

**Autonomous Socially Assistive Drones Performing Personalized Dance Movement Therapy
An Adaptive Fuzzy-Logic-Based Control Approach for Interaction with Humans**

Ascensao, Tomas; Jamshidnejad, Anahita

DOI

[10.1109/ACCESS.2022.3143992](https://doi.org/10.1109/ACCESS.2022.3143992)

Publication date

2022

Document Version

Final published version

Published in

IEEE Access

Citation (APA)

Ascensao, T., & Jamshidnejad, A. (2022). Autonomous Socially Assistive Drones Performing Personalized Dance Movement Therapy: An Adaptive Fuzzy-Logic-Based Control Approach for Interaction with Humans. *IEEE Access*, *10*, 15746-15770. <https://doi.org/10.1109/ACCESS.2022.3143992>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Received November 3, 2021, accepted January 1, 2022, date of publication January 18, 2022, date of current version February 11, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3143992

Autonomous Socially Assistive Drones Performing Personalized Dance Movement Therapy: An Adaptive Fuzzy-Logic-Based Control Approach for Interaction With Humans

TOMÁS ASCENÇÃO AND ANAHITA JAMSHIDNEJAD 

Department of Control and Operations, Faculty of Aerospace Engineering, Delft University of Technology, 2629 Delft, The Netherlands

Corresponding author: Anahita Jamshidnejad (a.jamshidnejad@tudelft.nl)

This work was supported by the Netherlands Organisation for Scientific Research (NWO)—Open Mind Project “Drones Interacting With Children With Autism via Dance Movement Therapy,” under Grant 18071.

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Human Research Ethics Committee of TU Delft under Approval No. 1990.

ABSTRACT Novel personalized and affordable approaches are needed in order to provide efficient therapeutic interventions for people with autism spectrum disorder. In this paper, we introduce a new category of socially assistive robots, i.e., socially assistive drones (SADs) for therapeutic interactions with humans. SADs autonomously perform and engage humans in dance movement therapy (DMT), which has proven to be highly effective for people with autism spectrum disorder when it is personalized and adapted carefully. The main focus of this paper is on developing adaptive, personalized, and interactive control approaches based on fuzzy logic, which efficiently deals with nonlinear dynamics and directly introduces the expert (linguistic) knowledge into the control system. The developed approaches are implemented via a small quadcopter in real-life interactions with 10 participants. The results of the case study prove the excellent performance of the SAD in adapting and personalizing its decisions with respect to each user and thus keeping them engaged in the proposed DMT plans.

INDEX TERMS Adaptivity and personalization, autism spectrum disorder, dance movement therapy, fuzzy logic, socially assistive robots.

I. INTRODUCTION

Autism Spectrum Disorder (ASD) refers to complex neurodevelopmental impairments that result in difficulties in social interactions and verbal and non-verbal communication. The prevalence of ASD is around 2.7% in boys and 0.67% in girls. Almost all people with ASD exhibit some level of language impairment, with a large percentage being non-verbal [1]. Moreover, 80%-90% of people with ASD are affected by an impaired development of motor skills¹ [2]. Studies show that the severity of motor skill impairments in ASD corresponds to the severity of social and commu-

nication impairments (see, e.g., [3]–[7]). Although currently ASD cannot be cured, people with ASD can improve their quality of life via early, long-term therapeutic interventions. For instance, personalized Dance Movement Therapy (DMT) has proven to significantly improve the motor skills of people with ASD, especially those who are non-verbal and may not benefit from other therapeutic interventions [8], [9]. DMT includes mirroring (i.e., matching, reflecting, or echoing) the movements of the therapist by the client (for improving empathetic expressions) and encouraging the client to initiate more spontaneous movements that should be mirrored by the therapist (for improving the connection with and exploration of the environment).

The main challenges in providing DMT for people with ASD include (1) the need for personalizing the DMT plans based on the stage of ASD and specific needs of every

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

¹Motor skills include abilities learned by humans that can result in pre-terminated movement outcomes with maximum certainties.

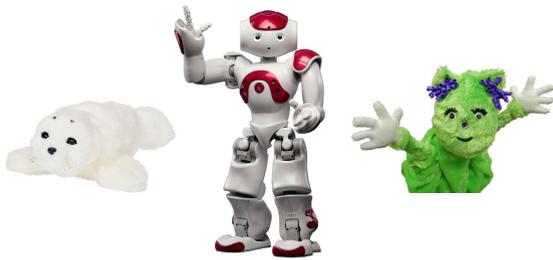


FIGURE 1. Different SARs used for ASD therapy in literature (from left to right: Paro [10], Nao [11], and Charlie [12]).

person, (2) making the intervention available, with any desired frequency, and engaging in long terms, and (3) keeping the corresponding costs low, so that DMT is affordable for more people. Therefore, it is essential to find alternative approaches for providing affordable personalized DMT.

Socially Assistive Robots (SARs) provide aid or support for humans (e.g., in rehabilitation, education, mobility, etc.), while maintaining social interactions with them. It has been shown that therapeutic interventions for ASD via SARs can significantly improve the impact and therapeutic outcome of these interventions (see, e.g., [13]–[18]). Correspondingly, various SARs have been developed or adapted for use in therapy sessions for ASD (see Figure 1). In order to achieve desired therapeutic performance and outcome via SARs, these robots should possess the following key characteristics:

- 1) **Desirable appearance:** Although humanoid robots are highly accepted by humans, they may hinder the effectiveness of therapeutic interactions for people with ASD, since they tend to experience discomfort around humans (or robots that resemble humans) and thus avoid interacting with them [1]. In general, the less complex and expressive the appearance of a robot, the more desirable it is for therapeutic applications for ASD.
- 2) **Autonomy:** Autonomy in decision making is required for SARs in order to continue performing independently without constant supervision of therapists or caregivers. This also allows therapists or caregivers to focus on therapeutic aspects that are not necessarily addressed by the SAR or to participate in several parallel therapy sessions.
- 3) **Adaptivity:** For effective interactions, SARs should adapt their behaviour and decisions according to the environmental and behavioral changes of humans. Adaptivity of SARs positively affects their acceptance by humans, and encourages humans to interact with them in longer terms [19].
- 4) **User-friendliness:** Therapists, clinicians, and caregivers should be able to operate or re-program SARs easily and without need for in-depth technical knowledge. More specifically, the user interface of a SAR should be simple and intuitive in providing the desired commands and for entering or accessing data of therapy sessions.

- 5) **Responsiveness:** Since people with ASD may exhibit unpredictable patterns of behavior and interaction or may withdraw from the interactions [20], SARs should be responsive to the immediate changes a therapist or caregiver provides during a therapy session. Note that while adaptivity is related to the autonomous performance of SARs, responsiveness is related to the robot's performance with regards to the external control inputs it receives.

Considering these five characteristics, drones are highly suitable candidates for assisting in therapeutic interventions for ASD. However, drones have never been used in systematic therapeutic interventions for ASD, or generally as SARs. High mobility, manoeuvrability, three degrees of freedom suited for performing DMT, and their simple, appealing appearance make drones promising candidates for ASD therapy.² Therefore, in this paper we introduce a new concept, socially assistive drones or briefly SADs, and we develop the first SAD used for performing personalized DMT in live interactions with humans. Autonomy, adaptivity, user-friendliness, and responsiveness of SADs should mainly be provided by the approaches that steer SADs. More specifically, adaptivity and autonomy can be provided by proper development of the SAD's control system, while user-friendliness and responsiveness are linked to both the control system and the user interface. The main focus of this paper is on the development of the control system of a SAD, where adaptive fuzzy-logic-based approaches are considered.

Fuzzy-logic-based (FL) controllers were first introduced by Mamdani [23] (based on the concept of fuzzy sets [24]) and were later on extended by Takagi and Sugeno [25]. FL controllers make decisions using a fuzzy inference system and according to a rule base that consists of rules formulated as if-then statements that include linguistic terms. The input and output of an FL controller are fuzzy values. Therefore, a fuzzifier and a defuzzifier are used to transform the crisp values into fuzzy ones and vice versa, since sensors and actuators of the controlled system usually perform according to crisp values. FL controllers are usually model-free, which makes them suitable when no model of the controlled system is available. Moreover, FL controllers perform based on approaches that are very close to decision making of humans. When building up the rule base of an FL controller, human expert knowledge (expressed via linguistic terms) can directly be incorporated within the controller. In Takagi-Sugeno-Kang (TSK) FL controllers [25], although the antecedent of the rules include fuzzy sets, their consequent directly produces crisp values based on an affine combination of the inputs. In this paper, TSK-based FL controllers are used for developing the controllers of the SAD, since the TSK approach is computationally efficient, is easy to be tuned, and can easily be made adaptive.

²The drone summer camps for children with ASD have shown that drones are very appealing to these children (see, e.g., [21], [22])

II. MAIN CONTRIBUTIONS & STRUCTURE OF THE PAPER

In this paper, we introduce a socially assistive drone (SAD) that autonomously designs and performs personalized DMT plans in live interactions with humans. In particular, the main contributions include:

- 1) developing an image-processing-based data analysis module with a fault-tolerant algorithm for online processing of the human’s motions
- 2) developing a generalized adaptive fuzzy-logic-based control system that autonomously steers the SAD to personalize its decisions
- 3) implementing the proposed approaches to a Parrot Bebop 2 quadcopter

Analysis and assessment of the performance of the resulting SAD via real-life experiments with volunteer human participants, show that the developed SAD possesses autonomy, adaptivity, user-friendliness, and responsiveness, which together with its simple, appealing appearance make it suited for DMT for ASD.

The rest of the paper is organized as it follows: Section III details the proposed approaches for real-time human-SAD interactions via DMT. In particular, the proposed approaches developed for data analysis and adaptive, personalized control of the SAD are discussed. Section IV explains the case study with the developed SAD. In Section V the results of the experiments are represented and discussed. Finally, Section VI concludes the paper and gives topics for future research.

III. PROPOSED RESEARCH METHODOLOGIES

In this section, we explain data analysis and control approaches developed for the SAD, which together build up the decision making module that steers the SAD to autonomously perform personalized DMT plans in live interactive sessions with humans.

A. HUMAN-SAD INTERACTIONS

In order to autonomously perform systematic DMT plans and to maintain the interactions with humans in long terms, the SAD needs to accomplish the following tasks successfully:

- 1) gathering and processing relevant data from humans and generating reliable, comprehensive information
- 2) analyzing the generated information according to the purposes and criteria of the interactions
- 3) injecting the results of the assessments into the SAD’s control system, which steers the SAD according to the aim of interactions

Figure 2 illustrates the three steps indicated above. In the next sections, we provide details on these steps.

B. DATA CAPTURING VIA SAD’s CAMERA

The proposed SAD is a quadcopter equipped with a camera that records live video footage during DMT sessions. We assume that at most one human at a time is present in the frame of view of the SAD. A fixed time interval, called the *wait time*, is considered, during which the SAD remains

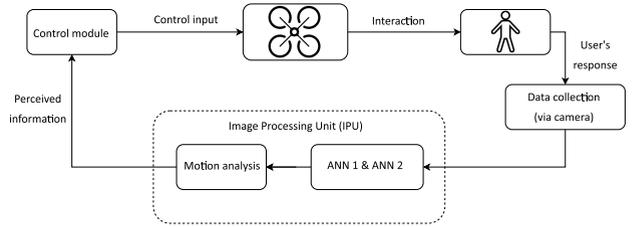


FIGURE 2. Schematic view of the human-SAD interaction procedures.

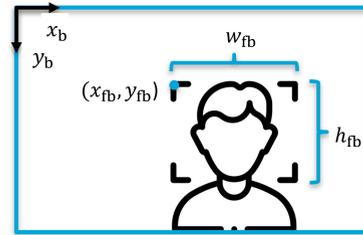


FIGURE 3. Output of ANN 1: Values in pixels corresponding to the user’s face-box, including the coordinates of the top left corner (x_{fb}, y_{fb}) , width w_{fb} , and height h_{fb} .

in the same position and records all the relevant positions of the user (details are given in Sections III-B1 and III-B2). In order to identify the presence of a user within the recorded video footage and to analyze and evaluate the movements of the user, two artificial neural networks (ANNs) and a motion processing algorithm are developed, which together form the image processing unit (IPU) that should process the captured images in real time. Next we discuss the details of the two ANNs in the data analysis module.

1) ANN 1: FACE DETECTION

The first ANN, called ANN 1, allows the SAD to identify the face of a human within the captured images by sketching a rectangle (called the face-box) that delimits the user’s face (see Figure 3). The output of ANN 1 includes the following values in pixels: (1) coordinates (x_{fb}, y_{fb}) of the upper left corner of the face-box, (2) the width, w_{fb} , of the face-box, and (3) the height, h_{fb} , of the face-box. In Figure 3 the largest rectangle shows the SAD’s frame of view and x_b and y_b refer to the SAD’s body frame (i.e., axes attached to the drone’s body).

2) ANN 2: BODY/JOINT DETECTION

The second ANN, called ANN 2 (developed by Openpose Python [26], [27]), is responsible for identifying the joint positions of a detected user. Overall, a total of 18 joint positions (see Figure 4) can be identified, where the output of ANN 2 includes the coordinates of these joint in pixels.

Remark 1: In addition to the coordinates and values generated by ANN 1 and ANN 2, the observation time associated with each of these values is stored on a remote computer. This information will later on be used by the SAD to estimate the speed of the user’s movements, for both assessment of

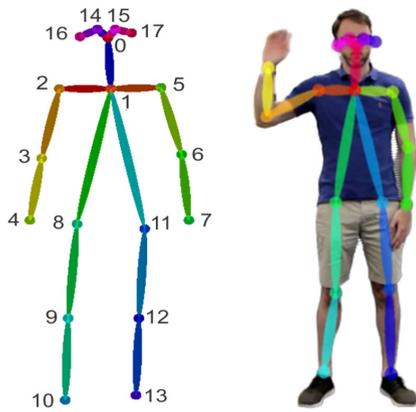


FIGURE 4. Output of ANN 2: Coordinates in pixels of the 18 joints of the user’s body.

the interactions and determining the speed and frequency of various DMT plans that will be proposed by the SAD.

C. DATA ANALYSIS MODULE

The outputs of the ANNs are injected as input into the motion processing algorithm of the SAD, which uses these inputs to (1) estimate the current position of the user, (2) evaluate the level of engagement and performance of the user according to the DMT plan, and (3) capture the movements of the user that the SAD is expected to mirror. Next, we explain the motion processing algorithm in detail.

