This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101092889, Topic HORIZON-CL4-2022-HUMAN-01-14

SHARESPACE

Embodied Social Experiences in Hybrid Shared Spaces

Project Reference No	**101092889**
Deliverable	D5.1 Architecture design and validation (L1 and L2 avatars)
Workpackage	WP5: Cognitive Architecture
Nature	D (Deliverable)
Dissemination Level	PU - Public
Date	11/07/2023
Status	Draft v1.0
Editor(s)	Francesco De Lellis (CRdC) Marco Coraggio (CRdC) Antonio Grotta (CRdC) Mario di Bernardo (CRdC) Marta Bienkiewicz (UM) Nathan Foster (UKE)
Involved Institutions	CRdC, UKE, DFKI, UM, Golaem
Document Description	This deliverable reports the first design of the Cognitive Architectures to be used to drive L1 and L2 virtual humans in the SHARESPACE project. We introduce algorithms and strategies to drive the motion of nonautonomous (L1) and semiautonomous (L2) virtual humans. The architectures proposed are first described and then validated numerically considering the specificities of the proof-of-principle scenarios.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Table 1: List of Abbreviations

Term / Abbreviation	Definition
AI	Artificial Intelligence
CA	Cognitive Architecture
NN	Neural Network
PoP	Proof of Principle
RL	Reinforcement Learning
VH	Virtual Human

1 INTRODUCTION

1.1 PURPOSE OF THE DOCUMENT

The present deliverable describes the first set of algorithms and strategies implementing the *Cognitive Architecture* (CA) that drives the behavior of the L1 and L2 avatars (or virtual humans as defined in the living glossary, [deliverable D1.1]) in the SHARESPACE platform. In what follows, we provide a description of the algorithms, datasets, and feedback control strategies that were developed within the first 9 months of the project to achieve the first version of the CA for L1 and L2 virtual humans. Moreover, we present a thorough numerical validation of the proposed architectures and verify their compatibility with the software and hardware constraints of the other components of the SHARESPACE platform as described in D1.7.

1.2 STRUCTURE OF THE DOCUMENT

The remainder of this document is organized as follows:

- **Section 2: State of the Art in Human-AI Interaction in VR Spaces:** Here, we provide an overview of the current state-of-the-art in AI-driven techniques for simulating human interactions within virtual reality environments.
- **Section 3: The SHARESPACE Vision - Autonomy Levels:** In this section, we introduce the concept of autonomy levels within virtual humans, categorizing them into L1 and L2. We describe L1 as entirely human-driven virtual humans and L2 as semi-autonomous virtual humans capable of adjusting and adapting their actions in response to their human counterparts.
- **Section 4: L1 Cognitive Architecture:** This section briefly describes the tasks of the cognitive architecture when dealing with human-driven virtual humans, which are mostly related to compensating possible delays or loss of communication.
- **Section 5: L2 Cognitive Architecture:** Here, we delve into the potential integration of kinematic coding into AI-based algorithms to drive L2 semi-autonomous avatars, preserving the essential characteristics of a human user's intentions and movements. Additionally, we present the validation of this proposed architecture using PoP 1, as outlined in D1.2, as a representative case study.
- **Section 6: Conclusions and Future Works:** In the final section, we draw conclusions based on the preceding discussions and outline the directions for future research and development.

1.3 METHODOLOGY AND APPROACH

This first version of the CA has been designed to satisfy the requirements of the *Proof-of-Principles* (PoPs) presented in the deliverable D1.2 and to allow compatibility and integration with the *System Architecture* described in the deliverable D1.7. Both design and development of the CA involved discussion with partners of the consortium operating in other relevant work packages (i.e., WP1: "System Architecture", WP2: "Sensorimotor primitives of social interaction", WP3: "Capturing", WP4: "Rendering"; WP6: "System Integration, Real-World Scenarios and Validation"). Furthermore, we report preliminary numerical results concerning the PoP on Kinematic Chinese Whispers to analyze all the possible advantages and disadvantages of the different techniques required by the CA when combined with all other elements of the SHARESPACE platform.

2 STATE OF THE ART, HUMAN-AI INTERACTION IN VR SPACES

In the current literature, there are some examples of algorithms which were used to build cognitive architectures to drive avatars in motor interaction tasks with humans. A robust paradigm to study sensorimotor interaction is the *mirror game*, introduced in the seminal work by (Noy et al., 2011). In this game, a person is performing a movement that one or more participants must replicate as closely as possible in a dyadic interaction exercise. Autonomous cognitive architectures were proposed to make an avatar play the mirror game with a human. For example, in (Zhai et al., 2016), the authors consider the problem of achieving a dyadic interaction between a human player and a virtual one. The strategy adopted is optimal control which minimizes the mismatch between the positions of the two players. The controller design is based on the use of a Haken-Kelso-Bunz model (Haken et al., 1985) that, via a feedback control strategy, generates the oscillatory motion needed in the task. The same problem is solved in (Zhai et al., 2018) via an optimal adaptive control strategy. In these papers, a model describing accurately the motion of the human players is assumed to be known, and both the models and the controllers are deterministic. To overcome these limitations, in (Lombardi et al., 2021a) a cognitive architecture was proposed based on reinforcement learning. In particular, the virtual player decides its motion to mimic a certain person, through a (tabular) Q-learning algorithm, trained by observing the movement of that person playing the mirror game with another human. The problem of making a virtual human perform motor coordination tasks involving multiple agents were only seldom studied and some preliminary results were reported in (Lombardi et al., 2019, 2021b), which extend the work presented in (Lombardi et al., 2021a). In these papers, a Deep Q-learning algorithm is adopted to drive virtual human behaviour. The architecture is trained using a combination of data from real experiments and a sim2real paradigm where synthetic data is generated via appropriate models of multiple agents playing the mirror game.

Despite these advances, the integration of social factors into AI-based Cognitive Architectures remains a significant challenge, despite its profound implications for understanding human behavior within group tasks. Conversely, the wealth of data pertaining to human interactions during collective activities offers a promising avenue for effectively training autonomous agents in complex and socially oriented tasks. For instance, Babajanyan et al. (2022) conducted a comprehensive analysis of human participants engaged in a "pick and place" task. Their research illuminates how hidden intentions can be discerned from the emergent behavioral patterns when individuals collaborate to achieve a shared objective. In a similar vein, Calabrese et al. (2021) delved into the realm of social effects, demonstrating how subtle variations in body kinematics can be categorized into three distinct patterns for accurate leadership identification and classification. These patterns were meticulously scrutinized and confirmed using human movements observed during collective tasks aimed at achieving motor synchronization within groups. Furthermore, Scaliti et al. (2023) revealed the feasibility of discerning human intentions from recorded kinematic data when individuals are tasked with completing specific assignments.

