. Inversely, the high IC count implied that PI, YA, & UD axes encouraged more upright posture and/or less self-motion, allowing for clearer interpretation of sensory-cues and thus resulting in correct initial detection. Posture and self-motion are likely to cause perceptual motion differences because it has been shown that a tilted head position in comparison to an upright head position during roll and left-right translation stimulation resulted in different directional motion perception [26]. Next physical disorientation was evaluated with respect to perceptual motion detection, using the SSQ disorientation sub-scale, because physical health was reported to be one of the causes for SD accidents [1], [4], [6]. One-third of the participants experienced physical disorientation during the task; however, no significant relationship between physical disorientation and motion detection was found. There was a trend where participants who initially detected unsuccessfully felt worse after the experiment than participants who initially detected successfully or did not try. The goal of comparing scalar motion detection values and physical disorientation questionnaire values was to initially determine whether a physical health measure could be a reliable marker for motion detection in the future. Scalar measures showed that there was a relationship between physical disorientation and motion detection. Thus a continuous physiological and/or motion measures that implies physical discomfort, such as EEG, NIRS, heart rate, electrodermal activity, IMU, and human pose camera measurements, are likely to be a reliable AI feature for predicting SD. Additionally we do not claim that questionnaire methods can not quantify SD, however before and after questionnaire samples may not produce enough data to find statistically significant correlations with other SD measures especially when population sample size is small. Questionnaire data to quantify physical sickness or SD use-case symptoms could be exploited using DL recommender systems, where pilots rate a set number of symptoms and/or SD use-cases after each flight and a recommender system model could predict potential SD symptoms or use-cases that the pilot may experience in the future. Finally, Part I experiment and SD dataset scientific contributions are interpreted with respect to classical psychophysical motion detection experimentation and an expert population. Initial detection performance was statistically quantified, regardless of experiment, the best performers achieved 76% detection accuracy and average performers achieved 51% accuracy; no learning or performance improvements over time for any conditions were found. 51-76% task performance success is low with respect to classical psychophysical motion detection experimentation, indicating that the task may have been experimentally too difficult and did not capture clear participant motion detection ability. However, interpreting task difficulty for uncontrolled real-world HAR experiments such as this, with respect to controlled experiments, may not be appropriate. As long as the HAR experiment mimics a real-world phenomenon, the experimental context is scientifically useful despite task difficulty. A real-world piloting task is challenging because many choices need to be made, often times beyond the pilot's attentional capacity, using self-generated and/or non self-generated sensory-cues. The scientific contribution of this experiment and dataset was to measure and understand motion detection response with respect to a piloting task. Our goal was not to measure motion detection response without an on-going task, like is typically used in controlled experiments, where choices and self-motion are limited thus enabling better detection results. Despite the fact that the population was novice, these results are useful for understanding an expert population because the novice results established a baseline for general human motion detection performance. A similar SD dataset using expert pilots will likely have similar axis and speed detection trends, with less overall joystick movement error and variance.

In Part II, supervised and unsupervised classification were used to build reliable SD occurrence predictive models and SD labels from unlabeled joystick data, respectively. A time derivative & spectral analysis feature matrix and three ground-truth labels derived from Part I detection performance categories (e.g. IC, EC, and NC) were created and used in three main model parameter selection studies: evaluation of model architecture and feature usage, ground-truth label comparisons, and unsupervised label comparisons with respect to ground-truth labels. Overall model architecture predictive ability was evaluated using mean test prediction accuracy across all conditions, from best to worst LSTM, RF, Transformer, SVM, MLP, CNN, and LSTM-CNN model architectures had 0.84, 0.82, 0.77, 0.67, 0.58, 0.45, and 0.44 mean accuracy respectively. Model architecture performance results were in alignment with HAR literature.