1) MOTION PROCESSING ALGORITHM

The information regarding the i^{th} measurement captured by the SAD is skimmed to contain: (1) capture time t_i in s, (2) horizontal coordinate x_{t_i} of the relevant data in pixels, and (3) vertical coordinate y_{t_i} of the relevant data in pixels. For the sake of simplicity of the notations we use $r_{t_i} = (x_{t_i}, y_{t_i})$. The identified motions are registered in five categories: motion to the left, to the right, upwards, downwards, and pause, considering every two consecutive coordinates r_{t_i} and $r_{t_{i+1}}$:

- 1) In case the coordinates (almost) overlap, i.e., $\|r_{t_i} - r_{t_{i+1}}\| < \tau_m$, with τ_m a motion threshold, then the corresponding motion belongs to the category “pause”.
- 2) In case the distance between the two consecutive coordinates is larger than or equal to the motion threshold, i.e., $\|r_{t_i} - r_{t_{i+1}}\| \geq \tau_m$, a motion is detected. The motion is horizontal when $|y_{t_i} - y_{t_{i+1}}| < \gamma|x_{t_i} - x_{t_{i+1}}|$, and is vertical when $|x_{t_i} - x_{t_{i+1}}| < \gamma|y_{t_i} - y_{t_{i+1}}|$, with $0 < \gamma < 1$ a ratio that can be identified per person.³
- 3) To specify the heading of the motion the sign of $x_{t_i} - x_{t_{i+1}}$ and $y_{t_i} - y_{t_{i+1}}$ are considered for, respectively, horizontal and vertical motions. A positive sign indicates

³In reality, it is very unlikely for a user to move consistently in exactly either a horizontal or a vertical direction. For instance, a user who aims to move horizontally may still make some small vertical movements inadvertently. The ratio γ has been considered in analysis of the user’s motions to determine whether or not these secondary motions should be considered by the SAD.

a motion to the right or downwards, while a negative sign indicates a motion to the left or upwards.

Finally, based on the coordinates and registered time instants, a speed (in pixels/s) is associated to every identified motion category.

Fault tolerance for motion processing algorithm:

The motion processing algorithm should possess an acceptable level of fault tolerance, in order to cope with (1) *missing data*, i.e., when the user and/or the relevant joints are not identified by the IPU; (2) *erroneous data*, i.e., when the position of the user and/or the corresponding joints have been captured, but involve (non-negligible) errors. A missing coordinate may occur due to an IPU failure or because the user leaves the SAD’s frame of view. In case the number of consecutive missing coordinates is small, it is unlikely that the user has left and returned to the SAD’s frame of view. Thus it is assumed that an IPU failure has occurred. Otherwise, it is assumed that the user has left the SAD’s frame of view.

Missing coordinates due to IPU failure: The first time instant that corresponds to a missing coordinate is called the time of failure and is denoted by t_f . The most recent and the first next time instants with respect to t_f when a reliable measurement has been captured are specified by t_f^b and t_f^a , respectively. The coordinates corresponding to time instants $r_{t_f^b}$ and $r_{t_f^a}$ are represented by $r_{t_f^b}$ and $r_{t_f^a}$. In case the coordinates $r_{t_f^b}$ and $r_{t_f^a}$ imply the same category of motion as the most recent time interval $[t_f^b - \Delta t, t_f^b]$ (with Δt the fixed time step for capturing a measurement by the IPU), and the speed for moving from coordinate $r_{t_f^b}$ to coordinate $r_{t_f^a}$ within time interval $[t_f^b, t_f^a]$ compared to the speed of the captured movement within time interval $[t_f^b - \Delta t, t_f^b]$ indicates a uniform motion (see Figure 5), then the motion processing algorithm ignores the missing intermediate points (as they are assumed to be positioned uniformly between $r_{t_f^b}$ and $r_{t_f^a}$).

If based on the coordinates $r_{t_f^b - \Delta t}$ and $r_{t_f^b}$, and $r_{t_f^b}$ and $r_{t_f^a}$ the direction of the motion has remained the same within time intervals $[t_f^b - \Delta t, t_f^b]$ and $[t_f^b, t_f^a]$, while the speed of the movement has changed (see Figure 6), then n intermediate coordinates, with $n = (t_f^a - t_f^b)/\Delta t - 1$, between the coordinates $r_{t_f^b}$ and $r_{t_f^a}$ are specified by the motion processing algorithm, such that the magnitude of the speed corresponding to these intermediate time intervals evolves according to a fixed rate, and the SAD’s position for time step t_f^a reaches the position captured for this time step by the IPU. More specifically, the rate of changes of the speed is given by:

$$\Delta v = \frac{2(|r_{t_f^a} - r_{t_f^b}| - (n + 1)|r_{t_f^b} - r_{t_f^b - \Delta t}|)}{(n + 1)(n + 2)\Delta t} \quad (1)$$

When according to the motion processing algorithm, the direction of the motion has changed between t_f^b and t_f^a (see Figure 7), the intersection of the piece of line that connects the points corresponding to $r_{t_f^b - \Delta t}$ and $r_{t_f^b}$ with the piece of line that connects the points corresponding to $r_{t_f^a}$ and $r_{t_f^a + \Delta t}$ will be considered as the point where the direction of the motion

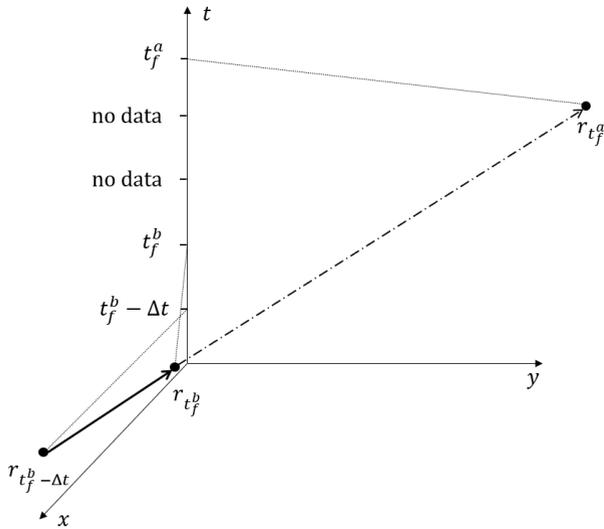


FIGURE 5. Missing data: Uniform motion during the time interval when data has not been captured.

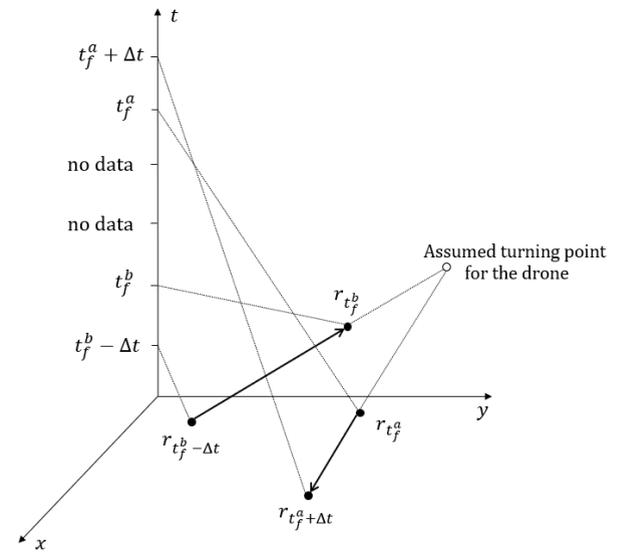


FIGURE 7. Missing data: Direction of motion has changed during the time interval when data has not been captured.

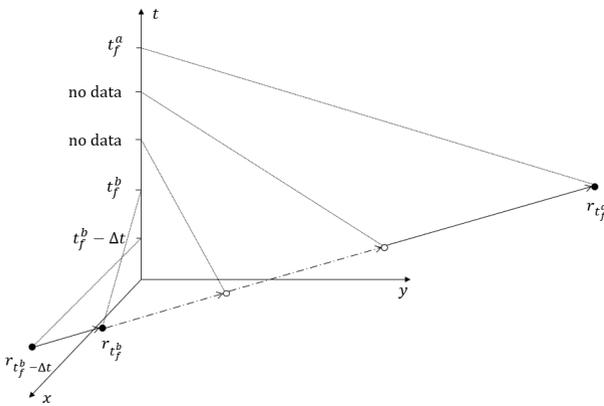


FIGURE 6. Missing data: Motion in the same direction as before with a varying speed during the time interval when data has not been captured.

has changed. In case based on the most recent and most posterior time intervals with respect to the time interval $[t_f^b, t_f^a]$, the speed of the motion has also changed the path determined by the motion processing algorithm from time instant t_f^b to time instant t_f^a will be divided into n intermediate coordinates, such that the magnitude of the speed corresponding to these intermediate time intervals evolves according to a fixed rate (see (1)).

Missing coordinates due to the user leaving the SAD's frame of view: When it is assumed that the user has left the SAD's frame of view, the last registered coordinates (x_{user}^L, y_{user}^L) of the user is considered and the distances d_i^L of this position with respect to the eight border points b_i of the SAD's frame of view with $i \in \{1, \dots, 8\}$ are computed (see Figures 8 and 9).

In case there is no later coordinates registered for the user (see Figure 8), the algorithm fills in the missing coordinates by the coordinates of the border point that is the closest to the

point (x_{user}^L, y_{user}^L) . However, when there is a new registered position (x_{user}^R, y_{user}^R) for the user (see Figure 9), the motion processing algorithm considers the distances d_i^R of this position with respect to the eight border points as well as the distances $d_j^{O,R}$ and $d_k^{O,L}$ between the positions corresponding to $(x_{user}^{O,j,k}, y_{user}^{O,j,k})$ are computed, where $j, k \in \{1, \dots, 8\}$. Note that $(x_{user}^{O,j,k}, y_{user}^{O,j,k})$ corresponds to the intersection point of the lines that pass through the point (x_{user}^L, y_{user}^L) and the border point b_j , and point (x_{user}^R, y_{user}^R) and the border point b_k . Finally, the coordinates of the two border points b_j and b_k , for which $d_j^L + d_j^{O,L} + d_k^R + d_k^{O,R}$ is the smallest are used to fill in the missing registrations (e.g., in Figure 9 $b_j = b_2$ and $b_k = b_4$).

Erroneous data: Erroneous data is identified by abnormal deviations from a specific motion pattern. Mathematically speaking, an error corresponds to a registered position \tilde{r}_t that satisfies both of the following conditions:

$$\min \left\{ \|\tilde{r}_t - r_{t-\Delta t}\|, \|\tilde{r}_t - r_{t+\Delta t}\| \right\} > \tau_m \quad (2)$$

$$\max \left\{ \|\tilde{r}_t - r_{t-\Delta t}\|, \|\tilde{r}_t - r_{t+\Delta t}\| \right\} > \tilde{C} \|r_{t+\Delta t} - r_{t-\Delta t}\| \quad (3)$$

Where $\tilde{C} > 1$ is called the anomaly coefficient. First, a motion across the three consecutive registered positions $r_{t-\Delta t}$, \tilde{r}_t , and $r_{t+\Delta t}$ should be detected (see condition (2)). Condition (3) defines a limit on the deviation from a given motion pattern for every registered position. Erroneous data at time instant t is identified whenever the distance between the associated registered position \tilde{r}_t , and either of the previous, i.e., $r_{t-\Delta t}$, or following, i.e., $r_{t+\Delta t}$, registered positions is too large.

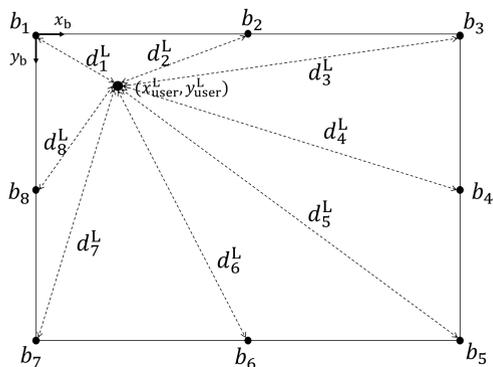


FIGURE 8. Missing data due to the user leaving the SAD’s frame of view: A case where no new position has been registered for the user later on.

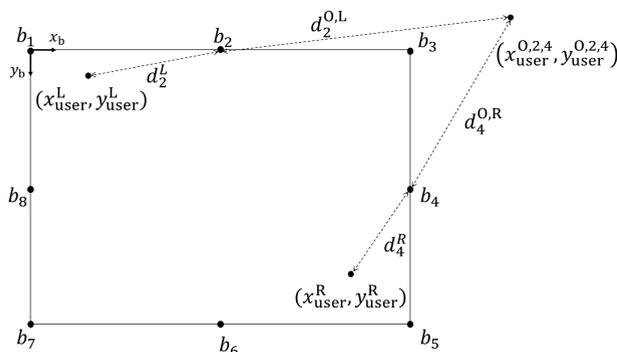


FIGURE 9. Missing data due to the user leaving the SAD’s frame of view: A case where the user has left and later on returned to the SAD’s frame of view.

Figure 10 shows two examples with absence and presence of erroneous data. In this figure, satisfaction of condition (2) means that d_{23} (which is assumed to be smaller than d_{12}) is larger than τ_m , meaning that r_{t_2} falls outside of the illustrated circular areas of radius τ_m . The left-hand side plot shows a case where condition (3) (supposing that $\tilde{C} = 2$) does not hold for the registered position r_{t_2} , i.e., d_{12} (which is supposed to be larger than d_{23}) is not larger than $2d_{13}$. Therefore, there is no erroneous data detected for time instant t_2 . However, in the right-hand side plot, d_{12} is larger than $2d_{13}$, which according to (3) implies erroneous data for r_{t_2} .

Whenever erroneous data is identified, the corresponding registered position should be corrected. Suppose that we have either of the following cases:

$$|x_{t+\Delta t} - x_{t-\Delta t}| < 2\tau_m \tag{4}$$

$$|y_{t+\Delta t} - y_{t-\Delta t}| < 2\tau_m \tag{5}$$

Then the registered position that is prone to error is ignored and the SAD considers $r_{t-\Delta t}$ and $r_{t+\Delta t}$ as two consecutive registered positions. In case (4) does not hold, while the following condition holds (see the sub-area in between the dashed vertical lines in the right-hand side plot in Figure 10):

$$\begin{aligned} \min \{x_{t-\Delta t}, x_{t+\Delta t}\} + \tau_m < \tilde{x}_t < \\ \max \{x_{t-\Delta t}, x_{t+\Delta t}\} - \tau_m, \end{aligned} \tag{6}$$

only the y-coordinate of \tilde{r}_t should be corrected, i.e.:

$$x_t = \tilde{x}_t \text{ and } y_t = 0.5(y_{t-\Delta t} + y_{t+\Delta t}) \tag{7}$$

If (5) does not hold, while the following condition holds (see the sub-area in between the dashed horizontal lines in the right-hand side plot in Figure 10):

$$\begin{aligned} \min \{y_{t-\Delta t}, y_{t+\Delta t}\} + \tau_m < \tilde{y}_t < \\ \max \{y_{t-\Delta t}, y_{t+\Delta t}\} - \tau_m, \end{aligned} \tag{8}$$

only the x-coordinate of \tilde{r}_t is corrected. We have:

$$x_t = 0.5(x_{t-\Delta t} + x_{t+\Delta t}) \text{ and } y_t = \tilde{y}_t \tag{9}$$

D. SAD’s CONTROL SYSTEM: DMT GAME MODES

The main goal of the SAD is to sustain effective interactions according to systematic DMT plans with a user in long terms. Therefore we have considered a number of therapeutic scenarios, called “game modes”. The proposed game modes fall within one of the following two categories:

- 1) *Passive game modes*: These game modes mainly aim at promoting empathetic illustrations in the user. Hence, the SAD mirrors the movements of the user and takes a passive role in the DMT interactions.
- 2) *Active game modes*: These game modes mainly aim at enhancing the connection of the user with their environment, thus the user is expected to follow the movements that are initiated by the SAD, which takes an active role in the DMT interactions.

In passive game modes, during a given wait time the SAD captures the motions of the user, which it mirrors afterwards. In active game modes, the SAD first performs a specific DMT movement and then captures and assesses the reactions or responses of the user within the given wait time.