These findings suggest that human behavior can be classified and clustered to understand intention in advance under specific circumstances. These aspects play a relevant role in building *human-aware* AI cognitive architectures to engage in complex group synchronization tasks and participate in the process of achieving a collective goal in human tasks. In what follows, we show how some of these findings can be combined with state-of-the-art machine learning techniques to effectively engage in human-driven contexts in a controlled and quantifiable way.

3 THE SHARESPACE VISION: AUTONOMY LEVELS

In SHARESPACE, the notation L0, L1, L2, quantifies the level of autonomy of a virtual human.[1] In particular, we recall the definitions given in deliverable D1.1:

- L0: A real human in a physical space.
- L1 VH: A virtual human who replicates the movement of a human, with the possibility of minor processing of its motion signals (e.g., noise filtering, compensation of time delays, and losses of data packets).
- L2 VH: A virtual human whose movements are a modified version of those of an associated human participant (e.g., change of amplitude and speed); the alteration is performed to achieve a specific goal while retaining resemblance to the human's original motion.

Moreover, we list the inputs, outputs, and main functions of the L1 and L2 CAs in the high-level description in Figure 1 and the block diagrams in Figure 4 and Figure 10. The inputs and the outputs of the CAs have been discussed and agreed with other partners of the Consortium, namely those operating in work packages WP1 ("System Architecture"), WP3 ("Capturing"), and WP4 ("Rendering"). From Figure 1, it is possible to see that the L1 CA only requires data from the associated participant in the real space, whereas the L2 CA could use data from all VHs in the hybrid space: as a matter of fact, information on the behavior of other VHs is necessary to establish how to alter the motion of the participant represented by L2, to achieve a goal that normally depends on the motion of the whole group. One of the requirements of the L2 CA is that the altered motion is not too different from that of the relative participant. To accomplish this, it is possible to enforce a constraint on the distance between the motion of the participant and that of its L2 counterpart.

Table 2: High-level description of the cognitive architectures for different levels of autonomy of virtual humans.

LA	Goal	Inputs	Outputs
L1	Manipulate minimally the input motion so that the output motion replicates the input while compensating for digital alterations such as delays and noise	Motion of a participant in real space	Motion of corresponding VH in hybrid space
L2	Modify the input motion so that the output motion achieves better values of one or more metrics while still being similar to the input	• Motion of a participant in the real space • Motions of other VHs in hybrid space • Metrics to assess the compliance of the input and output motion with a goal • Movement primitives to guide the synthesis of realistic motion	Motion of corresponding VH in hybrid space

[1] Additionally, L3 is the level of autonomy of a fully autonomous VH, to be discussed in deliverable 5.2.

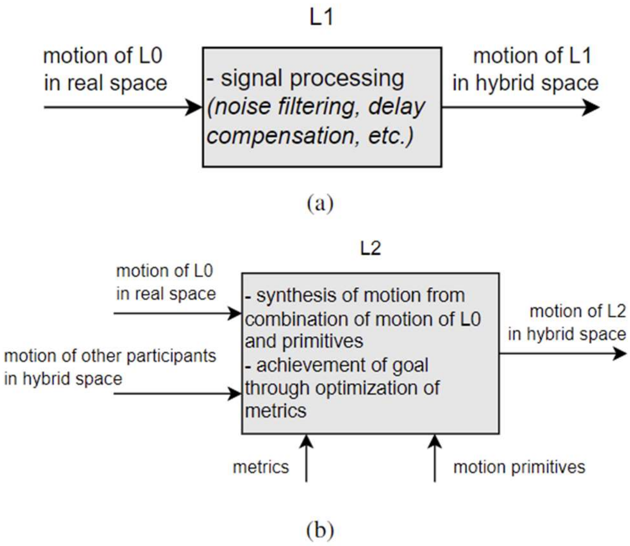

Figure 1: Abstract schematics of the input/output and tasks for the cognitive architectures for different levels of autonomy of virtual humans (VH). (a) and (b) portray the schemes for L1 and L2 cognitive architectures, respectively.

3.1 Assumptions of the Cognitive Architecture

Next, we list the assumptions that were used in the design of the L1 and L2 CAs.

A1 Motions that are input and output to the CA are multidimensional discrete-time signals, with $3d$ dimensions over a fixed time-span $T \in R_{>0}$, where $d \in N_{\geq 1}$ is a number of points in space (e.g., an arm position, a finger position).

A2 For each task and associated goal, at least one metric (i.e., a function that produces a real number) is available to assess how good a certain motion is with respect to the goal.

A3 For each task, a *movement primitive library* exists: the library is a set of paradigmatic motions, called *movement primitives*. The library is exhaustive concerning the kinds of motion that can be performed in a given task, in the sense that there are no motions that can reasonably be performed in the task and are not close (according to some distance function) to the motions in the library. The derivation of such libraries is currently part of WP2: "Sensorimotor primitives of social interaction".

A4 L2 cognitive architectures alter only a single degree of freedom of the skeleton representation of a participant (e.g., the end-effector position, such as the position of a fingertip or of a wrist).

The input signals mentioned in Assumption A1 are stored in a circular buffer, which acquires new data points as the participant/VHs move, contextually deleting old data points. In an L2 CA, the motion of the relative participant is analyzed through the principles and methodology under development in the WP2 work package ("Sensorimotor primitives of social interaction"); the results of that analysis inform the interpretation of the participant's motion as performed by the CA. Moreover, Assumptions A2 and A3 are required to carry out the training process of the Cognitive Architecture of L2 virtual humans. In particular, the movement primitives' library and appropriate metrics are used by the CA to identify what modifications of the human movements are possible in a task and decide which ones to perform and how. As a preliminary case of study we assumed that the degrees of freedom to be manipulated by the L2 CA is low as the exact number of degrees of freedom to be involved in *kinematic coding* (definition 5.11.4 of deliverable D1.1) of the motion in the tasks relevant to the SHARESPACE project will be determined through the pilot experiments on the PoP scenarios that are currently being carried out. Findings concerning this aspect will help inform whether future implementations of the CA retain Assumption A4.

4 L1 COGNITIVE ARCHITECTURE

Within the SHARESPACE project, L1 virtual humans are tasked with replicating user movements accurately. To ensure a faithful representation of the virtual human, the L1 Cognitive Architecture (CA) may need to engage in data processing as the information streams in.

Specifically, the L1 CA must take into account the technical specifications of the sensors employed and the typical transmission time delays, as detailed in WP3: "Capturing" within deliverable D1.7. The current technology for capturing body kinematics involves the use of inertial motion units (IMUs), which are strategically placed to capture the movements of most joints in the human body. To address potential signal drift issues associated with IMUs (as documented by Ahmad et al., 2013), the acquired data will be fused with data from cameras. Consequently, at the current developmental stage, the L1 CA does not necessitate the incorporation of signal compensation mechanisms from the sensors.