For example, reports showed that LSTM was more performant than SVM using time-series data, and RF & Transformer architectures were similarly performant as LSTM when they were tuned with appropriate features-types [9]. After identifying globally performant model architectures, five detailed parameter selection analyses were performed to investigate ideal feature matrix and ground-truth label selection for accurate SD prediction. The first detailed parameter analysis evaluated prediction accuracy and ROC-AUC for models that used all of data, called general data usage, in comparison to models that used a subset of the data per speed or axis experimental use-case. Model accuracy and ROC-AUC of general data usage and speed or axis use-cases were statistically compared using two pairwise tests: per ground-truth label and across feature quantity and model architecture, per model architecture and across feature quantity and ground-truth label. Speed and axis use-case comparisons for the first pairwise test showed no significant differences in accuracy & ROC-AUC and no significant differences in ROC-AUC, respectively. However, axis use-case accuracy was significantly higher for the translational lenient label use-case in comparison to general data usage. This result shows that joystick motion was repetitively similar per axis, causing data points per axis to better align with labels. Repetitive misestimation or detection strategies were more likely to occur for the translational task because motion detection was more difficult, similar to the EDA result of Part I where high IC and EC counts for PI, YA, & UD and RO, LR, & FB axis respectively. Additionally, the lenient label exposed this data trend because it had the highest number of samples per class using IC and EC responses as non-SD, such that more samples allowed for better prediction. Finally, speed and axis use-case comparisons for the second pairwise test showed no significant differences in accuracy & ROC-AUC for both use-cases. These two results were useful and important for SD monitoring because currently SD is characterized by use-case and not in a general manner, based on logic association with little numerical confirmation that use-case characterization is more effective. The results showed that for use-cases with repetitive motion, individual use-case models were more advantageous than general data usage. However, general data usage models were sufficiently reliable, therefore SD definitions and/or predictions could be simplified by combining all data use-cases into a single model. The second detailed parameter analysis evaluated PIM and model accuracy prediction for each model architecture across data use-case and ground-truth label, such that the impact of feature quantity could be quantified. PIM showed that Transformer, SVM/MLP, and RF models were likely to require 6, 4, and 2 important features for reliable SD prediction, respectively. Model accuracy prediction comparisons showed that using at least three features, in comparison to using all features resulted in higher prediction. Thus, three features were identified as a sufficient number of features for reliable SD prediction regardless of model architecture; Transformer, LSTM, and RF generated the highest accuracy predictions for both rotational and translational experiments where accuracy was 0.76, 0.74, & 0.66 and 0.93, 0.9, & 0.85 respectively. Knowing that only three features are sufficient for accurate SD prediction contributes to SD monitoring and HAR fields because they provide information about the minimal number of features required to predict SD. SD monitoring and HAR fields have hardware and software constraints, thus requiring small and rapid computational technologies and methods. ML & DL predictions are computationally faster when less features are used, thus these minimal feature quantity results can be directly applied to real-world human activity applications. The third and fourth detailed parameter analysis evaluated the effects of feature-type on model architecture using two types of features; time and/or frequency features; and joystick derivative component features. Regarding time and/or frequency feature usage, feature ordering from most to least important, dictated by PIM, was used to construct a time, frequency, and time & frequency dependency score for each model architecture. The dependency score showed that model architectures used specific feature-types consistently. LSTM, CNN, and LSTM-CNN models used time features. RF, SVM, and MLP used DWT frequency features, and Transformer depended heavily on CWT and short-time FFT time & frequency features. Regarding human movement science motivations to use derivatives of the joystick as features, small non-significant improvements in prediction accuracy were observed when joystick derivatives were used as features in comparison to joystick-only features. Feature-type results with respect to model architecture contribute to both SD monitoring and HAR fields. It is commonplace to test feature-type combinations with different model architectures, typically using a process of trial and error. However, our dependency score constructed from PIM provided insight about which feature attributes were exploitable by model architectures. Time spent on testing features and models could be reduced by having beforehand knowledge about which feature attributes are most appropriate for specific model architectures. Additionally, these results support a feature extraction approach for HAR modeling, by: using a standardized feature matrix containing all feature-type transformations, and ranking their feature and feature-type importance per model architecture. Recent approaches support using model architectures that can predict raw time-series data like LSTM and Transformer, instead of other model architectures like RF or CNN that require transformed time-series data for accurate prediction, because feature extraction can be computationally expensive [24]. However, instead of focusing on model architecture selection for a given feature-type, raw time-series, an alternative approach could focus on the optimal selection of feature-types for model architectures. We demonstrate a standardized approach to identify feature-types for each model architecture, thus allowing for both accurate human activity prediction and novel model generated information about the scientific topic. The fifth detailed parameter analysis demonstrated one way in which ML & DL parameters can be used