Overall, four game modes are proposed, which will be discussed in the next sections. A modular control system is developed for the SAD, where this control system includes one controller per game mode. Moreover, a *standby controller* is included in order to adjust the initial states of the SAD at the beginning of every game mode and to regulate these states in transitions between every two game modes.

Remark 2: In order to control the SAD, in our case studies (see Section IV) the PyParrot [28] interface for Python is used. With this library, in order to control the speed of the drone (i.e., the displacement and the time the drone needs to implement the displacement), five control variables are considered: duration of the displacement (in [s]), vertical speed of the drone (in m/s), roll, pitch, and yaw displacements of the drone (in deg), all given as a percentage ranging in $[-100, 100]$ of their maximum allowed values, with the signs determining the direction of the corresponding displacement (see Figure 11 for the definition of the roll, pitch, and yaw angles, noting that the axes have been defined such that at the initial state (i.e., at $\theta = 0, \psi = 0, \phi = 0$) they are aligned with x_b and y_b axes defined earlier for the SAD’s frame of

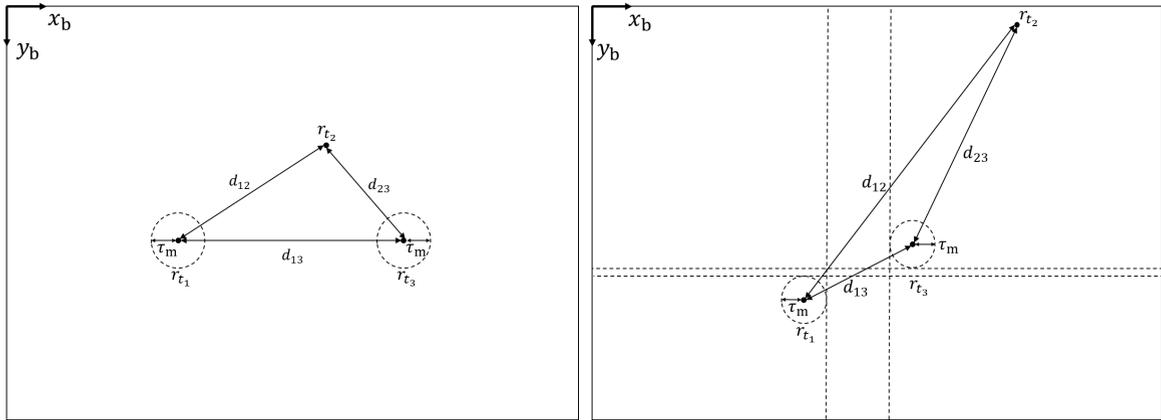


FIGURE 10. No erroneous data for r_{t_2} (left-hand side picture). Erroneous data detected for r_{t_2} (right-hand side picture).

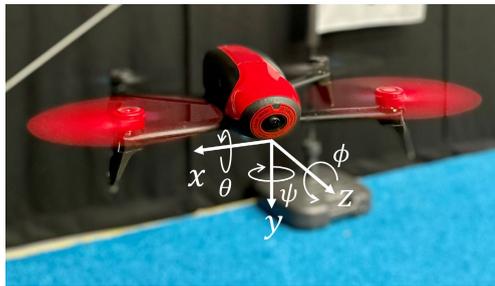


FIGURE 11. Reference inertial frame for the SAD with the pitch θ , yaw ψ , and roll ϕ angles being defined as the angular displacements around the x , y , and z axis, respectively.

view (the same as the SAD’s body frame)). Consequently, the controllers that will be developed to steer the motions of the SAD in the x and z directions will generate, respectively, the corresponding roll and pitch angles as a percentage of their maximum allowed values, where the tilting should be executed for a given time to create the desired linear displacement with the desired speed. Appendix A represents the dynamics of a quadcopter and formulates the relationships between its linear and angular displacements.

1) STANDBY CONTROLLER

The standby controller acts as a support module in initializing and sustaining the DMT game modes. The standby controller meets four main objectives:

- 1) Finding the user in the environment
- 2) Positioning the user in the center of the SAD’s frame of view
- 3) Maintaining an appropriate, safe distance between the SAD and the user according to a proper distance estimator algorithm
- 4) Sustaining the altitude of the SAD, such that it fits the corresponding game mode

Finding the user: In case neither of the two ANNs detects a user, the standby controller steers the SAD to tilt according to a high yaw angle within a given time (in case information

is available, towards the position that the user has lastly been identified). Then the SAD analyzes the captured images and continues the entire procedure until the user is detected.

Centering the user in the SAD’s frame of view: After detecting the user, the standby controller positions the user in the center of the SAD’s frame of view according to a TSK-based FL controller with the following rules:

- R1 : If x_{user} is **left**, then $v_{SAD}^{standby} = a_1 x_{user} - b_1$
- R2 : If x_{user} is **right**, then $v_{SAD}^{standby} = a_2 x_{user} - b_2$
- R3 : If x_{user} is **center**, then $v_{SAD}^{standby} = a_3 x_{user} - b_3$ (10)

Where x_{user} is the user’s detected position in pixels, the terms “left”, “right”, and “center” (referring to the horizontal position of the user’s chest or face-box w.r.t. the center of the SAD’s frame of view) are mathematically represented by fuzzy sets, the output $v_{SAD}^{standby}$ of the standby controller is a percentage of the maximum yaw angle of the SAD, which will be executed within a given fixed time (see Figure 12), and $a_1, b_1, a_2, b_2, a_3, b_3$ are parameters that will be identified per user. Figure 13 shows one example of the membership functions for the fuzzy sets “left”, “center”, and “right”, where the user is considered to be in the center of the SAD’s frame of view, whenever the following condition is satisfied:

$$\frac{1 - 10^{-2} \cdot \tau_c}{2} x_{image} \leq x_{user} \leq \frac{1 + 10^{-2} \cdot \tau_c}{2} x_{image} \quad (11)$$

with x_{image} the maximum number of pixels in the horizontal direction of the captured images and τ_c a specific threshold that is generally personalized per user.

Maintaining an appropriate distance with the user: Sustaining effective interactions with the user for long periods of time is essential for the SAD to achieve satisfactory therapeutic outcomes via DMT. In order to achieve this goal, an adequate distance to the user must be maintained at all times. On the one hand, the SAD should avoid getting too close to the user, since a too small distance raises safety concerns and might intimidate the user, thus negatively impacting the therapeutic interactions. On the other hand, if the distance of the

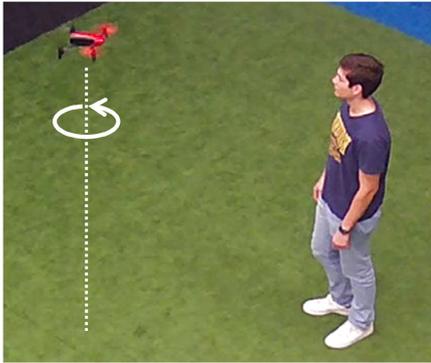


FIGURE 12. The standby controller steers the SAD to turn according to a yaw angle in order to center the user's image in its frame of view.

SAD to the user is too large, the user's attention to the SAD and thus the user's engagement may be lost. Consequently, the control system of the SAD should constantly receive the detected relative distance of the SAD and the user and adjust it whenever needed. In case the SAD is not equipped with any sensor that directly measures the longitudinal or lateral distances of the SAD to the user, the distance should be estimated based on the values that are gathered and analyzed by the IPU.

In order to check whether the relative distance of the SAD and the user is too small, the values provided by ANN 1 are used because the face-box can properly be detected when the user and the SAD are close enough to each other, while for a too close distance ANN 2 fails to detect the entire body of the user. Based on the width w_{fb} and height h_{fb} of the latest face-box (see Figure 3), the area A_{fb} of the face-box in squared pixels is computed and is divided by the total image area A_{image} in squared pixels to determine the face-box ratio. Whenever the magnitude of the face-box ratio is large, the SAD is considered to be too close to the user and thus their distance should be increased according to the following fuzzy rule:

$$\mathbf{R4} : \text{If } \frac{A_{fb}}{A_{image}} \text{ is large, then } \pi_{SAD}^{standby} = a_4 A_{fb} + b_4 \quad (12)$$

Where the term "large" is mathematically represented by a fuzzy set and its corresponding fuzzy membership function, $\pi_{SAD}^{standby}$ is the SAD's pitch displacement as a percentage of its maximum allowed value, and parameters a_4 and b_4 will be identified per user. Figure 14 shows a fuzzy membership function for the term "large", where the threshold $\tau_{d,1}$ is identified per user based on the distance they find safe.

To assess whether the SAD is too far from the user, the values provided by ANN 2 are used, since in larger distances the estimates provided by ANN 1 are unreliable. Out of the 18 joints illustrated in Figure 4, the coordinates of joints 1, 8, and 11 corresponding to the chest and the right and left edges of the user's waistline are considered (see Figure 4). Note that joint 1 is the reference joint, i.e., all other joints are detected w.r.t. joint 1. Therefore, whenever information

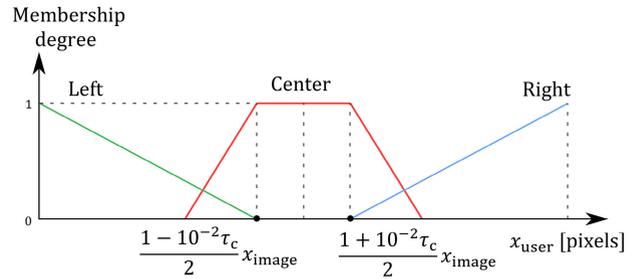


FIGURE 13. Membership functions corresponding to fuzzy sets "left", "center", and "right" used by the standby controller to position the user in the SAD's frame of view.

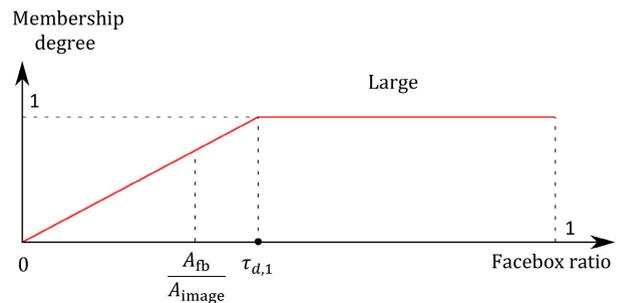


FIGURE 14. Membership function corresponding to fuzzy set "large" used by the standby controller to estimate the distance of the user with respect to the SAD.

from ANN 2 is available, it certainly includes data regarding joint 1. Moreover, while the legs and arms of a user (i.e., joints 2-13 in figure 4) may move out of the SAD's frame of view, the chances are high that the data regarding joints 8 and 11 is available whenever the user is not too close to the SAD. Finally, the joints corresponding to the user's face (i.e., joints 0 and 14-17) are more prone to being swayed and thus less reliable.

Once the coordinates for joints 1, 8, and 11 are known, the vertical distance $\delta_{1,11}$ between joints 1 and 11, and the vertical distance $\delta_{1,8}$ between joints 1 and 8 (see Figure 4) are divided by the overall vertical length y_{image} of the image to generate the upper body ratios. The closer the user to the SAD, the larger the values of the upper body ratios. More specifically, whenever the maximum of the two values estimated for the upper body ratio is small, the distance of the SAD and the user is too large and should be reduced according to the following rule:

$$\mathbf{R5} : \text{If } \max \left\{ \frac{\delta_{1,8}}{y_{image}}, \frac{\delta_{1,11}}{y_{image}} \right\} \text{ is small, then } \pi_{SAD}^{standby} = a_5 \max\{\delta_{1,8}, \delta_{1,11}\} + b_5 \quad (13)$$

Where the term "small" is mathematically represented by a fuzzy set and its corresponding fuzzy membership function and parameters a_5 and b_5 will be identified per user. Figure 15 shows one example for representing the term "small" by a fuzzy membership function, where the threshold $\tau_{d,2}$ is identified based on the relative distance between the SAD and

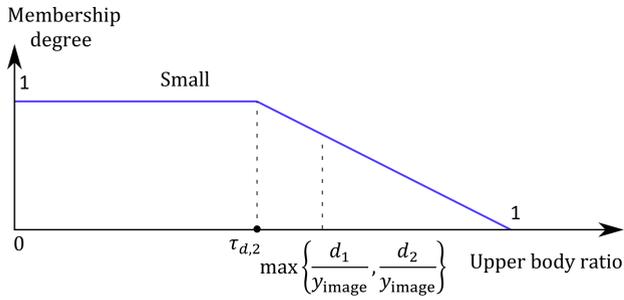


FIGURE 15. Membership function corresponding to fuzzy set “small” used by the standby controller to estimate the distance of the user with respect to the SAD.

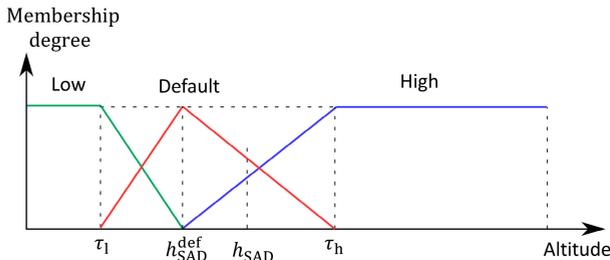


FIGURE 16. Membership functions corresponding to fuzzy sets “low”, “default”, and “high” for assessing the SAD’s altitude.

the user, for which the IPU can still perform satisfactorily, while the user remains attentive to the SAD.

Sustaining the SAD’s altitude: When the SAD’s altitude is too high or too low, two issues may occur: (1) The user may not accurately capture all the movements and the corresponding coordinates of the SAD, which negatively impacts the human-SAD interactions. (2) The IPU may fail to capture the movements and the corresponding coordinates of the user correctly, which negatively impacts the analysis and decision making of the SAD. Most quadcopters are equipped with an altitude measurement sensor. For the SAD, these measurements are used in two cases during the DMT sessions: (1) After the SAD accomplishes a game mode and intends to transition to a new game mode. (2) After the SAD is done executing a vertical movement.

In order to regularly sustain a proper altitude for the SAD, three fuzzy sets corresponding to the concepts “short”, “default”, and “high” are defined. Figure 16 shows one example, where thresholds τ_l and τ_h and the default altitude h_{SAD}^{def} of the SAD specify the corresponding membership functions. Generally speaking, the user’s height h_{user} is considered to identify the thresholds τ_h and τ_l . More specifically, $\tau_h = \alpha_h h_{user}$ and $\tau_l = \alpha_l h_{user}$ with all the altitudes given in cm and $\alpha_h > 1$ and $0 < \alpha_l < 1$ parameters that will be identified per user. The default altitude h_{SAD}^{def} of the SAD is determined corresponding to the user’s shoulder level (see Figure 17). The ratio of the body of a user of height h_{user} above their shoulder is almost $0.15h_{user}$. Therefore, we consider $h_{SAD}^{def} = 0.85h_{user}$.

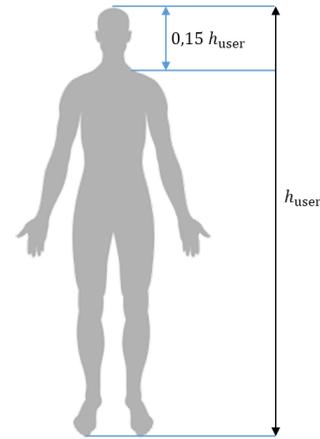


FIGURE 17. Standard ratios of the user’s body above the shoulders (almost 0.15 of the user’s height) used to adjust the default altitude of the SAD.