Another critical consideration pertains to the data streaming platform under development in task 5.3 of the project. Depending on the characteristics of the streaming protocol, there may be instances of delay compensation operations.

At the current stage of development the L1 CA architecture will replicate as outputs the movements of the human driving it as they are, with minimal filtering to reduce noise levels if needed.

5 L2 COGNITIVE ARCHITECTURE

As previously mentioned in Section 3, L2 virtual humans can make autonomous decisions to modify the movement of a participant to improve user experience during group tasks and amplify/attenuate user intentions encoded in their movement. In particular, the L2 cognitive architecture has to satisfy three crucial requirements:

- The L2 VH's movements need to be as close as possible to the original participant's motion.
- L2 VH's movements need to express intention and information coherent with those of the participant, possibly amplifying/attenuating the information encoded in the input kinematics.
- The L2 CA needs to process incoming data and generate the VH's motion in real time, thus minimizing computation time.

These requirements introduce a multi-objective challenge that the L2 cognitive architecture must grapple with. In resolving this challenge, the L2 architecture must delicately navigate a tripartite equilibrium: firstly, facilitating effective information flow from the participant to the L2 virtual human (VH); secondly, enacting meaningful modifications to the original motion; and thirdly, optimizing computational efficiency. The actual configuration of the Cognitive Architecture (CA) hinges significantly on the specific group task at hand. Consequently, to elucidate the implementation of the L2 CA, outline its structure, and elucidate the training and deployment phases, we turn to the Proof of Principles, as presented in deliverable D1.2, which serve as benchmark problems and focus on PoP 1 to illustrate the capabilities and performance of the current L2 CA design.

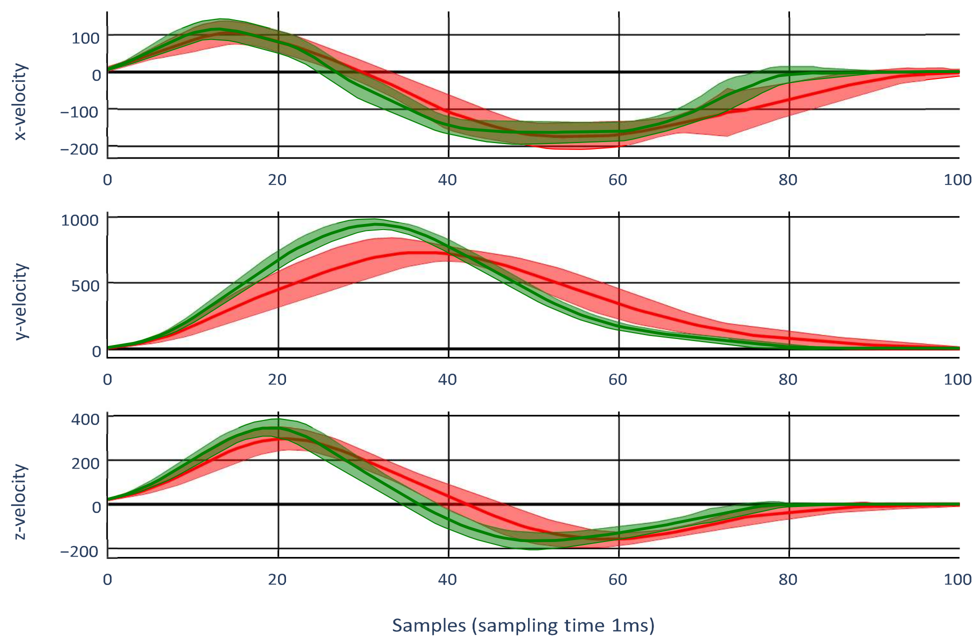

Figure 2: Speed profiles projection on x, y, and z planes of a test user performing reach to grasp movement primitive with fear(red) and without fear(green). Measurements refer to the average wrist marker in multiple sessions. The solid lines represent the average, while the shaded areas correspond to two times the standard deviation.

5.1 L2 CA FOR THE POP OF AMPLIFICATION (KINEMATIC CHINESE WHISPERS)

In this Proof of Principle, the participants are asked to pass an object between each other to study how information propagates through body kinematics. An in-depth description of this Proof of Principle is provided in deliverable D1.2. However, for the scope of this document, we consider the case where some of the virtual humans in the chain pass the object and transmit fear through *kinematic coding* (definition 5.11.4 of deliverable D1.1). In this context, the objective of the L2 CA is to alter the movement of a given participant to minimize fear transmission. In the future, the L2 CA will be used to both minimize and maximize information transmission.

Currently, the working assumption is that *kinematic coding* is performed through specific speed profiles of the user's wrist; thus, we limit the analysis to the act of picking an object, referred to as *reach to grasp* motion. In later stages of the SHARESPACE project, a more comprehensive description of the key sensorimotor primitives (cf. deliverable D1.2) required for the kinematic coding of this PoP will be provided: this will inform the design of the next implementation of the CA, which might also consider additional motion signals, as a result.

An example of speed profiles for wrist motion, captured with a VICON motion capture system, is reported in Figure 2. These preliminary data show that there is a visible difference in the speed profile of individuals who feared the object and others who did not. Similar evidence is discussed in depth in (Turri et al., 2022; Scaliti et al., 2023), where the authors show how it is possible to predict human intention through kinematic encoding and readout.

A set of measured motion signals provided by the team in Hamburg, as the ones described in Figure 2, is used to populate a *movement primitives' library*, which serves as a database of examples of speed profiles that *kinematically encode* fear and speed profiles that do not. In general, the library required by the L2 CA is depicted in Figure 3 and is intended to be a collection of signals that define the movement primitives.

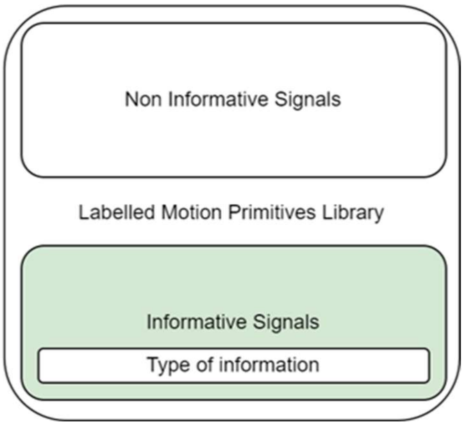

Figure 3: Block scheme of the movement primitives library.

In such a collection of examples, each sample is a set of signals (specified by the *kinematic coding*) that are labeled to identify their properties. In particular, the labeling is required to specify:

- the type of motion (e.g., reach to grasp, pass the object, return to rest).
- the presence of socially relevant information (e.g., fear of an object).

As the library has to contain movement primitives that cover all possible human movements in the task (see Assumption A3), the next iterations of the CA will employ a wider spectrum of motion signals, to be acquired in WP2 ("Sensorimotor primitives of social interaction").