to generate information about scientific phenomenon. The LSTM model architecture consistently and more accurately predicted SD than the other model architectures, for both the overview model architecture analysis and four detailed parameter analyses. LSTM prediction per timestep was used to find the ideal amount of joystick data required for accurate SD prediction, and to scientifically identify activity event timing during a piloting task. Hyperparamter tuning found that 20 and 4 second timesteps caused the most accurate LSTM predictions for rotational and translational tasks respectively, the full 40 seconds of trial data was not necessary. Identification of an ideal temporal window for SD prediction contributes to the field of SD monitoring, and scientifically, timestep information of SD or non-SD event activity can facilitate study of human response capacity in both aeronautical and psychophysical fields. The second analysis concerned determination of appropriate SD ground-truth labels. Three ground-truth labels constructed with respect to motion detection and compensation performance, demonstrated that certain characterizations/definitions of SD describe joystick feature data more than others. Lenient, strict, and complex ground-truth labels corresponded to the SD cases where IC & EC responses were considered non-SD, IC responses were considered non-SD, and IC response were non-SD while EC responses were moderate SD, respectively. Prediction accuracy, regardless of other conditions, across model architecture was more accurate for the lenient label in comparison to the other labels. Thus, the lenient label convention best characterized the data, and reaffirmed the existing functional definition, where SD occurrence is defined as 'involving successive failures and major performance errors' [6]. These results contribute to the field of SD monitoring, assisting with defining SD from a numerical perspective. The third and final analysis employed unsupervised classification to quantify the error between ground-truth labels and unsupervised clustering method labels, via the rand score. In addition, ideal features for clustering methods were identified such that the error between ground-truth labels and unsupervised labels were minimum. The lenient ground-truth label received the highest rand-score across all clustering methods, in comparison to strict and complex ground-truth labels. K-medoids using position & velocity or position, velocity, & acceleration features replicated the three ground-truth labels better than other clustering methods and feature combinations. In particular, K-medoids with position & velocity or position, velocity, & acceleration features best replicated the lenient label with rand scores of 0.77 and 0.82 for the rotational and translational task respectively. This unsupervised classification analysis result is significant for the field of SD monitoring and HAR because it provided a quantitative error measure for labeling SD occurrence from joystick data, thus confirming that K-medoids with at least position & velocity features can identify SD with reliable accuracy. Correct labeling of SD is currently unknown and unsupervised methods alone without ground-truth labels can not confirm SD occurrence accuracy. Additionally, we can not reasonably say how similar real-world joystick manipulations are to our experimental dataset, however we demonstrate the usage of clustering methods on two very different joystick trajectories and tasks, rotational and translational. Realistic data is likely to have less variance, than our novice participant dataset, however after scaling, the data is likely to be similar to our experimental dataset because human response movement is limited to a small frequency range. Therefore, these clustering methodologies are strongly likely to apply to expert piloting joystick data.

There were several limiting aspects that could have been improved: measurement of head and body motion using HAR measurements, less real-time experimental control to preserve correct data measurement ordering, class balancing, and image preparation for CNN modeling. Regarding Part I, due to experimental design complexity, head and body motion camera analysis was of less interest. Thus it remained uncertain whether postural differences or following sensory cues caused IC and EC detection count differences for certain axes. In future work, IMU sensors could be attached to the head and body while perception of detected motion in different orientations or directions are measured using a joystick. In this manner, head and body motion can be measured with respect to motion detection perception and physiological reasons for why SD occurred could be concretely identified, allowing for the creation of SD prevention solutions. Next, the reason real-time experimentation was used was to minimize delays in data transfer, however a true implementation of real-time can cause errors if sequentially dependent parts of code are executed in the wrong order. The engineering of the motion simulation experiment could have been controlled better, using less real-time functionality such that required sequential events were executed in a desired order and not in the order of fastest execution. Less trials would have been removed during the data standardization step, if event order was sequentially guaranteed. In addition, data pre-processing during experimental data collection could have been integrated such that only necessary data for analysis was saved. For example, data was saved for the entire experiment however it would have been more efficient if data from the start and end of each trial was only saved. Due to the fact that the entire experiment was saved, an additional preprocessing step was needed to remove the unrequired data. In addition to the mentioned work limitations in Part I, class balancing and image preparation could have been performed differently in Part II. Class balancing using oversampling was performed, such that samples from each minority class were randomly selected to pad the respective class until samples were equivalent to the majority class. Oversampling produces datasets with less natural variability, therefore it would have been better to use techniques that numerically generate samples for the minority class, such as GANs or SMOTE using kmeans [13]. Undersampling was not considered because there were already few class samples due to the rigorous data standardization process. Finally, LSTM-CNN and CNN prediction accuracy was lower than HAR literature reports