To maintain the SAD’s altitude, the standby controller implements the following fuzzy rules:

- R6 : If h_{SAD} is **low**,
then $\zeta_{SAD}^{standby} = a_6 h_{SAD} + b_6$
- R7 : If h_{SAD} is **default**,
then $\zeta_{SAD}^{standby} = a_7 h_{SAD} + b_7$
- R8 : If h_{SAD} is **high**,
then $\zeta_{SAD}^{standby} = a_8 h_{SAD} + b_8$ (14)

Where $\zeta_{SAD}^{standby}$ is the speed of the SAD in the y direction as a percentage of its maximum value and parameters $a_6, b_6, a_7, b_7, a_8, b_8$ will be identified per user.

2) PASSIVE GAME MODE 1: MIRROR THE USER

In passive game modes the SAD mainly responds, via mirroring, to the user’s movements, where the main goal of the SAD is to accurately replicate all relevant motions of the user that have been captured during the wait time, including vertical and horizontal motions, and their combinations. In passive game mode 1, the SAD mirrors the motions of the user’s chest position (see Figure 18). In order to start the interactions, the SAD emits a speech signal (via the speakers of a remote computer) to invite the user to move.

After analyzing the motions captured within the wait time and deducing the time interval (in s) corresponding to the movement and the average horizontal and vertical speeds (in pixels/s), these quantities are given to the controller. Three quantities are output from the controller: (1) the mimicking time (in s), (2) the roll displacement (in deg) of the SAD, such that it results in a desired horizontal displacement, and (3) the vertical speed (in m/s) of the SAD such that within the given mimicking time the desired vertical displacement is obtained. The mimicking time of the SAD is the time associated to the identified motion category. For the horizontal movements of the SAD, the following fuzzy rules are used to

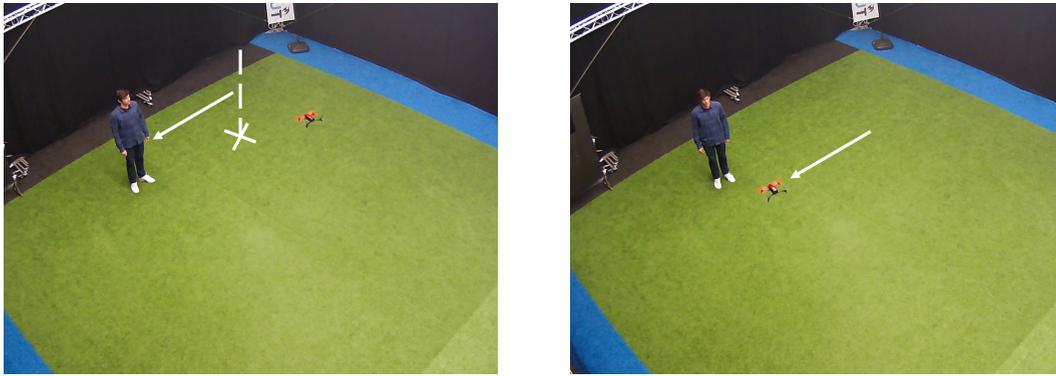


FIGURE 18. Passive game mode 1: The standby controller has originally centered the user's face-box in the SAD's frame of view); the user moves to the left from the perspective of the SAD (left-hand side picture), and the SAD mimics the same motion (right-hand side picture).

determine the percentage ρ_{SAD}^{GM1} of the maximum roll angle of the SAD, which should be executed by the SAD for the given mimicking time:

- R9** : If \dot{x}_{user}^{chest} is **small**,
 then $\rho_{SAD}^{GM1} = a_9 \dot{x}_{user}^{chest} + b_9$
- R10** : If \dot{x}_{user}^{chest} is **medium**,
 then $\rho_{SAD}^{GM1} = a_{10} \dot{x}_{user}^{chest} + b_{10}$
- R11** : If \dot{x}_{user}^{chest} is **large**,
 then $\rho_{SAD}^{GM1} = a_{11} \dot{x}_{user}^{chest} + b_{11}$ (15)

with \dot{x}_{user}^{chest} the average horizontal speed of the user's chest in pixels/s. The terms "small", "medium", and "large" are mathematically represented by fuzzy sets and their corresponding fuzzy membership functions, and the parameters a_9 , b_9 , a_{10} , b_{10} , a_{11} , and b_{11} will be identified per user.

The range of the vertical displacements of the SAD, in practice, are small in game mode 1 since the human's chest cannot move large distances in the vertical direction. Therefore, the controller reduces to a proportional control policy that determines a percentage ζ_{SAD}^{GM1} of the maximum allowed vertical speed of the SAD:

$$\zeta_{SAD}^{GM1} = K^{GM1} \dot{y}_{user}^{chest} \quad (16)$$

with \dot{y}_{user}^{chest} representing the average vertical speed of the user's chest in pixels/s and K^{GM1} a tuning parameter.

3) PASSIVE GAME MODE 2: MIRROR THE USER'S HAND

In passive game mode 2, the SAD mirrors the movements of the user's hands (see Figure 19). The SAD encourages the user by emitting the speech "waiting for your hand motion!". During the wait time the SAD captures and analyzes all the movements of the user's hands. There are two main differences in passive game mode 2 compared to passive game mode 1:

- 1) The positions r_{ti} that are recorded and analyzed by the IPU correspond to the wrist positions provided by ANN2 (see joints 4 and 7 in Figure 4). The controller corresponding to game mode 2 first decides which hand

to follow by estimating the total distance (in pixels) that is travelled by each wrist during the wait time. The hand that has been more active (i.e., with a larger total travelled distance) is selected.

- 2) Compared to the chest point, capturing the wrist positions is more prone to failures or errors for the IPU since the movements of the hands may correspond to frequencies that are sometimes higher than the real-time frequency of capturing data by the IPU. Therefore, missing or erroneous data may occur more often and the fault tolerance algorithm (see Section III-C1) may more frequently be called by the controller in game mode 2.

The TSK-based FL controller corresponding to game mode 2 performs according to the following rule base for the horizontal movements of the SAD:

- R12** : If \dot{x}_{user}^{wrist} is **small**,
 then $\rho_{SAD}^{GM2} = a_{12} \dot{x}_{user}^{wrist} + b_{12}$
- R13** : If \dot{x}_{user}^{wrist} is **medium**,
 then $\rho_{SAD}^{GM2} = a_{13} \dot{x}_{user}^{wrist} + b_{13}$
- R14** : If \dot{x}_{user}^{wrist} is **large**,
 then $\rho_{SAD}^{GM2} = a_{14} \dot{x}_{user}^{wrist} + b_{14}$ (17)

with \dot{x}_{user}^{wrist} the average horizontal speed of the user's (more active) wrist in pixels/s, and ρ_{SAD}^{GM2} the percentage of the maximum roll displacement of the SAD that results in the desired horizontal movement. The terms "small", "medium", and "large" will mathematically be represented by fuzzy sets and their corresponding fuzzy membership functions, and the parameters a_{12} , b_{12} , a_{13} , b_{13} , a_{14} , and b_{14} will be identified per user.

For the vertical displacements of the SAD, a proportional control policy provides a percentage ζ_{SAD}^{GM2} of the maximum allowed vertical speed of the SAD to steer it according to game mode 2:

$$\zeta_{SAD}^{GM2} = K^{GM2} \dot{y}_{user}^{wrist} \quad (18)$$

Where \dot{y}_{user}^{wrist} is the average vertical speed of the user's wrist in pixels/s for the active hand.

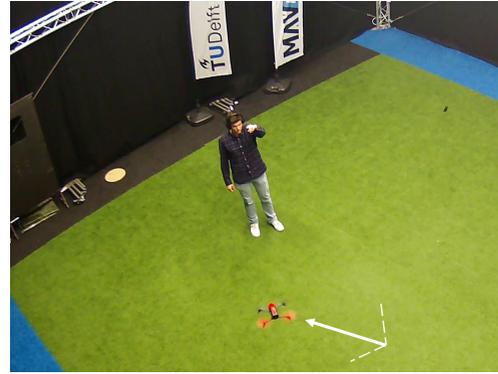


FIGURE 19. Passive game mode 2: The user initiates a diagonal motion with the left hand (left-hand side picture) and the SAD mimics (a scaled version of) the diagonal motion of the user’s hand (right-hand side picture).

Remark 3: In game mode 1, the parameters of the corresponding TSK-based FL controllers are tuned such that the SAD mimics the movements of the user’s body accurately (i.e., with scale 1). In game mode 2, however, the parameters of the corresponding TSK-based FL controllers are tuned such that the resulting movements of the SAD are scaled with a factor larger than 1 compared to the movements of the user’s hand. This is mainly to stress the movements of the user in game mode 2 for a more engaging interaction.

4) ACTIVE GAME MODE 3: MIMIC THE SAD WITH BODY MOTIONS

Game modes 3 and 4 are active, meaning that the SAD interacts with the user by initiating movements that the user should mimic. To start, the SAD emits a speech signal “follow me with your body” for game mode 3 and “follow me with your hand” for game mode 4 (via the speakers of a remote computer). The amplitude of the SAD’s movements in active game modes should be personalized per user. After executing a movement, the SAD pauses according to the wait time to give the user the chance to follow the SAD’s movement (see Figure 20). Afterwards, using the motion processing algorithm, the SAD analyzes and assesses the user’s data that is captured via the IPU, and attributes a *performance score* to the analyzed movement of the user, where this score implies how well the motion of the SAD has been mimicked by the user. In order to further stimulate the interactions with the user, the controllers corresponding to active game modes respond to higher performance scores with larger displacement amplitudes. Gradually, the frequency, category, duration, direction, and speed of the SAD’s movements are adapted according to the responses received from the user and the performance scores attributed (see Section III-E).

While the first movement suggested by the SAD may be selected randomly, for the consecutive movements the input to the TSK-based FL controller of active game mode 3 is the horizontal distance Δx_{user} in pixels between the last registered position x_{user} of the user and the center of the SAD’s frame of view. The output ρ_{SAD}^{GM3} of the corresponding FL controller is a percentage of the maximum

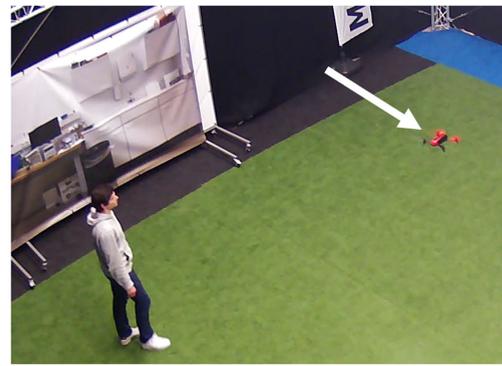


FIGURE 20. Active game mode 3: While the user is mimicking the SAD, the SAD captures the user’s movements via its IPU and assesses the user’s performance.

allowed roll displacement of the SAD based on the following rule base:

- R15 : If Δx_{user} is **negligible**,
then $\rho_{SAD}^{GM3} = a_{15} \Delta x_{user} + b_{15}$
- R16 : If Δx_{user} is **significant**,
then $\rho_{SAD}^{GM3} = a_{16} \Delta x_{user} + b_{16}$ (19)

Where the terms “negligible” and “significant” will mathematically be represented by fuzzy sets and their corresponding membership functions and parameters a_{15} , b_{15} , a_{16} , and b_{16} will be identified according to the preferences and responses of every user.

A proportional control policy, similar to the previous game modes, may also be considered to steer the vertical movements of the SAD, i.e.:

$$\zeta_{SAD}^{GM3} = K^{GM3} \Delta y_{user} \tag{20}$$

Where ζ_{SAD}^{GM3} is a percentage of the maximum vertical speed of the SAD and Δy_{user} is the vertical distance between the last registered position of the user and the center of the SAD’s frame of view.

5) ACTIVE GAME MODE 4: MIMIC THE SAD WITH HAND MOTIONS

In active game mode 4, the SAD expects the user to follow its movements by their hand. The TSK-based FL controller corresponding to game mode 4 for generating the horizontal movements of the SAD consists of the following rule base:

- R17** : If Δx_{hand} is **negligible**,
then $\rho_{SAD}^{GM4} = a_{17}\Delta x_{hand} + b_{17}$
- R18** : If Δx_{hand} is **significant**,
then $\rho_{SAD}^{GM4} = a_{18}\Delta x_{hand} + b_{18}$ (21)

Where the terms “negligible” and “significant” are represented by fuzzy sets and their corresponding membership functions, ρ_{SAD}^{GM4} is a percentage of the maximum allowed roll displacement of the SAD such that it results in the desired horizontal displacement, $a_{17}, b_{17}, a_{18},$ and b_{18} are parameters that will be identified per user, and Δx_{hand} is the horizontal distance between the wrist of the user and the center of the SAD’s frame of view. Whenever data regarding both hands of the user is available, the hand that is closer to the center of the SAD’s frame of view is considered.

For the vertical movements of the SAD, the following proportional control policy is used:

$$\zeta_{SAD}^{GM4} = \min \left\{ K^{GM4} \cdot \frac{1}{\Delta y_{hand}}, \zeta_{SAD}^{max} \right\} \quad (22)$$

Where ζ_{SAD}^{GM4} is a percentage of the maximum vertical speed of the SAD, ζ_{SAD}^{max} further limits the maximum vertical displacements of the SAD during active game mode 4 (since hands are expected to mirror more abrupt sequences of motions of the SAD, the scale of these motions should be scaled down), and Δy_{hand} is the vertical distance between the user’s hand and the center of the drone’s frame of view. Note that a larger value for Δy_{hand} corresponds to a worse performance for the user, and in response to that the SAD makes a smaller displacement according to (22).

E. SAD’s ADAPTIVITY AND PERSONALIZATION

Adaptivity and personalization are essential characteristics required for the SAD’s control system. Adaptivity refers to updating the parameters that identify DMT plans (e.g., the wait time of the SAD) or parameters of the controllers of the SAD according to the conditions of users and the DMT sessions. Personalization refers to tuning such parameters or selecting the order, frequency, and duration of various game modes in a user-specific way. While personalization parameters may remain constant for that user after being identified, parameters that are adaptive may vary more frequently. The designer can specify whether a tuning parameter is within one of these two categories. Note that some of the adaptive parameters may be a function of personalization parameters (e.g., while the SAD’s wait time is an adaptive parameter that may vary in the course of DMT sessions, its upper or lower values depend on the preference of every user).

Different adaptivity modules are developed for passive and active game modes based on two main metrics: (1) user’s performance, which implies how well the user has accomplished a particular task (i.e., initiating or mirroring) corresponding to a specific game mode, (2) user’s engagement, which is based on whether or not a user is attempting to interact with the SAD. While a high performance implies a high level of engagement, the opposite is not necessarily true. On the one hand, the adaptivity policies that are developed lead the controller to act more leniently towards a user who exhibits a low performance but a high engagement. On the other hand, for a high performance (which also implies a high engagement), the adaptivity approaches make the game modes more challenging for the user.

1) ADAPTIVITY FOR PASSIVE GAME MODES

Within a *trial* in passive game modes the SAD (1) pauses according to the wait time for the user to move and simultaneously captures data from the user, (2) processes the captured movements, and (3) mimics the analyzed movements. An *episode* is the total number of trials in one continuous game mode. Both the performance and level of engagement of the user in passive game modes are quantified according to the number of inactive trials, i.e., trials with no significant (based on a threshold, e.g., a multiple of τ_m) motions. Good performance and engagement are associated with low numbers of inactive trials.