In the next sections, we will describe the components of the CA when it is deployed in an application (Section 5.1.1) and the offline training phase of the algorithms involved (Section 5.1.2).

5.1.1 Implementation of the Cognitive Architecture

The main blocks making up the L2 CA in this PoP and its integration with the System Architecture (described in the deliverable D1.7) are represented graphically in Figure 4. Namely, the L2 CA has to communicate with the platform to obtain positional and velocity data of the participant performing the movements that need to be altered.

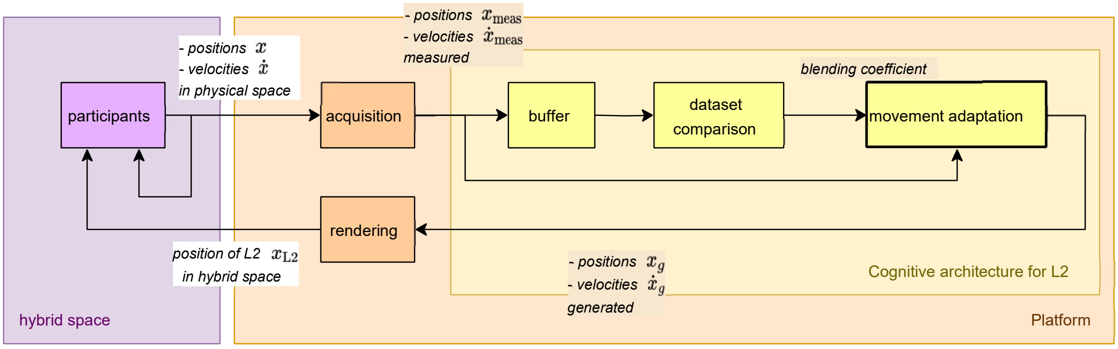

Figure 4: Block scheme of the proposed L2 cognitive architecture and its integration with other components being under development by other work packages (WP3: "Capturing", WP4: "Rendering")).

In what follows, we describe the key elements of the L2 CA. To do so, we refer to the representative application of the *reach to grasp* phase. In the *reach to grasp* phase, participants are asked to pick an object so that they can pass it to the next person sitting by their side. Some of the participants have learned to fear picking up the object, as in previous iterations mild electric shocks were administered to them when they touched the object. Hence, the movement of these participants will *kinematically encode* fear information. The objective of an L2 virtual human is to alter the motion of these participants (L0) so that the modified motion does not exhibit fear. In this task, the movement primitives library contains both speed profiles that kinematically encode fear and others that do not.

The key blocks of the CA as depicted in Figure 4 have the following characteristics.

"Buffer" block: Data acquired by the human participant connected to her/his L2 VH passes through a buffer (a temporary memory), which is used to store short pieces of the movement being performed. The buffer has a fixed length and is updated with a FIFO policy.

"Dataset comparison" block: This block searches in the movement primitives library the movement primitive that is the closest to the motion stored in the buffer and yields it as output.

"Movement adaptation" block: This block applies the amplification of movement primitives encoding social information. This processing is applied to the data stream as long as the previous *dataset comparison* block outputs the same movement primitive from the library. In this way, the L2 CA treats the user's movement in the same way as the sample in the library selected as the closest to the user execution itself.

As the CA has to act in real-time, we preemptively train the L2 architecture to remove the fear information from motion signals in the library by combining them with other signals in the library that do not carry fear information. This process is further clarified in the next Section 5.1.2.

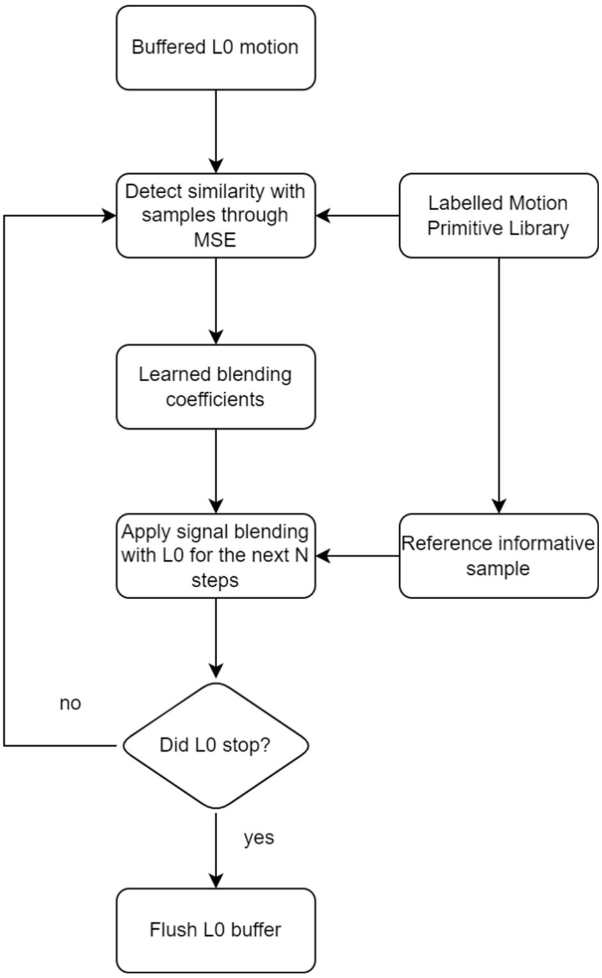

Figure 5: Flowchart describing the deployment of an L2 cognitive architecture for the Proof-of-Principle of amplification: Kinematic Chinese Whispers.

The deployment of the L2 CA follows the rationale depicted in Figure 5. Initially, the motion signal in the buffer is compared with the samples in the movement primitives library, with the aim to find the movement primitive that is the closest, say m_{mpl} , to the buffered signal. The comparison is performed by minimizing the mean square error between the data points of the two signals; for the operation to be meaningful, these two signals must be defined at the same time instants. Moreover, the operation is possible because the buffered signal and the movement primitives are short segments of human movements. Next, the cognitive architecture alters the streamed signal of the user movements say, m_{L0} , as though it was m_{mpl}. In particular, the alteration is made by blending m_{L0} with another signal in the movement primitives library, say $m_{mpl}^{no\,fear}$, that contains the target level of information (e.g. absence of fear). In particular, the blending is carried out via a *convex combination* between m_{L0} and $m_{mpl}^{no\,fear}$, with the blending coefficient being the value α^* that is used to blend m_{mpl} and $m_{mpl}^{no\,fear}$ to obtain a signal without fear. The value α^* is learned (for all combinations of m_{mpl} and $m_{mpl}^{no\,fear}$) during the training phase of the architecture. The Cognitive Architecture then stops if the human participant stops and waits for them to perform a new motion, meaning that the L2 CA is always active on the lookout for the intentions of the user through their movement.

To reduce the computational burden, each search for the closest movement primitive in the library is triggered on a different temporal frame, which is a multiple of the sampling time.