because we used matrix representations of data, instead of equivalent figure rendered representations [12]. Specifically, matrix representations of CWT and short-time FFT features were used instead of mathplotlib figure rendered representations; figure representations display more detailed color variations of both CWT and short-time FFT information. This additional shading information that is generated by plotting the matrix may give better CNN accuracy. Initially, CWT and short-time FFT features were plotted using mathplotlib and saved as png images. The image files were opened and used as inputs to the CNN, average results had an accuracy of approximately 0.6-0.7 for roughly 30 data use-cases of the required 144 data use-cases per model architecture. Saving and opening thousands of image files was computationally expensive, therefore the data matrix representation was used instead of the image representations and average accuracy results decreased to approximately 0.45. Despite the lower accuracy values for CNN and LSTM-CNN, we used the matrix representations such that all models could be equally compared using the same numerical data.

VI. CONCLUSION

The two-part SD study demonstrated effective measurement and predictive methods for quantifying SD occurrence. This multi-disciplinary work contributes to many domains of study including aviation, human movement science, motion detection, control theory human-in-the-loop, and HAR. In Part I, human motion detection and compensatory control were measured via a joystick for two speed (sub, sup) and three axis orientation (RO, PI, YA) or direction (LR, FB, UD) stimuli, using a rotational and translational vestibular compensatory whole-body tracking dead-reckoning task. Four motion detection results confirmed that the SD dataset was in alignment with motion detection literature trends, where the results showed that: SD occurred less for faster detectable sup speeds than slower sub speeds, SD occurred least to most for PI, RO, FB, LR, UD, and YA axes, there was no sensory advantage for UD detection due to gravity, reaction times were longer for difficult stimuli. An additional motion detection result, caused by the uncontrolled nature of HAR experimentation, showed that sensory-cues and posture may play an important role for motion detection for specific axes; high IC and EC response count corresponded to PI, YA, & UD and RO, LR, & FB axes respectively. Physical disorientation questionnaire results did not significantly correlate with motion detection performance, however there was a trend that poor and good performers felt worst and similarly, respectively, after the experiment. Physical disorientation and health could be a potential feature for predicting SD, using continuous measurement sensors like EEG, NIRS, HR, EDA, IMU, or a camera. Vestibular dead-reckoning was a difficult task however it was not impossible, because evaluation of global performance showed that best performers could correctly detect motion 76% of the time and average performs could detect motion 51% of the time. The study of SD using vestibular dead-reckoning was not unrealistic from a HAR viewpoint because the goal was to measure motion detection response with respect to the piloting task such that responses contained sufficient samples of SD and non-SD behavior. Additionally, despite the fact that responses were from novice pilots, general axis and speed response trends can be generalized to an expert population, establishing an average motion detection performance baseline. HAR SD experimentation and creation of an SD dataset contributes to the field of SD monitoring and HAR, because SD experimentation has rarely been investigated in a generalised manner measuring continuous task and behavioral measures in a naturalistic setting; SD has been predominantly quantified by use-case using questionnaire reports. Additionally, few HAR studies tackle activity recognition for functional institutionalized tasks such as piloting, the majority of HAR studies investigate simplistic daily-life behaviors such as walking, running, and stair climbing. Using the SD dataset that was constructed in Part I, modeling methods for predicting SD were investigated during Part II. Using HAR ML & DL techniques, model parameter tuning selections for SD prediction were investigated, and ideal tuning parameters are reported and explained. The following key model construction parameters were tested: model architecture, data use-case, feature quantity, feature-type, quantity of data required, ground-truth label, and unsupervised clustering with respect to ground-truth labels. The LSTM model architecture had the highest mean prediction accuracy of 0.84 across experiments, data use-cases, ground-truth labels, feature-types, and feature quantities. The LSTM model required at least three features, and 4 & 20 seconds of data for translational and rotational data respectively, for accurate SD prediction. General data usage was shown to generate reliable SD models, however for repetitive behavioral use-cases, like the translational axis use-cases, modeling with use-case specific data provided more accurate predictions. Permutation importance with the dependency score showed that specific model architectures performed better with time, frequency, or time & frequency feature-type; SVM, RF, and MLP models depended mostly on frequency features; LSTM, CNN, and LSTM-CNN models strongly depended on time features; the Transformer model depended on time & frequency features. The lenient ground-truth label characterized the feature data better than the strict and complex labels; the lenient label definition was in alignment with the current functional SD definition. Unsupervised clustering revealed that K-medoids using at least position and velocity features most accurately replicated all ground-truth labels, with a rand score greater than 0.7, implying that SD occurrence could be reliably predicted from general flight joystick data.