Passive game modes involve two adaptive parameters: the wait time t_w per trial and the number N_{trial} of trials per episode. These parameters are updated at the end of each episode. The wait time is adjusted according to the episodic average pause time \bar{t}_{ep} , which is the average time since the user finishes a movement that the SAD should mimic until the end of the current wait time. A limiting interval $[\bar{t}_{ep}^l, \bar{t}_{ep}^u]$ is considered per user for \bar{t}_{ep} . The wait time is updated according to the following relationships:

$$t_w \leftarrow \max \{ t_w - (\bar{t}_{ep} - \bar{t}_{ep}^u), t_w^l \} \quad \text{when } \bar{t}_{ep} > \bar{t}_{ep}^u \quad (23)$$

$$t_w \leftarrow \min \{ t_w + (\bar{t}_{ep}^l - \bar{t}_{ep}), t_w^u \} \quad \text{when } \bar{t}_{ep} < \bar{t}_{ep}^l \quad (24)$$

Where t_w^l and t_w^u are the lower and upper values for the wait time, with $\bar{t}_{ep}^l, \bar{t}_{ep}^u, t_w^l,$ and t_w^u user-specific parameters.

The number N_{trial} of trials per episode is updated according to the number of inactive trials. Whenever this number is equal to or larger than a threshold, N_{trial} is reduced 1 unit. Otherwise, the user is considered to be fully engaged in the game mode and N_{trial} is increased by 1. In order to prevent many repetitions or avoidance of a game mode (which reinforce users to remain inside their comfort zone), user-specific lower N_{trial}^l and upper values N_{trial}^u for N_{trial} are considered.

2) ADAPTIVITY FOR ACTIVE GAME MODES

A main goal of the adaptive module in active game modes is to ensure that, on the one hand, the motions performed by the SAD challenge the user and result in an increased engagement

level and, on the other, these motions are always feasible for the user to mimic and do not endanger the user or damage the SAD. For an active game mode, within a *trial* the SAD (1) initiates a motion, (2) pauses according to the wait time for the user to follow the motion and simultaneously captures data from the user, and (3) Evaluates the performance and engagement of the user according to the captured data. The total number of trials in one continuous game mode is called an *episode*.

At the end of every episode, the adaptive parameters may be updated based on the episodic average performance \bar{p}_{ep} (i.e., the mean value of the performance scores corresponding to all the trials in that episode) and episodic average engagement \bar{e}_{ep} of the user. In active game modes, a good performance for the user is identified whenever the user's final position during the wait time is close enough (based on the threshold τ_c given in (11)) to the center of the SAD's frame of view. Additionally, whenever the performance score is high, the engagement level is also high. Otherwise, the engagement level receives a low score, unless - despite a poor performance score - the user exhibits noticeable activity or large movements (i.e., displacements with an amplitude larger than a multiple of τ_m).

Remark 4: Whenever the episodic average engagement \bar{e}_{ep} is larger than a specific value the adaptivity procedure is triggered. Otherwise, the personalization module (will be detailed in Section III-F) may decide to completely skip that game mode.

For active game modes, the parameters that specify the membership functions corresponding to the terms “negligible” and “significant” in (19) and (21) are adaptive. For Gaussian membership functions, the standard deviations σ_{neg} and σ_{sig} will be updated according to the following relationships:

$$\sigma_{neg} \leftarrow \max \{ \sigma_{neg} - \Delta\sigma, \sigma^l \} \quad (25)$$

$$\sigma_{sig} \leftarrow \min \{ \sigma_{sig} + \Delta\sigma, \sigma^u \} \quad (26)$$

with σ^l and σ^u lower and upper values for the standard deviations that may be personalized per user. In general, when the average episodic performance \bar{p}_{ep} is high, the adaptive module makes the game mode more challenging for the user by providing a more strict definition for the terms “negligible” and “significant”, which is realized by decreasing σ_{neg} and increasing σ_{sig} . Considering α_σ and β_σ as parameters that should be identified based on real-life DMT interactions, we define:

$$\Delta\sigma = \alpha_\sigma \bar{p}_{ep} + \beta_\sigma \quad (27)$$

The parameters a_{15} - a_{18} and b_{15} - b_{18} in (19) and (21) are also adaptive parameters that will be updated according to the following relationships for $i = 15, 16, 17, 18$:

$$a_i \leftarrow \min \left\{ \max \{ \lambda_a a_i, a_i^l \}, a_i^u \right\}$$

$$\lambda_a = \alpha_a \bar{p}_{ep} + \beta_a, \text{ where } |\alpha_a| < 1, |\beta_a| < 1 \quad (28)$$

$$b_i \leftarrow \min \left\{ \max \{ \lambda_b b_i, b_i^l \}, b_i^u \right\}$$

$$\lambda_b = \alpha_b \bar{p}_{ep} + \beta_b, \text{ where } |\alpha_b| < 1, |\beta_b| < 1 \quad (29)$$

Note that the upper and lower values $a_i^u, b_i^u, a_i^l, b_i^l$ are determined such that unsafe movements of the SAD are prevented. Moreover, the personalization parameters $\alpha_a, \beta_a, \alpha_b, \beta_b$ are determined such that in the upcoming trials the corresponding controller provides larger movement amplitudes for higher average episodic performances, and vice versa. For vertical movements the parameters K^{GM3} and K^{GM4} in (20) and (22) are considered as adaptive. For $i = 3, 4$ we have:

$$K^{GMi} \leftarrow \min \left\{ \max \{ \lambda^{GMi} K^{GMi}, K^{GMi,l} \}, K^{GMi,u} \right\}$$

$$\lambda^{GMi} = \alpha^{GMi} \bar{p}_{ep} + \beta^{GMi} \quad (30)$$

with $K^{GMi,u}, K^{GMi,l}, \alpha^{GMi}$, and β^{GMi} personalization parameters.

Similarly to passive game modes, in active game modes the wait time t_w is adaptive. In active game modes, however, adaptation of t_w is according to the user's episodic average performance, i.e.:

$$t_w \leftarrow \min \left\{ \max \{ t_w + \alpha_w \bar{p}_{ep} + \beta_w, t_w^l \}, t_w^u \right\} \quad (31)$$

with α_w, β_w, t_w^l , and t_w^u personalization parameters.

Finally, the parameter τ_c is adaptive and will be updated according to the following equation:

$$\tau_c \leftarrow \min \left\{ \max \{ \tau_c - \alpha_c \bar{p}_{ep} + \beta_c, \tau_c^l \}, \tau_c^u \right\} \quad (32)$$

Where $\alpha_c, \beta_c > 0$ and $\alpha_c > \beta_c$. Based on (32), when a user exhibits a high average episodic performance \bar{p}_{ep} , τ_c varies such that the control system of the SAD becomes more strict in scoring the user's performance.

F. PERSONALIZATION VIA MASTER CONTROLLER

The personalization module of the SAD is executed via a master controller (see Figure 21). At the end of every episode, the master controller decides which game mode to be executed next (a game mode may be selected several times consecutively). Therefore, a non-zero weight w_i ($i = 1, 2, 3, 4$) is assigned to each game mode i such that $\sum_{i=1}^4 w_i = 1$. The weight w_i corresponds to the chance of game mode i for being selected by the master controller. At the end of every episode, these weights may be updated such that a game mode that has proven to be highly engaging for the user receives a higher weight. The following relationship is used to update the values of the weights for $i \in \{1, \dots, 4\}$ and $j \in \{1, \dots, 4\}/\{i\}$:

$$w_i \leftarrow \min \left\{ \max \{ w_i + (-1)^\ell \Delta w, w^{\min} \}, 1 \right\} \quad (33)$$

$$w_j \leftarrow \min \left\{ \max \{ w_j - (-1)^\ell \frac{\Delta w}{3}, w^{\min} \}, 1 \right\} \quad (34)$$

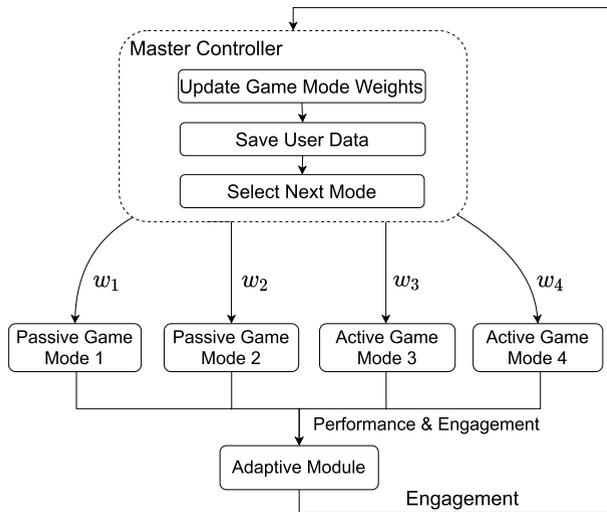


FIGURE 21. Master controller personalizes the DMT plans.

Where $\ell = 0$ for \bar{e}_{ep} above a given threshold, and 1 otherwise. Moreover, Δw is a design parameter and w^{\min} is a minimum value for the weight of each game mode, where $w^{\min} > 0$ in order to prevent a specific game mode to be completely excluded from the DMT plans.

In addition to updating and storing the weights corresponding to various game modes, the master controller stores and makes use of the personal information of a user (e.g., their height) that affect the SAD's decisions. Moreover, the master controller stores a brief report from every DMT session per user, including their average episodic performance values, the number of inactive trials, the profiles for the weights of various game modes, and the level of engagement of the users for each game mode.

Figure 22 recaps the main steps of the human-SAD interaction in a simplified flowchart.

IV. CASE STUDY

Next we introduce the setup and the main facilities that have been used in the experiments.

A. DRONE USED AS SAD

A Parrot Bebop 2 drone (see Figure 23) was used as the SAD in all the experiments. Parrot Bebop 2 is a small quadcopter that weighs 500 g and is equipped with a 2700 mAh battery, which enables the drone to fly non-stop for almost 10 min. Two spare fully charged batteries with the same capacity were available to enable continuous DMT sessions of around 30 min per user. Parrot Bebop 2 has a 14 MP camera (which records 1080 p video at 30 fps) and a WiFi network that allows the drone to connect to remote computers or tablets. The small size, high battery power, safe propellers, high-quality camera, and simple appealing appearance of Parrot Bebop 2, make it suitable as SAD for DMT.

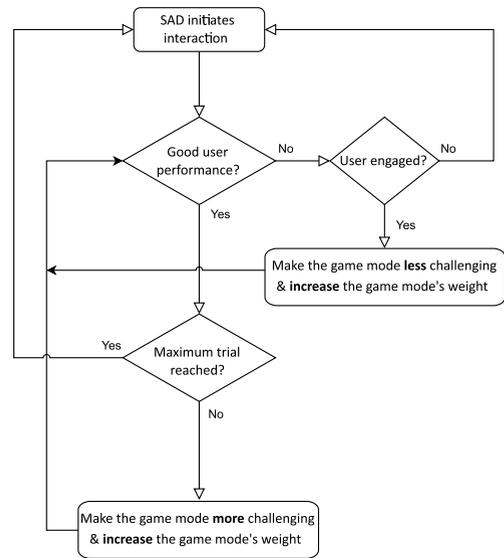


FIGURE 22. Flowchart summarizing the human-SAD interaction steps.



FIGURE 23. Top view of a Parrot Bebop 2 drone.

B. EXPERIMENTAL SETUP

All DMT sessions were simulated with one person at a time as the user in the Cyber Zoo⁴ at the Delft University of Technology, where a 10×10 m² synthetic turf surrounded by safety nets is available. Twelve high-tech cameras are installed at the Cyber Zoo and record the interactions of the user and the SAD.

The computations corresponding to the online data analysis and decision making by the SAD were performed on a remote computer that was connected to the SAD via its WiFi network. In addition to saving battery power for a longer performance of the SAD, running the computations off-board allows for more real-time computational power, and for using a high-level, interpreted programming language (e.g., Python, Olympe library [29], Pyparrot library [30], and Pyfuzzylite library [31]).

C. PARTICIPANTS

In total, 10 participants took part in the experiments. Each experiment was analyzed according to three time intervals of 8-12 min (overall 24-36 min), which covered various

⁴<https://tudelftroboticsinstitute.nl/labs/cyber-zoo>

TABLE 1. Parameters used in the case study to develop the IPU and the controllers of the SAD.

Motion processing			Standby controller											FL controllers			
τ_m [pixels]	γ [-]	\bar{C} [-]	$\max \psi$ [deg/s]	τ_c [%]	x_{image} [pixels]	y_{image} [pixels]	$\tau_{d,1}$ [-]	$\tau_{d,2}$ [-]	$\max \phi$ [deg]	t_ϕ [s]	α_l [-]	α_h [-]	$\max \theta$ [deg]	t_θ [s]	$\max v_y$ [m/s]	t_{v_y} [s]	
9	0.2	2	80	15	856	480	0.04	0.2	20	1	0.375	1.375	20	1	1	1	

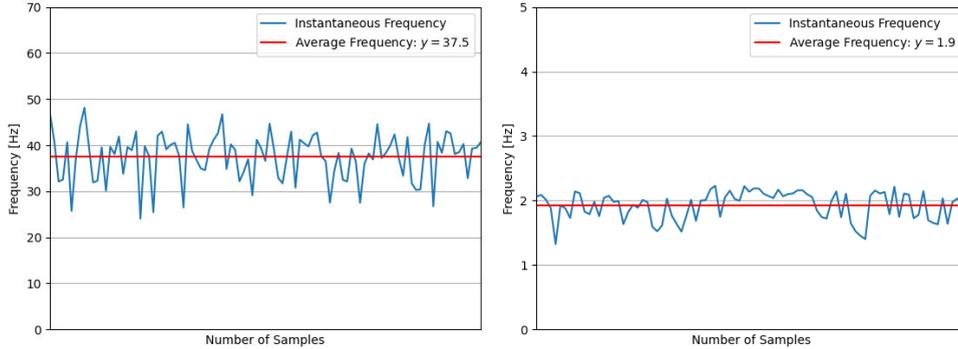


FIGURE 24. Frequency of capturing pictures by the SAD’s camera (left-hand side plot) and real-time IPU performance including both ANN 1 and ANN 2 (right-hand side plot).

environmental and personal changes and allowed both the personalization and adaptation modules to be assessed properly. The age and height of participants varied between 21-24 years and 155-185 cm. Before the experiments, participants were informed about the nature of the four game modes, adaptability of the SAD’s behaviour with respect to their preferences, and that they were not obliged to continue interacting with the SAD. Before running the experiments, the four game modes were executed for the participants in order to familiarize them with the game modes, and if any interactions occurred at this stage, the SAD already used the corresponding information to execute adaptation and personalization.

D. PARAMETERS

The main parameters used to develop the IPU and the controllers of the SAD are given in Table 1.

In order to assess and score the performance of the user in active game modes 3 and 4, three performance scores, 0, 0.5, and 1 were used. Users received a score 1 whenever their final position was centred (based on the threshold τ_c) in the frame of view of the SAD. A score 0.5 was given to users whenever they moved in the correct direction, although the last captured position after the wait time was not perfectly centered. In case neither of the above cases occurred, users received a score 0.

E. IMPLEMENTING THE IPU

In order to train the ANNs (see Section III-B), data collected from movies on YouTube and the COCO dataset [32] were used. Since the motion processing algorithm and the SAD’s controller receive the output of the IPU as their input, the real-time performance of the IPU based on the frequency of image processing (defined as the inverse of the time elapsed to analyze two consecutive images) was evaluated.