5.1.2 Training of the Cognitive Architecture

The L2 CA uses a set of Supervised Learning techniques to acquire information from data generated by human participants. In particular, the CA is required to perform two key operations:

- detect if target information is present in each recorded movement (e.g., fear).
- learn a transformation between any movement signal that *kinematically encodes* certain information into another that does not, and vice versa.

These two steps are the essential core of the L2 CA, in particular, to address the first point, we train a Neural Network (NN), that we call the *Intersection Information Classifier*, to discriminate whether a certain movement primitive contains fear information; the network is trained on the data contained in the *movement primitives library*. It is to be noted that this training requires a considerable amount of data, and the motion samples in the movement primitives library (used for this training) need to be labeled with the type of information kinematically encoded in the movement. Hence, after training is completed, the Intersection Information Classifier can recognize if any given movement primitive kinematically encodes the presence of fear.

The second important step is to devise a method to learn a table of blending coefficients that map two signals into a new one that contains a target level of some information (e.g., minimizing fear), which is specified before training. In Section 5.1.3, we show how to perform this training.

Even if the training of these two elements is carried out in two separate stages, they interact closely in the fine-tuning of the architecture as depicted in Figure 6. The training process starts by selecting a sample from the movement primitives library which kinematically encodes undesired information (e.g., fear). Then, the CA performs a convex combination of the selected motion sample with a second motion sample extracted from the library. The latter movement primitive is a sample that kinematically encodes the target information (e.g., absence of fear). The objective of the training is to find the blending coefficient used to perform the convex combination to find a balance between the following goals:

- the resulting sample is classified as a sample that kinematically encodes the target information.
- the resulting sample is the closest to the motion which kinematically encodes the undesired information.

The main intuition here is that the CA needs to learn a transformation that generates a new sample that kinematically encodes the target information. At the same time, the CA is required to perform a minimal modification of the sample. This is a key step as the L2 CA must minimize its motion alteration to reduce the impact on the user’s movements.

The learning process of the blending coefficient is made by performing a search in an array of values between 0 and 1 discretized in a finite number of steps. This process is carried out for all the samples contained in the movement primitives library. Therefore, as a result of the search, the L2 CA obtains a table that contains the blending coefficients to morph any sample of the library into a new movement primitive that kinematically encodes the target information using a reference movement primitive.

5.1.3 Numerical Validation

In this section, we analyze the properties of the proposed L2 CA considering the task of *reach to grasp*. In this example, we analyze a single type of movement primitive to understand if the CA can strike a balance between manipulating the information kinematically encoded in the movement and keeping the movement similar to that executed by the user.

Movement primitive: To train the architecture, we use a preliminary movement primitives library containing the samples shown in Figure 2. In this case, the library is made of a single movement primitive, that is the *reach to grasp* motion, and contains many executions of such primitive classified based on the encoding of fear. The collection of the signals is labeled to note whether they kinematically encode fear or not, resulting in a movement primitives library as depicted in Figure 3. In particular, we use 17 samples that contain fear information and 14 samples that do not. For the sake of clarity, in numerical experiments, when we refer to a sample, we mean the 3D speed measurement of the wrist marker of the full execution of the primitive *reach to grasp*.

Intersection Information Classifier: Using the signals contained in the movement primitives library as training dataset, we train a NN to assess whether a given sample contains fear. The inputs to the NN are the speed signals that make a movement primitive. Therefore, the structure of the NN uses a single hidden layer of two times the maximum signal length recorded, with a linear activation function, and with a single output layer having a sigmoid activation function. The output of such network is a binary value, which is 1 if the input sample (i.e., the movement primitive) kinematically encodes fear, and 0 otherwise. Due to the small number of preliminary samples, we use the entire library for training and validate the outcome with the synthetic data generated by the L2 itself.

Architecture Training: With the movement primitives library and the intersection information classifier, we proceed to train the architecture. This stage is carried out by following the flow diagram of Figure 6.

The training process begins by choosing a *reference informative sample*, say *s1,* that guides the CA in the process of generating an amplified movement. At this stage, as s_1, we select a sample that does not kinematically encode fear, from the movement primitives library. The role of s_1 is to participate in the generation process of a new motion which inherits the characteristics contained in the reference informative sample.

Then, the algorithm selects a second sample, say s_2, that kinematically encodes fear from the movement primitives library and performs a convex combination between s_1 and s_2, generating a new sample, say s_3. In particular, the architecture generates 500 instances of s_3, by *blending* the s_1 and s_2 500 times, each time with a difference *blending coefficient*, with values in the range [0, 1]; 0 corresponds to the $s_3 = s_1$, while 1 corresponds to $s_3 = s_2$. All the 500 instances of possible s_3 obtained from this process are then fed to the trained *Intersection Information Classifier* to check which of them does not kinematically encode fear. From this set of samples, the one that has the highest blending coefficients (that is, the closest to s_2) is stored in an array.

By doing so, the architecture learns a table of *blending coefficients* which achieves the following:

- the altered version (s_3) of the fear profile (s_2) is classified as a sample that does not kinematically encode fear by the *Intersection Information Classifier*.
- the convex combination keeps at minimum the weight of the *reference informative sample* (s_1), therefore maximizing the value of the blending coefficient.