REFERENCES

[1] W. Bles, "Spatial disorientation training demonstration and avoidance," North Atlantic Treaty Organisation (NATO), Soesterberg, The Netherlands, Tech. Rep. TR-HFM-118, 2008.

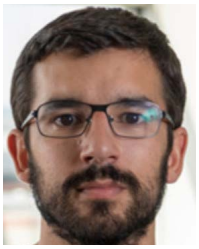
[2] R. Gibb, R. Gray, and L. Scharff, *Aviation Visual Perception: Research, Misperception and Mishaps*. Farnham, U.K.: Ashgate, 2010.

[3] G. Perdriel and A. J. Benson, "Spatial disorientation in flight: Current problems," Advisory Group Aerosp. Res. Develop., Neuilly-sur-Seine, France, Tech. Rep. ADA094913, 1980.

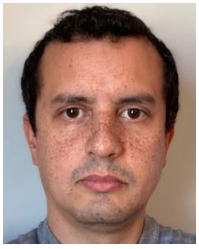
[4] K. K. Gillingham and F. H. Previc, "Spatial orientation in flight," Armstrong Lab. (AFMC), San Antonio, TX, USA, Tech. Rep. AL-TR-1993-0022, 1993.
[5] F. H. Previc and W. R. Ercoline, *Spatial Disorientation in Aviation*. Reston, VA, USA: American Institute of Aeronautics and Astronautics, 2004.
[6] D. G. Newman and A. Faicd, "An overview of spatial disorientation as a factor in aviation accidents and incidents," Austral. Transp. Saf. Bur., Canberra, ACT, Australian, Tech. Rep. B2007/0063, 2007.
[7] C. Hao, X. Fan, C. Dong, L. Qiao, X. Li, X. Li, L. Cheng, L. Guo, and R. Zhao, "A classification method for unrecognized spatial disorientation based on perceptual process," *IEEE Access*, vol. 8, pp. 140654–140660, 2020.
[8] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *Proc. Int. Workshop Ambient Assist. Living*. Cham, Switzerland: Springer, 2012, pp. 216–223.
[9] I. D. Luptáková, M. Kubovčík, and J. Pospíchal, "Wearable sensor-based human activity recognition with transformer model," *Sensors*, vol. 22, no. 5, p. 1911, Mar. 2022.
[10] B. Fu, N. Damer, F. Kirchbuchner, and A. Kuijper, "Sensing technology for human activity recognition: A comprehensive survey," *IEEE Access*, vol. 8, pp. 83791–83820, 2020.
[11] T. V. Marcard, B. Rosenhahn, M. J. Black, and G. Pons-Moll, "Sparse inertial poser: Automatic 3D human pose estimation from sparse IMUs," in *Computer Graphics Forum*, vol. 36. Hoboken, NJ, USA: Wiley, 2017, pp. 349–360.
[12] A. Nedorubova, A. Kadyrova, and A. Khlyupin, "Human activity recognition using continuous wavelet transform and convolutional neural networks," 2021, *arXiv:2106.12666*.
[13] S. An, Y. Tuncel, T. Basaklar, G. K. Krishnakumar, G. Bhat, and U. Y. Ogras, "MGait: Model-based gait analysis using wearable bend and inertial sensors," *ACM Trans. Internet Things*, vol. 3, no. 1, pp. 1–24, Feb. 2022.
[14] F. Xiao, L. Pei, L. Chu, D. Zou, W. Yu, Y. Zhu, and T. Li, "A deep learning method for complex human activity recognition using virtual wearable sensors," in *Proc. Int. Conf. Spatial Data Intell.* Cham, Switzerland: Springer, 2020, pp. 261–270.
[15] B. Cheung, K. Hofer, C. J. Brooks, and P. Gibbs, "Underwater disorientation as induced by two helicopter ditching devices," *Aviation, Space, Environ. Med.*, vol. 71, no. 9, pp. 879–888, 2000.