The left-hand side plot in Figure 24 illustrates the frequency of the SAD’s camera (i.e., the reciprocal of the time required by the camera to capture two consecutive images) for 100 sample images, which shows an average frequency of almost 38 Hz. By incorporating ANN 1 and ANN 2 in the IPU, the average frequency of the image processing (see the right-hand side plot in Figure 24) becomes almost 2 Hz (more specifically, the image processing frequency is between 1.5 Hz and 2 Hz). Therefore, the real-time performance of the SAD’s controller corresponds to frequencies between 1.5 Hz and 2 Hz (i.e., there is 0.5 s to 0.7 s between two IPU estimations).

F. IMPLEMENTING THE MOTION PROCESSING ALGORITHM

See Table 1 for the values of the parameters used in the motion processing algorithm explained in Section III-C1. Whenever at least three consecutive registers of the IPU contained invalid information (which, assuming an average image processing frequency of 2 Hz implies that the user has not been detected for at least 1.5 s), the fault tolerance algorithm was activated.

G. IMPLEMENTING THE STANDBY CONTROLLER

See Table 1 for the parameters used in implementing the standby controller. In order to detect and position a user, the standby controller allowed the SAD to follow a yaw rate equal to 50% of the maximum allowed yaw rate.

In order to center the user in the SAD’s frame of view, the membership functions illustrated in Figure 25 were used for “left”, “center”, and “right” in (10). These membership functions are Gaussian, which are mathematically identified by their mean and standard deviation, and are preferred over alternative options due to their smoothness and concise

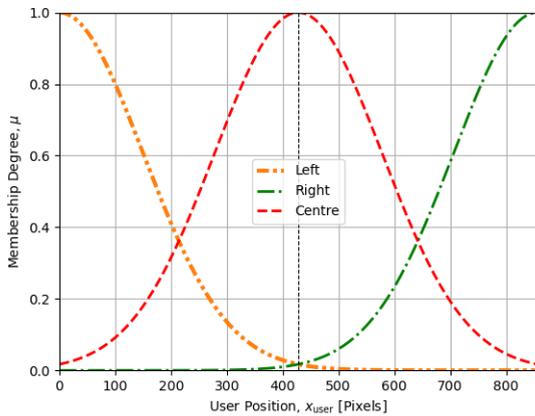


FIGURE 25. Membership functions for the terms “left”, “center”, and “right” used by the standby controller to center the user’s image in the SAD’s frame of view.

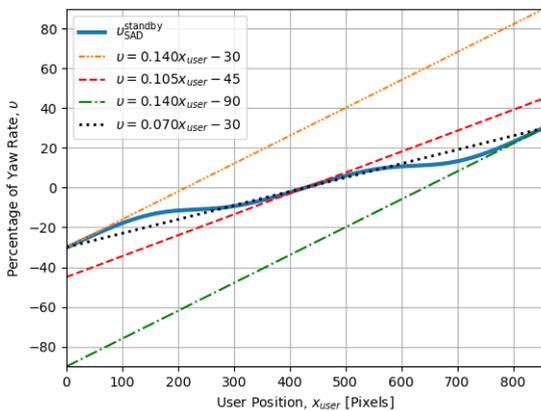


FIGURE 26. Standby controller centering the user’s image in the SAD’s frame of view: Input-output mapping.

mathematical notations (i.e., the number of parameters that should be tuned is reduced), which improve the adaptability and computational efficiency of the SAD’s controller in real time. The mean and standard deviation of the Gaussian membership functions were initially (i.e., before the adaptivity module adjusts these parameters) selected such that the corresponding functions are equally spaced in the horizontal range of the image and cover all possible input values for x_{user} (see Figure 25).

The parameters a_1 - a_3 and b_1 - b_3 in (10) were tuned to $a_1 = 0.140$ 1/pixels, $a_2 = 0.105$ 1/pixels, $a_3 = 0.140$ 1/pixels, $b_1 = 30$, $b_2 = 45$, $b_3 = 90$. Figure 26 shows the resulting input-output mapping corresponding to (10) for all possible input values (see the solid blue curve). The dash-double-dotted orange, dashed red, and dash-dotted green curves in Figure 26 are properly tuned outputs corresponding to, respectively, rules R1, R2, and R3 in (10). Note that whenever the input variable x_{user} reaches its minimum or maximum values, the output of the standby controller is equal to, respectively, -30% and 30% of the maximum yaw rate, resulting in the least overshoot when the SAD attempts to center the

user’s image. Moreover, when x_{user} is close to the center, the variations in the output of the standby controller provide a trade-off between a smooth motion and a reasonable speed (not less than 10% of the maximum yaw rate).

In order to maintain an appropriate distance between the SAD and the user, the parameters $\tau_{d,1}$ and $\tau_{d,2}$ (see Table 1) were selected such that the distance of the user and the SAD remains between 0.5 m (which was considered safe and convenient according to the majority of the participants) and 3 m (suited for capturing high-quality pictures). Note that due to the nature of the experiments (i.e., the fact that most participants had limited possibilities to move in the z direction), in most cases (12) and (13) were not triggered, or whenever they were triggered, the SAD managed to sustain the desired distance by simply executing the maximum pitch tilt for 1 s. Therefore, a_4 , b_4 , a_5 , and b_5 were not tuned in our experiments.

For sustaining the SAD’s altitude, the values given in Table 1 for α_1 and α_h were used to estimate the thresholds τ_h and τ_l . Since the variations in the vertical movement by the participants were limited, instead of considering (14), after estimating the difference Δh_{SAD} between the latest altitude h_{SAD} registered by the SAD’s sensors and the default altitude h_{SAD}^{def} , the standby controller executed the maximum vertical speed for Δh_{SAD} seconds to position the SAD according to the default altitude.

H. IMPLEMENTING THE FL CONTROLLER OF PASSIVE GAME MODE 1

The fuzzy membership functions used for the terms “small”, “medium”, and “large” are illustrated in Figure 27. The maximum horizontal speed of the user’s chest, 100 pixels/s, was determined based on several experiments with the participants. The mean values (0, 50, and 100 pixels/s) of the Gaussian membership functions were selected such that the resulting membership functions are equally spaced across the domain of the horizontal speed of the user’s chest. Moreover, the standard deviation of the Gaussian membership functions were set to 20 pixels/s, since this value allows any realization within the speed domain to correspond to a substantially high membership degree with regard to at least one of the membership functions. Moreover, the tuned values of the parameters in (15) include $a_9 = -0.08$ s/pixels, $a_{10} = -0.16$ s/pixels, $a_{11} = -0.08$ s/pixels, $b_9 = 0$, $b_{10} = -4$, and $b_{11} = 0$.

The resulting TSK-based FL controller for passive game mode 1 yields the input-output mapping illustrated in Figure 28. In general, whenever the speed of the user’s chest has more significant membership degrees corresponding to the sets “small” and “big” (see Figure 27) the output (see the solid blue curve in Figure 28) of the FL controller mainly approximates the behavior of a proportional controller with the gain 0.08 s/pixels (see the dash-dotted green curve in Figure 28), which guarantees safe speed values for the SAD. The rate of changes of the output of the controller (see the solid blue curve) is in general slightly smaller than that of

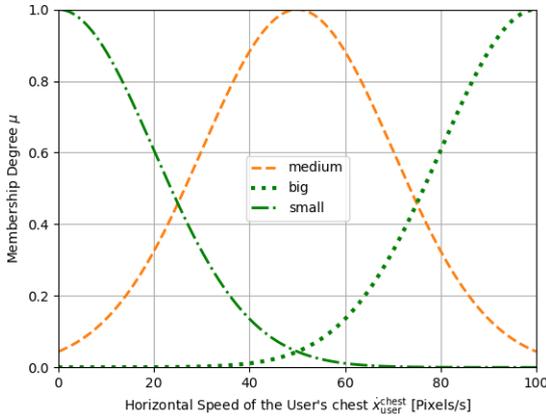


FIGURE 27. Fuzzy membership functions for the terms “small”, “medium”, and “large” used by the FL controller in passive game mode 1.

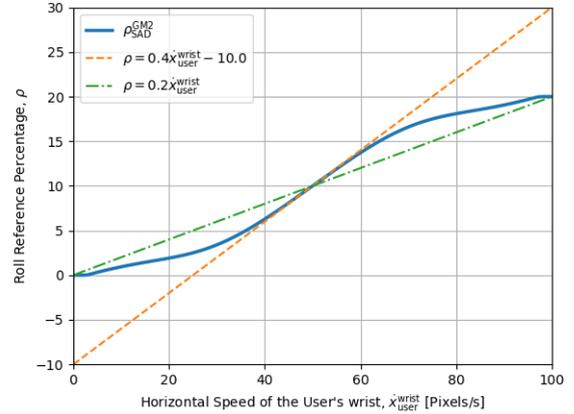


FIGURE 29. TSK-based FL controller steering the SAD to implement passive game mode 2: Input-output mapping for horizontal hand movements of participants.

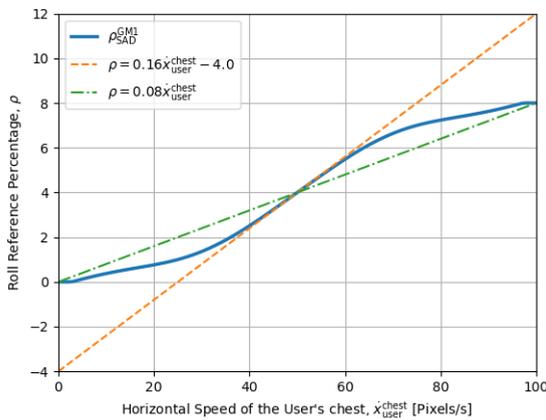


FIGURE 28. TSK-based FL controller steering the SAD to implement passive game mode 1: Input-output mapping for horizontal body movements of participants.

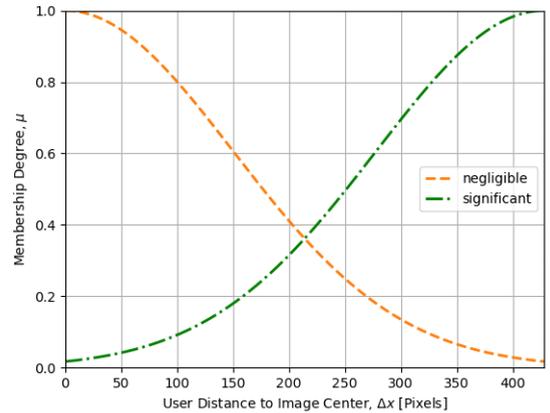


FIGURE 30. Membership functions for the terms “negligible” and “significant” in active game modes.

the dash-dotted green curve. This is to undermine the effect of unintentional movements of the user’s body observed in real-life experiments. Additionally, whenever the speed of the user’s chest has higher membership degrees corresponding to the set “medium” (see the dashed orange curve in Figure 27), the output of the FL controller resembles the behavior of a proportional controller with a slope of 0.16 s/pixels (see the dashed orange curve in Figure 28).

For the proportional controller given by (16), the maximum vertical speed of the user’s chest based on the experiments was 100 pixels/s, which correspond to a maximum value of 50% for ζ_{SAD}^{GM1} . The value of K^{GM1} was tuned to 0.5 s/pixels.

I. IMPLEMENTING THE FL CONTROLLER OF PASSIVE GAME MODE 2

The membership functions in (17) for the terms “small”, “medium”, and “large” were defined via the same fuzzy membership functions given in Figure 27. Since the horizontal displacements of the user’s chest were more significant than those associated with the user’s hands, the controller corresponding to game mode 2 was tuned such that the SAD

mimicked a scaled version of the motions of the user’s hands. This way the SAD stimulated a more engaging and entertaining interaction with the user. Consequently, the parameters of the controller given by (17) were tuned to $a_{12} = 0.2$ s/pixels, $a_{13} = 0.4$ s/pixels, $a_{14} = 0.2$ s/pixels, $b_{12} = 0$, $b_{13} = -10$, $b_{14} = 0$.

The resulting TSK-based FL controller for passive game mode 2 yields the input-output mapping shown in Figure 29. Comparing Figures 28 and 29, the roll displacements generated by the controller corresponding to passive game mode 2 are larger than those corresponding to the controller of passive game mode 1, which is due to the scaling factor in mimicking the motions of the user’s hands. For the proportional controller given by (18), the gain K^{GM2} was tuned to 0.5 s/pixels.

J. IMPLEMENTING THE FL CONTROLLER OF ACTIVE GAME MODE 3

The Gaussian fuzzy membership functions for the terms “negligible” and “significant” in (19) with standard deviations of 150 pixels are illustrated in Figure 30. Compared to

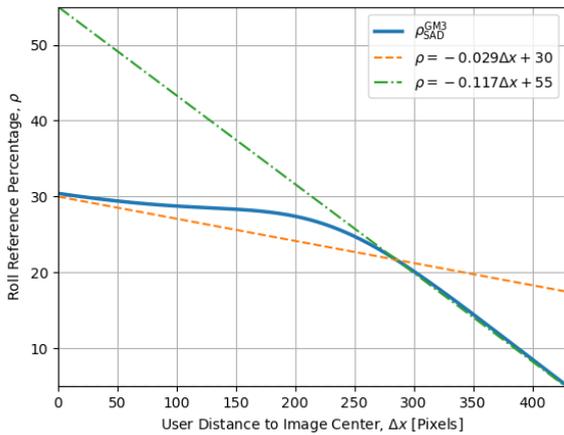


FIGURE 31. TSK-based FL controller steering the SAD to implement active game mode 3: Input-output mapping for horizontal movements of the SAD.

passive game modes where the user initiates the movements, in active game modes extra effort is needed to assure that the user remains engaged and follows the DMT plans. Therefore, the standard deviations of the membership functions for “negligible” and “significant” were adaptive. For instance, when users constantly experienced difficulty in positioning themselves in the center (according to the membership function defined for “negligible”) of the SAD’s frame of view and hence, receives low performance scores, the criteria for accepting Δx_{user} as negligible became more lenient via increasing the standard deviation of the corresponding membership function and by decreasing the standard deviation of the membership function for “significant”. The corresponding adaptive algorithm is further discussed in Section III-E. The tuned values for the consequent parameters of (19) include $a_{15} = -0.029$ and $a_{16} = -0.117$ (both in 1/pixels) and $b_{15} = 30$ and $b_{16} = 55$. These values guarantee that the distance travelled by the SAD in active game mode 3 is always safe and feasible (i.e., not larger than 3 m, which corresponds to a roll displacement percentage of 30%) and never becomes too small for the user to follow (i.e., not less than 0.5 m, which corresponds to a roll displacement percentage of 3%).