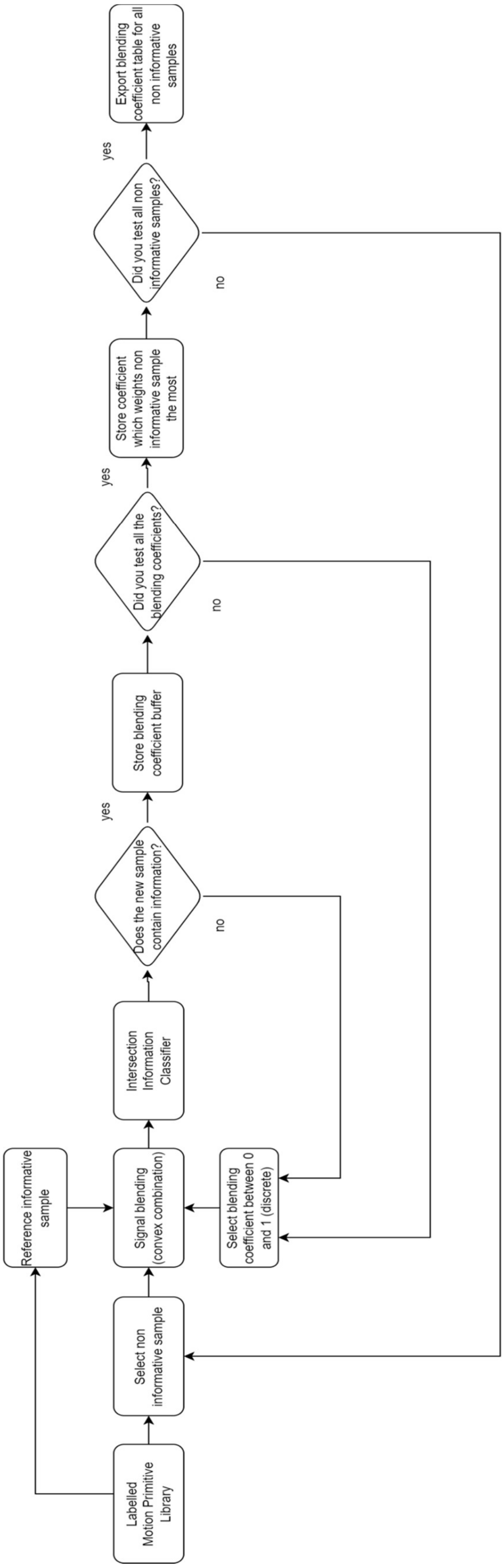

Figure 6: Flowchart describing the training of an L2 cognitive architecture for the Proof-of-Principle of amplification: Kinematic Chinese Whispers.

By repeating this process, we obtain an array containing the blending coefficients to be used to hide fear from the samples of the movement primitives library when a convex combination is applied with the *reference informative signal*.

To further investigate the effects of the architecture, we also repeat the training process by using all the samples of the library that do not kinematically encode fear as *reference informative samples*. In Figure 7, we report the results of the training as a matrix of values obtained for the 17 samples that *kinematically encode* fear combined with the 14 available *reference informative samples*.

The values of the coefficients obtained show that there might be high variability in terms of how the architecture needs to weigh the references with the other samples. However, we also note that 89.9% of the combinations require a significative alteration by the L2 Cognitive Architecture which has learned to keep the blending coefficients below 0.5. This fact is an indication that the CA has to act so that the reference informative sample is weighted the most in the blending process. This fact could lead to users experiencing a visible difference in their movements. Therefore, this aspect requires further investigation during the deployment of the proposed L2 CA in the virtual space that is currently under development in other work packages (WP1 ("System Architecture"), WP3 ("Capturing"), and WP4 ("Rendering")). Moreover, this might be a symptom of a lack of a sufficient number of samples in the movement primitives library to train the *Intersection Information Classifier* in these preliminary experiments; further investigation will be carried out as more samples are available.

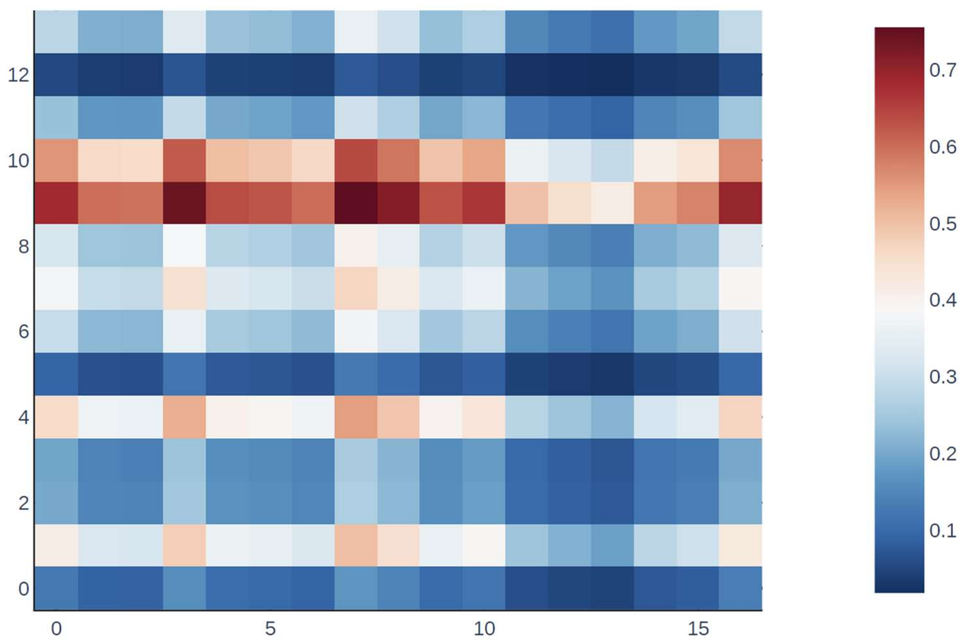

Figure 7: Values of blending coefficients associated with the samples that kinematically encode fear of the motion library for the reach to grasp phase. Each coefficient corresponds to the value learned during training to perform the convex combination with all the 17 samples which kinematically encode fear (on the x-axis) and the 14 reference informative samples (on the y-axis).

Architecture Deployment: After the training process, we obtain a set of coefficients that could be used in real-time to process the user's movement if it repeated the exact signals contained in the movement primitives library. However, most likely, the user will not replicate one of those samples exactly; hence the L2 CA needs to adapt the selection of a blending coefficient based on the information learned.

The main aspects of this adaptation mechanism follow the reasoning of the flow diagram in Figure 5. The key idea of the L2 CA is to analyze the movement of the user while they perform a movement expecting a precise movement primitive. In this case study, we replay a random sample taken from a movement primitives library that kinematically encodes fear to test if the architecture is capable of modifying it in real-time. Also, by limiting the movement primitives to be only the *reach to grasp*, we store in a buffer the wrist velocity measurements available as the L0 performs a movement. With this information, we also store the time of each point in the buffer relative to the start of the movement.

The samples contained in the buffer are then compared with the same time frame as the *reference informative sample*. This comparison is carried out by measuring the distance from the point in the buffer to select the sample closest to it from the ones present in the movement primitives library. This operation occurs on a different temporal scale to give the time of execution to the user represented by the replayed sample which kinematically encodes fear. This comparison aims to find the sample closest to the user execution. By doing so, the L2 CA applies the movement amplification as if the user were to perform the exact sample selected. This process becomes more and more accurate as the user performs the movement and fills the buffer. During numerical simulations, the comparison occurs every 5 times the sampling time.