[16] J. Sargent, S. Dopkins, J. Philbeck, and J. Arthur, "Exploring the process of progressive disorientation," *Acta Psychol.*, vol. 129, no. 2, pp. 234–242, Oct. 2008.
[17] F. Denquin, J. Foucher, S. Pla, J.-C. Sarrazin, and B. G. Bardy, "Optical and gravito-inertial contributions to the perception and control of height in a simulated low-altitude flight context," *Ergonomics*, vol. 64, no. 10, pp. 1297–1309, Oct. 2021.
[18] F. Soyka, P. R. Giordano, K. Beykirch, and H. H. Bülthoff, "Predicting direction detection thresholds for arbitrary translational acceleration profiles in the horizontal plane," *Exp. Brain Res.*, vol. 209, no. 1, pp. 95–107, Mar. 2011.
[19] Y. Wang, J. Tang, V. P. Vimal, J. R. Lackner, P. DiZio, and P. Hong, "Crash prediction using deep learning in a disorienting spaceflight analog balancing task," *Frontiers Physiol.*, vol. 13, p. 51, Jan. 2022.
[20] F. Fusier, V. Valentin, F. Brémond, M. Thonnat, M. Borg, D. Thirde, and J. Ferryman, "Video understanding for complex activity recognition," *Mach. Vis. Appl.*, vol. 18, nos. 3–4, pp. 167–188, 2007.
[21] L. Ding, J. Bo, Q. Wu, H. Liu, and S. Fu, "Multiple scales pilot action pattern recognition during flight task using video surveillance," in *Proc. Int. Conf. Hum.-Comput. Interact.* Cham, Switzerland: Springer, 2015, pp. 601–604.
[22] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal, "Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness," *Int. J. Aviation Psychol.*, vol. 3, no. 3, pp. 203–220, 1993.
[23] S. Bouchard, G. Robillard, and P. Renaud, "Revising the factor structure of the simulator sickness questionnaire," *Annu. Rev. Cyber Therapy Telemedicine*, vol. 5, pp. 117–122, Jan. 2007.
[24] K. Xia, J. Huang, and H. Wang, "LSTM-CNN architecture for human activity recognition," *IEEE Access*, vol. 8, pp. 56855–56866, 2020.
[25] S. E. Chaudhuri, F. Karmali, and D. M. Merfeld, "Whole body motion-detection tasks can yield much lower thresholds than direction-recognition tasks: Implications for the role of vibration," *J. Neurophysiol.*, vol. 110, no. 12, pp. 2764–2772, Dec. 2013.
[26] D. E. Angelaki and K. E. Cullen, "Vestibular system: The many facets of a multimodal sense," *Annu. Rev. Neurosci.*, vol. 31, no. 1, pp. 125–150, Jul. 2008.
[27] G. M. Jones and L. R. Young, "Subjective detection of vertical acceleration: A velocity-dependent response?" *Acta Oto-Laryngol.*, vol. 85, nos. 1–6, pp. 45–53, Jan. 1978.
[28] M. C. B. Rey, T. K. Clark, W. Wang, T. Leeder, Y. Bian, and D. M. Merfeld, "Vestibular perceptual thresholds increase above the age of 40," *Frontiers Neurol.*, vol. 7, p. 162, Oct. 2016.