The input-output mapping of the resulting TSK-based FL controller is represented in Figure 31. For $\Delta x_{\text{user}} \leq 150$ pixels, the membership function corresponding to “negligible” (see the corresponding output in Figure 31 represented by the dashed orange curve) plays the major role in the output of the controller. For $\Delta x_{\text{user}} \geq 250$ pixels, however, the membership function corresponding to “significant” is dominant (see the corresponding output represented by the dash-dotted green curve in Figure 31). Moreover, whenever Δx_{user} is roughly centered, the output of the controller does not deviate significantly from 30% of the maximum roll displacement, which guarantees safety. This is reflected in the relatively smaller slope of the dashed orange curve in Figure 31. However,

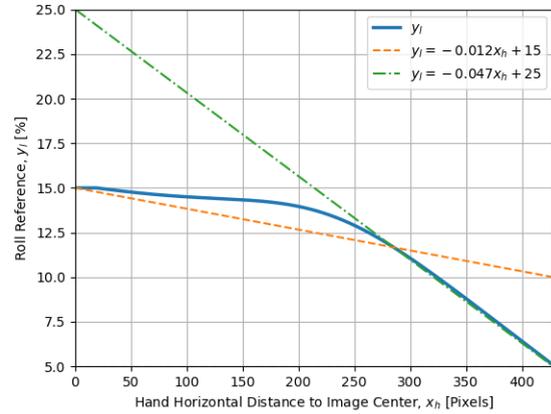


FIGURE 32. TSK-based FL controller steering the SAD to implement active game mode 4: Input-output mapping for horizontal movements of the SAD.

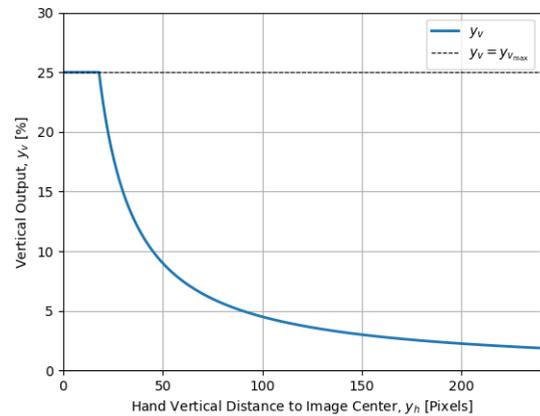


FIGURE 33. Proportional controller steering the SAD to implement active game mode 4: Input-output mapping for vertical movements of the SAD.

whenever Δx_{user} becomes more significant (which implies a worse performances for the user), the output of the controller varies more abruptly (see the solid blue curve in Figure 31) towards the minimum role displacement of 3% of the maximum roll rate.

K. IMPLEMENTING THE FL CONTROLLER OF ACTIVE GAME MODE 4

For active game mode 4, the same membership functions for the terms “negligible” and “significant” as those for game mode 3 were used (see Figure 30). Due to the larger vertical mobility of the hands compared to the chest, the proportional controller given by (22) was considered and tuned such that $a_{17} = -0.012$ 1/pixels, $a_{18} = -0.047$ 1/pixels, $b_{17} = 15$, $b_{18} = 25$, $K^{\text{GM4}} = 4.5$ pixels, and $\zeta_{\text{SAD}}^{\text{max}} = 25\%$.

The resulting TSK-based FL controller for the horizontal movements of the SAD yields the input-output mapping given in Figure 32. Moreover, Figure 33 represents the input-output mapping of the proportional controller that steers the vertical movements of the SAD for active game mode 4.

L. IMPLEMENTING THE ADAPTIVITY AND PERSONALIZATION MODULES

The values of the parameters for the adaptivity and personalization modules (see Sections III-E and III-F) are given in Table 2. The default values for the adaptivity and personalization parameters are given in Table 3.

V. RESULTS AND DISCUSSION

Tables 4-6 represent the results for the three time intervals considered per experiment (see Section IV-C) for the 10 participants, including the maximum, minimum, average value, and average change (in percentage) with respect to the default values of the personalization and adaptive parameters for the four game modes after being tuned.

Regarding the weights, the average change with respect to the default values was as high as about 60% (for passive game mode 1). Generally speaking, the rate of changes in the weights during the first time interval (i.e., almost the first 10 min of interactions) is less than the rate of changes in the subsequent time intervals. This is in line with the fact that participants gradually became more familiar and comfortable with the SAD and the setup of the experiments, and thus engaged more actively in the DMT plans resulting in a more significant personalization of the DMT plans. Moreover, the fact that higher variations in the weights are observed for passive game modes (i.e., when users are expected to initiate the movements) is because users felt more at ease after the first experiment and investigated more adventurous movements, implying that personalization was expected to become more significant by adjusting the weights in the subsequent time intervals. According to these results, the highest values of the tuned weights correspond to passive game mode 1. This matches the fact that all participants of these experiments were young and healthy people who found game mode 1 - which, in addition to demanding an active and initiative role from participants, requires the most physical activity amongst all the four game modes - the most joyful and engaging game mode.

Note that based on the results (see particularly the minimum values of the tuned weights in Tables 4 and 5s) the personalization algorithm sometimes converged too fast to the game modes preferred by users and thus gave other game modes little chance to engage the user (e.g., Table 5, shows that within almost 20 min of interaction, there is at least one participant per game mode for whom the weight of that game mode has reached its minimum). More specifically, for 80% of the participants the weight corresponding to at least one game mode has reached less than 0.1 and for 60% of them the weight has already reached the minimum value of 0.05 within the first 20 min. Thus, in order to avoid a premature convergence of the personalization algorithm and to give all game modes enough time to engage users, adjustments in the weight tuning algorithm are recommended by, e.g., reducing Δw in (33) and (34).

For the adaptivity module, we first consider the results for passive game modes 1 and 2. From Tables 4-6 the wait time t_w

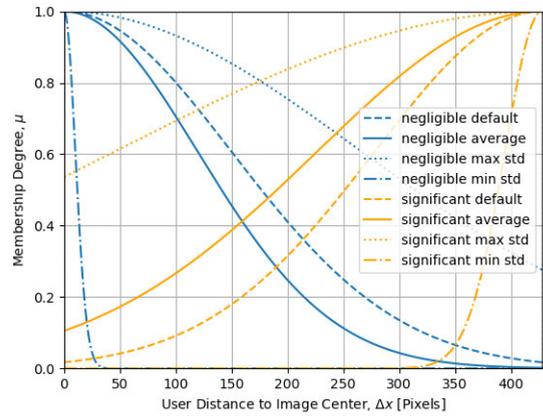


FIGURE 34. Variations generated by the adaptivity module in the membership functions “negligible” and “significant” corresponding to active game mode 3.

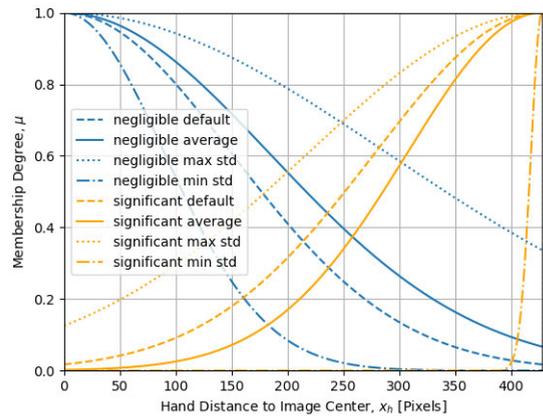


FIGURE 35. Variations generated by the adaptivity module in the membership functions “negligible” and “significant” corresponding to active game mode 4.

has evolved continuously during the experiments. Given the high diversity of the tuned values corresponding to different participants in different experiments, t_w has successfully been adapted for each user in the course of different conditions. Furthermore, passive game mode 1 corresponds to higher values of t_w compared to passive game mode 2, which is because whole body movements are slower and hence need a larger wait time than hand movements. For the number N_{trial} of trials within one episode, considering the high engagement of the participants in the two passive game modes, we expect N_{trial} to increase. The results shown in Tables 4-6 confirm this, where by the end of the third time interval N_{trial} has reached an average value of 3.8 for passive game mode 1 (where for 80% of the participants, N_{trial} has reached its upper bound, i.e., 4) and an average value of 4.8 for passive game mode 2 (where for 90% of the participants, N_{trial} has reached its upper bound, i.e., 5).

For active game modes, we first discuss the variations in the membership functions for “negligible” and “significant” (i.e., variations in σ_{neg} and σ_{sig}). Figures 34 and 35 illustrate

the corresponding Gaussian membership functions used initially (see the dashed curves), as well as the Gaussian membership functions corresponding to the minimum (see the dash-dotted curves), maximum (see the dotted curves), and average (see the solid curves) values of σ_{neg} and σ_{sig} after being tuned by the end of the experiments for active game modes 3 and 4. For active game mode 3, the maximum values for the standard deviations for “negligible” and “significant” are, respectively, 155% and 78% higher than their default values (i.e., 150). The minimum values of σ_{neg} and σ_{sig} for active game mode 3 are, respectively, 93.3% and 77.8% lower than their default values. The average values of σ_{neg} and σ_{sig} for active game mode 3 after being tuned are 119.7 and 201.7. Given the discussions in Section III-E2, a good performance observed from users corresponds to a decrease in the standard deviation for the term “negligible” and an increase in the standard deviation for the term “significant”. Taking into account that the proposed DMT plans in the experiments were not complicated for the participants and thus the majority of the participants exhibited a good performance, we expect to see the described effect in σ_{neg} and σ_{sig} , which is confirmed by the average value of the tuned parameters.

For active game mode 3, similarly to active game mode 2, the high diversity in the maximum, minimum, and average values of σ_{neg} and σ_{sig} after being tuned for various participants implies that these values have been estimated according to the variations in the experiments and users. In particular, for active game mode 4, the minimum values of the tuned parameters σ_{neg} and σ_{sig} show, respectively, a 40% and a 93% decrease with respect to their default values and the maximum tuned values are 93% and 40% larger than the default values for, respectively, σ_{neg} and σ_{sig} . The main difference compared to active game mode 3, is that the average value σ_{neg} has increased for 22.7%, while the average value of σ_{sig} has decreased for 19.3%, which imply a worse performance compared to active game mode 3.

Next, we discuss the changes made by the adaptivity module to the input-output mappings of the controllers corresponding to active game modes 3 and 4, where the default mappings and the average mappings after the parameters of the controllers were tuned are illustrated in Figures 36-38. On average, for active game mode 3, the adaptivity module yielded an entirely different input-output mapping, whereas for active game mode 4 the resulting input-output mapping showed a similar (but fine-tuned) behavior as the original input-output mapping. As a result of tuning the parameters of the controller for active game mode 3, a poor mimicking receives a higher reward, which corresponds to displacements with a larger amplitude (see the solid orange curve in Figure 37). The reason is that this game mode was too easy for nearly all participants, and thus their performance was almost always perceived as good by the adaptivity module, resulting in a fast convergence to the thresholds considered for the tuning parameters. Additionally, Table 6 shows that 9 out of 10 participants managed to minimize the parameter τ_c

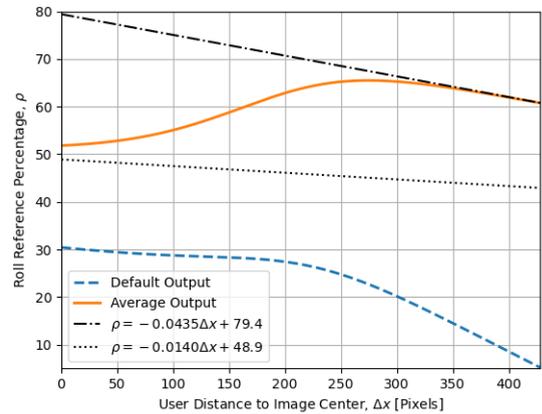


FIGURE 36. Input-output (default and average) mappings for the horizontal motion controller corresponding to active game mode 3.

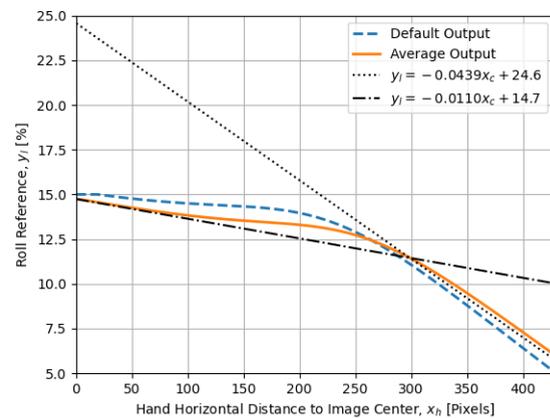


FIGURE 37. Input-output (default and average) mappings for the horizontal motion controller corresponding to active game mode 4.

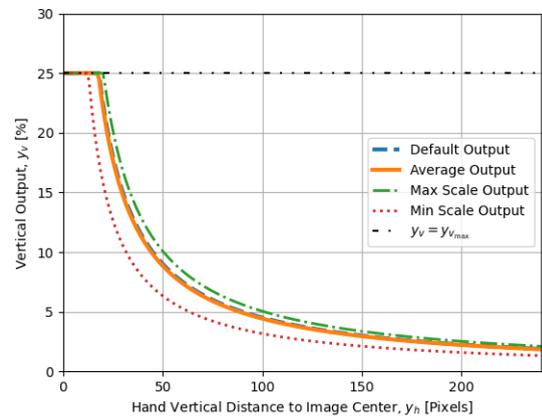


FIGURE 38. Input-output (default and average) mappings for the vertical motion controller corresponding to active game mode 4.

to its lowest bound 50 (which is an indication of an excellent performance by the users).

Active game mode 4, on the contrary, was not considered to be overly simplistic for the participants. The values obtained for τ_c (see Table 6) at the end of the third time interval remained close to the default value of 100. Moreover, for none

of the users the tuned value of τ_c reached its minimum value of 50. These results are consistent with the findings regarding the tuned membership functions for “negligible” and “significant” discussed earlier in this section implying that game mode 4 was relatively more challenging for participants. Similarly, the distribution of the tuned values of t_w for active game mode 3 (see Table 6) imply that, for the selected participants, this game mode was not very challenging, where the average value of t_w was around 5 s, with 80% of the participants being able to minimize the value of t_w to the lower bound (i.e., 4 s) for this parameter. For active game mode 4, however, a much wider range of values were observed for t_w .

VI. CONCLUSION AND TOPICS FOR FUTURE WORK

We introduced a socially assistive drone (SAD) that can autonomously interact with humans via planning and executing personalized dance movement therapy (DMT). In particular, we developed an image-processing-based data analysis module and a control system, which follows adaptive and fuzzy-logic-based decision making approaches. The proposed control system is capable of adapting the SAD’s decisions according to the changes in the environment and behavior of users. Moreover, the interactions are planned by the SAD in a personalized way according to the previous performance and engagement level of users. The developed approaches were successfully implemented via a Parrot Bebop 2 drone in real-life experiments with a sample of 10 random participants. The results of the experiments proved the ability of the SAD in both adaptation and personalization of the interactions and in engaging the users in DMT.

Increasing the number and variety of the participants (in terms of motor skill abilities and age and health conditions, particularly participants with ASD), is one of the next research steps. In order to avoid restricting the movements of the users (which is currently sometimes needed in passive game mode 2), the computational efficiency and frequency of performance of the IPU should be improved. Moreover, the IPU should be trained to detect physical barriers such as walls, which is crucial especially in smaller indoor spaces. Additionally, the IPU currently works with only one person present in the images. In the future, the IPU should be improved to deal with identifying more people (e.g., therapists or caregivers) in DMT sessions. Developing a music analysis module for the SAD in order to analyze music pieces in real time and decide accordingly about the speed and frequency of the proposed movements in active game modes is another topic for future research. Finally, designing and constructing a custom-made drone for DMT is an interesting topic for future research.

DATA AVAILABILITY

The data points for generating the figures (corresponding to the results) of this article are available via <https://doi.org/10.4121/16917940.v1>. The raw data of the experiments is available upon request. Note that only anonymized data can be shared. The corresponding author

Anahita Jamshidnejad may be contacted for requesting to access the data.