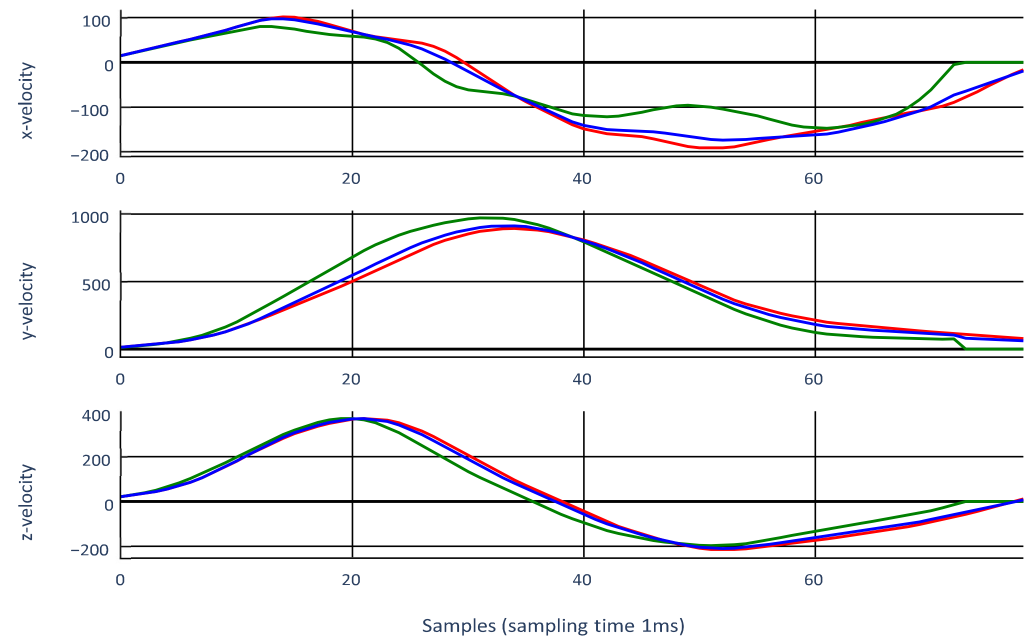

Figure 8: Speed signals referring to the wrist joint. To perform the numerical experiments the sample that does not kinematically encode fear (green solid line) and a sample that does kinematically encode fear (red solid line) are selected. The correspondent L2 virtual human speed signal is generated synthetically and is depicted in blue.

Discussion.

The outcome of this process is depicted in Figure 8 and Figure 9 where we replay the samples that required the most and the least manipulation by the L2 CA, respectively. Both the solutions obtained are classified by the *Intersection Information Classifier* as signals that do not kinematically encode fear, whereas the L0 samples do. In particular, the results of Figure 8 refer to the case where the L2 CA learns to use the highest values of the blending coefficient that most weight the L0 execution. This numerical simulation shows that the L2 CA is capable of minimally modifying the signals of the L0 to hide the fear information. Also, we note that the L2 CA acts as a smoother, especially on the x-axis, pointing out that the fear information might also be kinematically encoded into the acceleration signals of the human participants. Moreover, the results of Figure 8 refer to the opposite case where the L2 CA learns to use the highest values in the blending coefficient which weighs the *reference informative signal* the most. The result of the L2 virtual human simulation shows how the architecture can adapt the L0 movement regardless of their execution to hide the fear information. However, Figure 8 also shows that L2 CA might have a great impact on virtual human L2, which could, in principle, drive away the motion of the virtual human from that of the human participant.

The full experimental campaign has been carried out on all the possible combinations of reference motions and samples that kinematically encode fear. This preliminary study shows a rate of generation of speed signals that are classified as samples that do not kinematically encode fear 53.6% of the time according to the *Intersection Information Classifier*. Future implementations will take into account the possibility that information is kinematically encoded in more joints and signals (e.g., elbow acceleration, wrist jerk), a research question that will be investigated in coordination with WP2 ("Sensorimotor primitives of social interaction") and will use a larger movement primitives library to train the *Intersection Information Classifier*. The development of the movement primitives library represents a first step towards the building of the semi-autonomous L2 and fully autonomous L3 virtual humans which will be carefully analyzed in deliverable D5.4.

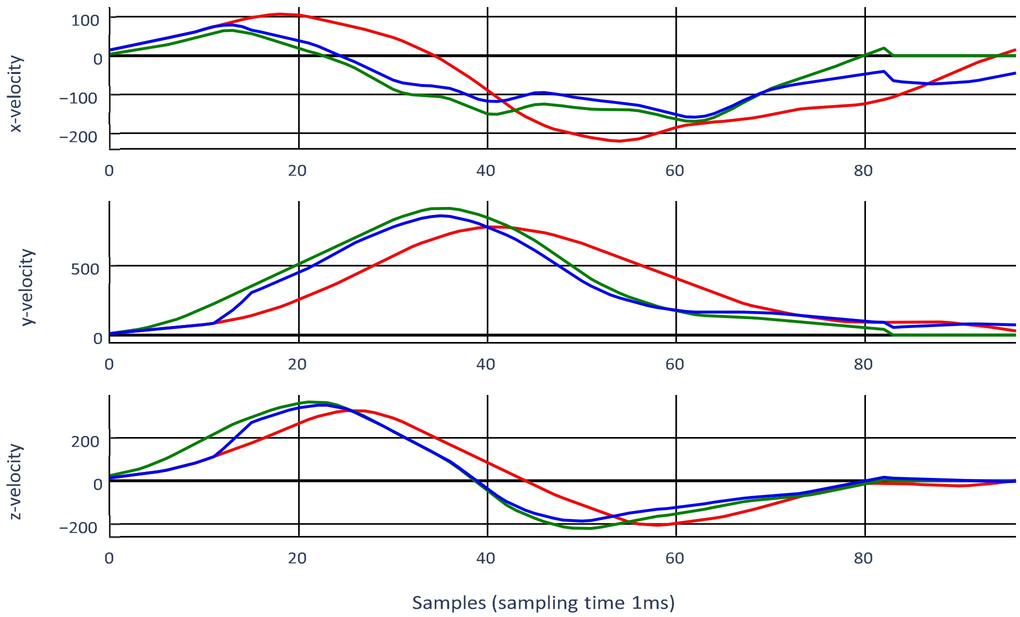

Figure 9: Speed signals referring to the wrist joint. To perform the numerical experiments the sample that does not kinematically encode fear (green solid line) and a sample that does kinematically encode fear (red solid line) are selected. The correspondent L2 virtual human speed signal is generated synthetically and is depicted in blue.

5.1.4 Computational Details

The numerical simulation has been carried out with Python. The training algorithms have been built using TensorFlow. The hardware used is a laptop equipped with an AMD Ryzen 9 6900HS processor, an NVIDIA GeForce RTX 3070 Ti graphic processor, and 16 GB of RAM DDR5. The dataset used has a sampling time T_s = 1ms and during deployment, the L2 CA compares the data stored in the buffer with the movement primitives library every 5 sampling times T_s. In this context, a single iteration can take up to about 2ms to perform the movement primitives library search and a RAM usage below 1GB.

However, these amount of times and storage are expected to increase when the library grows larger: this relation will be further investigated in the next implementations of the CA.

5.2 L2 CA FOR THE POP OF SOCIAL CONNECTEDNESS

In this PoP, all participants are asked to perform a synchronization task to provide evidence about the open diffusion of social information. Specifically, standing in a circle all facing each other, they are asked to move their arms in synchrony with each other (Alderisio et al., 2017). An in-depth description of this proof of principle is provided in deliverable D1.2. In this section, we will discuss a scenario based on a group of L1 virtual humans carrying out the same synchronization task described above (and more in depth in deliverable D1.2), assuming one of the participant is an L2 virtual human. Kinematics of this L2 virtual human will be manipulated by the CA with the goal of increasing the synchronization level, and presumably the social connectedness, of the group.