[29] M. Hartmann, K. Haller, I. Moser, E.-J. Hossner, and F. W. Mast, "Direction detection thresholds of passive self-motion in artistic gymnasts," *Exp. Brain Res.*, vol. 232, no. 4, pp. 1249–1258, Apr. 2014.
[30] F. Karmali, M. C. B. Rey, T. K. Clark, W. Wang, and D. M. Merfeld, "Multivariate analyses of balance test performance, vestibular thresholds, and age," *Frontiers Neurol.*, vol. 8, p. 578, Nov. 2017.
[31] Y. Valko, R. F. Lewis, A. J. Priesol, and D. M. Merfeld, "Vestibular labyrinth contributions to human whole-body motion discrimination," *J. Neurosci.*, vol. 32, no. 39, pp. 13537–13542, Sep. 2012.
[32] R. Shadmehr and S. P. Wise, *The Computational Neurobiology of Reaching and Pointing: A Foundation for Motor Learning*. Cambridge, MA, USA: MIT Press, 2004.
[33] P. Sarang, *Artificial Neural Networks With TensorFlow 2: ANN Architecture Machine Learning Projects*. Cham, Switzerland: Springer, 2021.
[34] V. Mazzia, S. Angarano, F. Salvetti, F. Angelini, and M. Chiaberge, "Action transformer: A self-attention model for short-time pose-based human action recognition," *Pattern Recognit.*, vol. 124, Apr. 2022, Art. no. 108487.
[35] A. S. Radomsky, S. Rachman, D. S. Thordarson, H. K. McIsaac, and B. A. Teachman, "The claustrophobia questionnaire," *J. Anxiety Disorders*, vol. 15, no. 4, pp. 287–297, Jul. 2001.
[36] A. S. Radomsky, A. J. Ouimet, A. R. Ashbaugh, M. R. Paradis, S. L. Lavoie, and K. P. O'Connor, "Psychometric properties of the French and English versions of the claustrophobia questionnaire (CLQ)," *J. Anxiety Disorders*, vol. 20, no. 6, pp. 818–828, Jan. 2006.
[37] J. Landrieu, J. Abdur-Rahim, J.-C. Sarrazin, and B. Bardy, "Time-to-collision estimates during congruent visuo-vestibular stimulations," in *Proc. 19th Int. Conf. Perception Action (IPCA)*. London, U.K.: Psychology Press, 2017, pp. 109–112.
[38] T. Bellmann, J. Heindl, M. Hellerer, R. Kuchar, K. Sharma, and G. Hirzinger, "The DLR robot motion simulator part I: Design and setup," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 4694–4701.
[39] *Python Language Reference, Version 3.9*, Python Software Found., Wilmington, DE, USA, 2021.
[40] A. Burkov, *The Hundred-Page Machine Learning Book*. Quebec City, QC, Canada: Andriy Burkov, 2019.
[41] Y. B. M. A. Ng and K. Katanforoosh, "Deep learning specialization [MOOC]," Coursera, 2021. [Online]. Available: https://www.coursera.org/specializations/deeplearning
[42] G. Louppe, "Understanding random forests: From theory to practice," 2014, *arXiv:1407.7502*.
[43] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Hoboken, NJ, USA: Wiley, 2005.
[44] T. A. Stoffregen, B. G. Bardy, L. J. Smart, and R. J. Pagulayan, "On the nature and evaluation of fidelity in virtual environments," in *Virtual and Adaptive Environments: Applications, Implications, and Human Performance Issues*. Mahwah, NJ, USA: Lawrence Erlbaum Associates, 2003, pp. 111–128.