APPENDIX A EQUATIONS OF MOTION FOR A QUADCOPTER

In this appendix, approximate relationships are derived that explain how a specific roll angle ϕ or pitch angle θ results in a, respectively, displacement Δx or displacement Δz . Note that the dynamics of a quadcopter is usually derived according to two frames: the inertial frame, which is fixed to the ground, and the body frame, which is attached to the body of the quadcopter and moves according to the quadcopter’s orientation. The inertial frame is represented by x , y , and z , whereas the body frame is represented by x_b , y_b , and z_b (see Figures 39 and 40). The orientation of the quadcopter’s body frame around the z axis is the roll angle ϕ (see Figure 39) and the orientation of the quadcopter’s body frame around the x axis is the pitch angle θ (see Figure 40).

The influential forces on the quadcopter for a displacement in the x and z direction are represented in, respectively, Figures 39 and 40, where v_x and v_z are the magnitude of the speed of the quadcopter with respect to the inertial frame in, respectively, the x and z directions, $F_{D,x}$ and $F_{D,z}$ are the linear drag forces, which appear in the opposite direction of

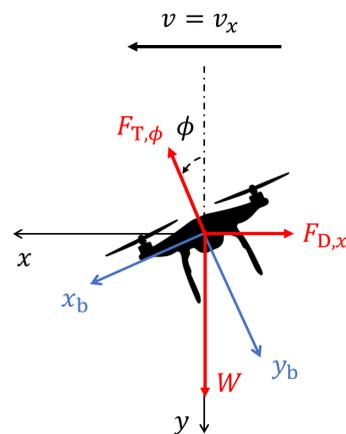


FIGURE 39. Force diagram for motion in the x direction.

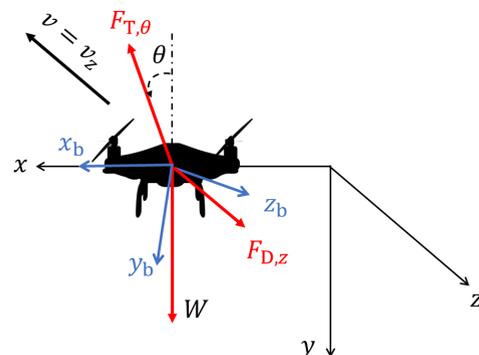


FIGURE 40. Force diagram for motion in the z direction.

the quadcopter’s motion, $F_{T,\phi}$ and $F_{T,\theta}$ are the thrust forces from the quadcopter’s propellers (perpendicular to the surface of propellers), and W is the quadcopter’s weight.

According to [33] the linear drag forces $F_{D,x}$ and $F_{D,z}$ are given by:

$$F_{D,x} = \frac{1}{2} C_{D,x} A_x \rho v_x^2 \quad (35)$$

$$F_{D,z} = \frac{1}{2} C_{D,z} A_z \rho v_z^2 \quad (36)$$

Where ρ is the air density, $C_{D,x}$ and $C_{D,z}$ are the drag coefficients in the x and z directions, and A_x and A_z represent the reference areas (cross sections considered perpendicular to the x axis and z axis, respectively). Based on (35) and (36), the linear drag force increases when the quadcopter speeds up. Therefore, it is assumed that an equilibrium state is achieved in the quadcopter’s direction of motion in a short time after the quadcopter tilts. Note that in real-life experiments, small angular displacements are implemented, meaning that the thrust force in the direction of motion of the quadcopter is very small. Therefore, a quasi-instantaneous equilibrium is a good estimation of the reality. After an equilibrium state is achieved, for the forces that result in a motion in the x direction (see Figure 39) we have:

$$\sum F_x = 0 \Rightarrow F_{T,\phi} \sin(\phi) = F_{D,x} \quad (37)$$

$$\sum F_y = 0 \Rightarrow F_{T,\phi} \cos(\phi) = W \Leftrightarrow F_{T,\phi} = \frac{W}{\cos(\phi)} \quad (38)$$

Based on (35), (37), and (38) we have:

$$W \tan(\phi) = \frac{C_{D,x} A_x \rho v_x^{eq2}}{2} \Leftrightarrow v_x^{eq} = \sqrt{\frac{2W}{C_{D,x} A_x \rho} \cdot \tan(\phi)} \quad (39)$$

with v_x^{eq} the magnitude of the equilibrium speed of the quadcopter in the x direction. Similarly, for the quadcopter’s motion in the z direction (see Figure 40), we obtain:

$$v_z^{eq} = \sqrt{\frac{2W}{C_{D,z} A_z \rho} \cdot \tan(\theta)} \quad (40)$$

with v_z^{eq} the magnitude of the equilibrium speed of the quadcopter in the z direction. These equilibrium speeds can be rewritten as:

$$v_x^{eq} = K_x^{quad} \sqrt{\tan(\phi)}, \quad \text{and} \quad v_z^{eq} = K_z^{quad} \sqrt{\tan(\theta)} \quad (41)$$

Where we have defined the constants K_x^{quad} and K_z^{quad} are defined by:

$$K_x^{quad} = \sqrt{\frac{2W}{C_{D,x} A_x \rho}}, \quad K_z^{quad} = \sqrt{\frac{2W}{C_{D,z} A_z \rho}} \quad (42)$$

Since the roll and pitch displacements of the SAD in all situations in the DMT game modes are relatively small (i.e., they do not exceed 10 degrees), the small angle approximation according to the first degree Taylor series expansion can be

used on the tangent function (i.e., $\tan x \approx x$), yielding the following expressions for the equilibrium speeds:

$$v_x^{eq} = K_x^{quad} \sqrt{\tan(\phi)}, \quad v_z^{eq} = K_z^{quad} \sqrt{\tan(\theta)} \quad (43)$$

Consequently, the corresponding linear displacements for a given angular displacement are formulated by:

$$\Delta x = v_x^{eq} t_{roll} \Leftrightarrow \Delta x = K_x^{quad} \sqrt{\phi} \cdot t_{\theta} \quad (44)$$

$$\Delta z = v_z^{eq} t_{\phi} \Leftrightarrow \Delta z = K_z^{quad} \sqrt{\theta} \cdot t_{pitch} \quad (45)$$

Where t_{θ} and t_{ϕ} are the time the quadcopter is asked to make the corresponding tilt and move in either the x or z direction.

Equations (44) and (45) explain why by steering the corresponding angular displacements of the SAD according to properly tuned control policies, desired linear movements in the x and z direction are achieved.

ACKNOWLEDGMENT

The authors thank Dr. R. Van der Hallen, Clinical Psychologist at Erasmus University Rotterdam, for providing input on expected therapeutic outcomes and their assessment in dance movement therapy for clients with autism spectrum disorders. Moreover, they also thank Dr. A. Sokolowska for training the artificial neural networks for the image processing unit. Finally, they would like to thank Dr. R. Samaritter for informative meetings and discussions about dance movement therapy.

REFERENCES

- [1] D. J. Ricks and M. B. Colton, “Trends and considerations in robot-assisted autism therapy,” in *Proc. IEEE Int. Conf. Robot. Automat.*, Anchorage, AK, USA, May 2010, pp. 4354–4359.
- [2] S. Kopp, E. Beckung, and C. Gillberg, “Developmental coordination disorder and other motor control problems in girls with autism spectrum disorder and/or attention-deficit/hyperactivity disorder,” *Res. Develop. Disabilities*, vol. 31, no. 2, pp. 350–361, Mar. 2010.
- [3] H. Dadgar, J. A. Rad, Z. Soleymani, A. Khorrami, J. McCleery, and S. Maroufizadeh, “The relationship between motor, imitation, and early social communication skills in children with autism,” *Iranian J. Psychiatry*, vol. 12, no. 4, p. 236, 2017.
- [4] M. A. Dziuk, J. C. G. Larson, A. Apostu, E. M. Mahone, M. B. Denckla, and S. H. Mostofsky, “Dyspraxia in autism: Association with motor, social, and communicative deficits,” *Develop. Med. Child Neurol.*, vol. 49, no. 10, pp. 734–739, Oct. 2007.
- [5] S. Goldman, C. Wang, M. W. Salgado, P. E. Greene, M. Kim, and I. Rapin, “Motor stereotypies in children with autism and other developmental disorders,” *Develop. Med. Child Neurol.*, vol. 51, no. 1, pp. 30–38, Jan. 2009.
- [6] T. Higashionna, R. Iwanaga, A. Tokunaga, A. Nakai, K. Tanaka, H. Nakane, and G. Tanaka, “Relationship between motor coordination, cognitive abilities, and academic achievement in Japanese children with neurodevelopmental disorders,” *Hong Kong J. Occupational Therapy*, vol. 30, no. 1, pp. 49–55, Dec. 2017.
- [7] A. Cummins, J. P. Piek, and M. J. Dyck, “Motor coordination, empathy, and social behaviour in school-aged children,” *Develop. Med. Child Neurol.*, vol. 47, no. 7, pp. 437–442, Feb. 2007.
- [8] M. Martin, “Moving on the spectrum: Dance/movement therapy as a potential early intervention tool for children with autism spectrum disorders,” *Arts Psychotherapy*, vol. 41, no. 5, pp. 545–553, Nov. 2014.
- [9] H. Takahashi, K. Matsushima, and T. Kato, “The effectiveness of dance/movement therapy interventions for autism spectrum disorder: A systematic review,” *Amer. J. Dance Therapy*, vol. 41, no. 1, pp. 55–74, Jun. 2019.

- [10] P. Marti, M. Bacigalupo, L. Giusti, C. Mennecozi, and T. Shibata, "Socially assistive robotics in the treatment of behavioural and psychological symptoms of dementia," in *Proc. 1st IEEE/RAS-EMBS Int. Conf. Biomed. Robot. Biomechatronics (BioRob)*, Pisa, Italy, Jul. 2006, pp. 483–488.
- [11] S. Shamsuddin, H. Yusof, L. Ismail, F. A. Hanapiah, S. Mohamed, H. A. Piah, and N. I. Zahari, "Initial response of autistic children in human-robot interaction therapy with humanoid robot NAO," in *Proc. IEEE 8th Int. Colloq. Signal Process. Appl.*, Malacca, Malaysia, Mar. 2012, pp. 188–193.
- [12] L. Boccanfuso and J. M. O'Kane, "CHARLIE: An adaptive robot design with hand and face tracking for use in autism therapy," *Int. J. Social Robot.*, vol. 3, no. 4, pp. 337–347, Nov. 2011.
- [13] T. Fong, I. Nourbakhsh, and K. Dautenhahn, "A survey of socially interactive robots," *Robot. Auton. Syst.*, vol. 42, nos. 3–4, pp. 143–166, Mar. 2003.
- [14] B. Scassellati, M. Matarić, and H. Admoni, "Robots for use in autism research," *Annu. Rev. Biomed. Eng.*, vol. 14, nos. 3–4, pp. 275–294, 2012.
- [15] A. Taheri, M. Alemi, A. Meghdari, H. PourEtemad, and N. M. Basiri, "Social robots as assistants for autism therapy in Iran: Research in progress," in *Proc. RSII/ISM Int. Conf. Robot. Mechatronics (ICRoM)*, Tehran, Iran, Oct. 2014, pp. 760–766.
- [16] J.-J. Cabibihan, H. Javed, M. Ang, and S. M. Aljunied, "Why robots? A survey on the roles and benefits of social robots in the therapy of children with autism," *Int. J. Social Robot.*, vol. 5, no. 4, pp. 593–618, Nov. 2013.
- [17] J. Kajopoulos, A. H. Y. Wong, A. W. C. Yuen, T. A. Dung, T. Y. Kee, and A. Wykowska, "Robot-assisted training of joint attention skills in children diagnosed with autism," in *Proc. Int. Conf. Social Robot.*, 2015, pp. 296–305.
- [18] L. I. Ismail, T. Verhoeven, J. Dambre, and F. Wyffels, "Leveraging robotics research for children with autism: A review," *Int. J. Social Robot.*, vol. 11, no. 3, pp. 389–410, Jun. 2019.
- [19] C. Moro, G. Nejat, and A. Mihailidis, "Learning and personalizing socially assistive robot behaviors to aid with activities of daily living," *ACM Trans. Hum.-Robot Interact.*, vol. 7, no. 2, pp. 1–25, Oct. 2018.
- [20] N. Giullian, D. Ricks, A. Atherton, M. Colton, M. Goodrich, and B. Brinton, "Detailed requirements for robots in autism therapy," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Istanbul, Turkey, Oct. 2010, pp. 2595–2602.
- [21] J. Astrouski. *Group Works With Autistic Children to Build, Fly Small Drones*. Accessed: Oct. 16, 2021. [Online]. Available: <https://wisconsinlife.org/story/group-works-with-autistic-children-to-build-fly-small-drones/>
- [22] L. Perez. *Cultivating Employment Skills With Drones*. Accessed: Oct. 16, 2021. [Online]. Available: <https://trajectorymagazine.com/tatts-cultivating-social-and-employment-skills-with-drones/>
- [23] E. H. Mamdani, "Advances in the linguistic synthesis of fuzzy controllers," *Int. J. Man-Mach. Stud.*, vol. 8, no. 6, pp. 669–678, 1976.
- [24] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965.
- [25] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan./Feb. 1985.
- [26] Z. Cao, G. Hidalgo, T. Simon, S. E. Wei, and Y. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 172–186, Jan. 2021.
- [27] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1145–1153.
- [28] A. McGovern. *Pyparrot's Documentation*. Accessed: Oct. 16, 2021. [Online]. Available: <https://pyparrot.readthedocs.io/>
- [29] P. Developers. *Olympe*. Accessed: Oct. 16, 2021. [Online]. Available: <https://github.com/Parrot-Developers/olympe>
- [30] A. McGovern. *Pyparrot*. Accessed: Oct. 16, 2021. [Online]. Available: <https://github.com/amymcgovern/pyparrot>
- [31] J. Rada-Vilela. *The FuzzyLite Libraries for Fuzzy Logic Control*. Accessed: Oct. 16, 2021. [Online]. Available: <https://fuzzyLite.com/info@cocodataset.org>
- [32] info@cocodataset.org. *COCO: Common Objects in Context*. Accessed: Oct. 16, 2021. [Online]. Available: <https://cocodataset.org/>
- [33] D. Gheorghiuță, I. Vintu, L. Mirea, and C. Brăescu, "Quadcopter control system," in *Proc. 19th Int. Conf. Syst. Theory, Control Comput. (ICSTCC)*, Cheile Gradistei, Romania, Oct. 2015, pp. 421–426.



TOMÁS ASCENSÃO received the B.Sc. degree in aerospace engineering from the Instituto Superior Técnico, Portugal, in 2019, and the M.Sc. degree in control and operations from the Aerospace Engineering Faculty, Delft University of Technology, The Netherlands, in 2022. He is currently working as a GNC/AOCS Control Engineer at GMV, Lisbon. His research interests include spacecraft attitude control, autonomous control, adaptive control, and fuzzy-logic-based control.



ANAHITA JAMSHIDNEJAD received the Ph.D. degree (*cum laude*) from the Delft University of Technology, The Netherlands, in 2017. She is currently an Assistant Professor of mathematical decision making at the Department of Control and Operations. She is the Founder and a Co-Director of the AI*MAN Laboratory at the Delft University of Technology. Her main research interests include optimization theory in engineering problems, AI-based control, integrated control methods, and

real-time model predictive control, with applications in autonomous and cognitive socially assistive robots.

• • •