JAMILAH FOUCHER (Member, IEEE) received the B.S. degree in electrical engineering from the State University of Binghamton, New York, in 2003, and the Ph.D. degree in mechanical engineering from the University of California at Santa Barbara, Santa Barbara, with emphasis in control theory, dynamical systems, and numerical analysis, in 2012, with application to neuroscience and psychophysical experimentation. She is currently a Data Scientist Consultant at Capgemini, a member of Sigma Xi, and a member of the Frontiers Neuroergonomics Editorial Board.

ANNE-CLAIRE COLLET received the Engineering degree, in 2010, a master's degree in biology and neurosciences, in 2012, and the Ph.D. degree in cognitive neurosciences, in 2016, with an emphasis on perception and object recognition. She is currently working with the Human Design Group as a Neuroscientist and is involved as a Subcontractor in Airbus Projects that require expertise in neuroscience and/or in human physiology.



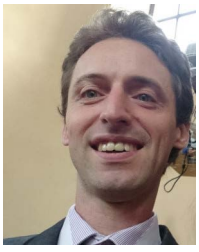
KEVIN LE GOFF received the Ph.D. degree in neuroscience from Aix-Marseille University, in 2016. He is currently a Human Factors Specialist working on cockpit design and physiological topics at Airbus Operations SAS, Toulouse.



THOMAS RAKOTOMAMONJY received the Diploma (Engineering) degree in aerospace from the l'Ecole Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse, in 2002, and the Ph.D. degree in automatics and systems control from Aix-Marseille University, in 2006. He is currently a Senior Research Scientist at ONERA—The French Aerospace Laboratory, working on model identification for piloting assistance and interfaces.

VALÉRIE JUPPET is currently an Airbus Expert at Airbus Helicopters Design Office, in the domain of human factor applied to the cockpit design and a Certification Verification Engineer for cockpit arrangement and human factor topics.

THOMAS DESCATOIRE received the master's degree in mechanical engineering. Currently, he is working as a Cockpit Operations Specialist in the cockpit design and the Human Factor Department, Airbus Helicopters Design Office.



JERÉMIE LANDRIEU received the M.S. degree in mechanical engineering and the Ph.D. degree in computer graphics with emphasis in virtual reality technologies. He is currently the Research and Development Manager of Gambi-M Company.

MARIELLE PLAT-ROBAIN received the Ph.D. degree in cognitive psychology from the University of Paris VIII, with emphasis on ergonomy, in 2001. Her Ph.D. thesis was about aviation atypical incidental situations (linked with automations surprises) and crew diagnosis of those situations. She is currently working on cockpit design at Airbus Operations SAS, Toulouse, for 21 years.

FRANÇOIS DENQUIN received the Diploma (Engineering) in electronics and informatics from the ENSEA Engineering School, in 2015, the M.S. degree in artificial intelligence and robotics from the University of Cergy-Pontoise, France, in 2015, and the Ph.D. degree in human movement sciences from the University of Montpellier, France, in 2021, with emphasis in neurosciences and human perception of the environment. He is currently an IA Engineer with the Research and Development Team, AnotherBrain Company.

ARTHUR J. GRUNWALD is currently a (retired) Associate Professor at the Faculty of Aerospace Engineering, Technion, Haifa, Israel. His research interests include flight simulation, human–machine systems, advanced electronic display formats in aerospace applications, visual perception models, and human interface design.

JEAN-CHRISTOPHE SARRAZIN received the Ph.D. degree. He is currently a Cognitive Psychologist. After two postdoctoral research in computational neuroscience (Postdoctoral Researcher at INRIA for one year) and cognitive neuroscience (Postdoctoral Researcher at Marie Curie for two years), respectively, he received a permanent position at the Office National d'Etudes et de Recherches Aérospatiales (ONERA, the French Aerospace Laboratory). At ONERA, he is with the Systems and Information Processing Department, he leads the research unit ''Cognitive Engineering and Applied Neurosciences'' and works on the scientific and technical development of human system integration studies.

BENOÎT G. BARDY received the Ph.D. and Habilitation degrees in movement sciences from Aix-Marseille University, France, in 1991 and 1998, respectively. In 1994, he was received a NATO Research Fellowship to develop his research on locomotion in virtual reality at Brown University, RI, USA. After being selected as a new Professor at the University of Paris (Paris-Sud), in 1999, he entered the Prestigious Institut Universitaire de France as a Junior Member, from 2001 to 2006. In 2005, he returned to the South of France and created EuroMov, a European Center for research, technology, and innovation at the crossover between movement, health, and digital sciences [now EuroMov Digital Health in Motion (DHM), (www.euromov.eu)]. In 2012, he was re-inducted into the Institut Universitaire de France as a Senior Member. His research interests include perception and action, in real and virtual environments, deploying technology-oriented, rehabilitation, industrial, or artistic solutions. He has coordinated several European, national, and regional initiatives, in both research and research and development projects, and he is a Regular Expert for the European Commission and the private sector.