5.2.1 Mathematical model

In general, given a signal $x : \mathbb{R} \to \mathbb{R}$, it is possible to associate a phase $\theta^H \in [0, 2\pi]$ to it, according to the analytic signal theory (Pikovskij et al., 2003, A2.1). In particular, the phase $\theta^H(t)$ is defined as the phase of the complex signal $x(t) + \mathrm{i}\, x_H(t)$, where $x_H(t)$ is the Hilbert transform of $x(t)$. It is possible to verify that if $x(t)$ has a narrow frequency band, $\theta^H(t)$ approximates the phase of the dominant frequency in $x(t)$.

As done in (Alderisio et al., 2017), this theory can be used to associate a phase, say $\theta_i^H(t)$, to the motion along an axis, say $x_i(t)$, of the finger of participant i in the social connectedness PoP, described in Section 5.2. In (Alderisio et al., 2017), it was verified that the evolution of the phases $\theta_i^H(t)$ is well approximated by the dynamics of a network of heterogeneous Kuramoto oscillators described by the following equations:

$$\dot{\theta}_i(t) = \omega_i(t) + \frac{c}{N}\sum_{j\in\mathcal{N}_i} a_{ij} \sin\left(\theta_j(t) - \theta_i(t)\right), \ \forall i \in \{1, \dots, N\}, \tag{5.1}$$

where $\theta_i(t) \in [0,2\pi]$ approximates $\theta_i^H(t)$, $\omega_i(t) \in \mathbb{R}$ is the natural frequency of the i-th participant, c is a scalar coupling strength, N is the number of participants, $\mathcal{N}_i$ is the set of neighbors of participant i, a_{ij} is 1 if participant i is visually coupled with participant j, and 0 otherwise. The value of $\omega_i(t)$can be estimated by having participant i carry out the oscillatory task in isolation.

In this setting, the level of synchronization of the group is measured by the Kuramoto order parameter $r: \mathbb{R}^N \to [0,1]$, given by

$$r := \frac{1}{N}\sum_{i=1}^{N} \left|e^{j\theta_i}\right| \tag{5.2}$$

Values of r close to 1 and 0 are associated to high and low synchronization, respectively. The goal of the L2 CA is to enhance the coordination of the group, by maximizing r.

5.2.2 Cognitive architecture

A schematic of the L2 architecture drive the L2 virtual human in the Social Connectedness PoP is reported in Figure 10.

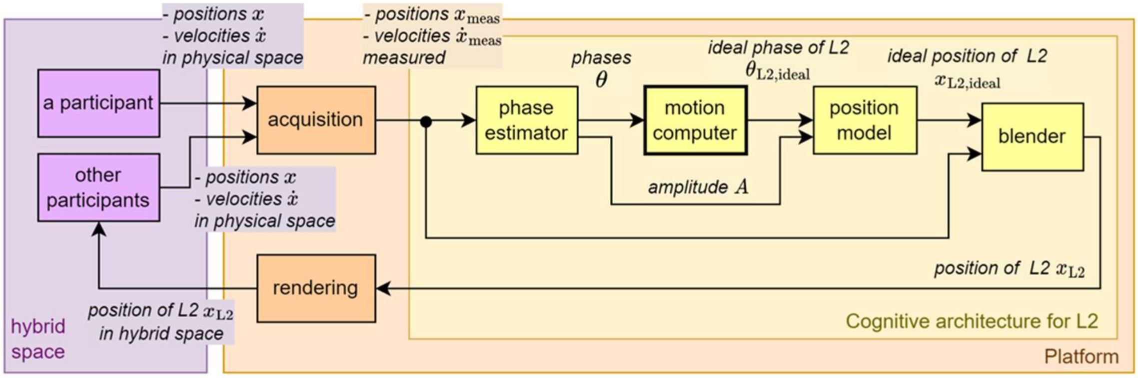

Figure 10: Block scheme of the proposed L2 cognitive architecture for the Social Connectedness and its integration with other components being under development by other work packages (WP3: "Capturing", WP4: "Rendering")).

The inputs to the architecture are the linear motions $x_{i,\mathrm{meas}}$ of the VHs/participants, acquired from the hybrid space, while the output is the linear motion of the L2 VH.

"Phase estimator": The "phase estimator" block takes as input the discretized motions $x_{i,\mathrm{meas}}(k)\forall i$ of the VHs/participants and outputs the estimated phases $\theta_i(k)$ of all VHs/participants and the estimated amplitudes, say $A(k)$, of their motion, according to Algorithm 1 (k is discrete time). The algorithm assumes that the signal $x(k)$ alternates one maximum, one crossing of zero, one minimum, one crossing of zero, and so on. If necessary, in future implementations of the CA, we might also consider extended versions of the algorithm that do not require this assumption.

Algorithm 1: Phase estimation from linear motion

Input: p_{t-1} (position at time $t-1$), p_t (position at time t), v_{t-1} (velocity at time $t-1$), v_t (velocity at time t), $A_{t-1}^{p\geq 0}$ (amplitude of position oscillation, when position is positive, at time $t-1$), $A_{t-1}^{p<0}$ (same, when position is negative), $A_{t-1}^{v\geq 0}$ (amplitude of velocity oscillation, when velocity is positive, at time $t-1$), $A_{t-1}^{v<0}$ (same, when velocity is negative).

Output: θ_t (estimated phase at time t), $A_t^{p\geq 0}$, $A_t^{p<0}$, $A_t^{v\geq 0}$, $A_t^{v<0}$.

if $p_{t-1} < 0$ *and* $p_t \geq 0$, **then** $A_t^{v\geq 0} = |v_t|$, **else** $A_t^{v\geq 0} = A_{t-1}^{v\geq 0}$;
if $p_{t-1} \geq 0$ *and* $p_t < 0$, **then** $A_t^{v<0} = |v_t|$, **else** $A_t^{v<0} = A_{t-1}^{v<0}$;
if $v_{t-1} \geq 0$ *and* $v_t < 0$, **then** $A_t^{p\geq 0} = |p_t|$, **else** $A_t^{p\geq 0} = A_{t-1}^{p\geq 0}$;
if $v_{t-1} < 0$ *and* $v_t \geq 0$, **then** $A_t^{p<0} = |p_t|$, **else** $A_t^{p<0} = A_{t-1}^{p<0}$;

if $p_t \geq 0$, **then** $p_t^{\mathrm{norm}} = p_t / A_t^{p\geq 0}$, **else** $p_t^{\mathrm{norm}} = p_t / A_t^{p<0}$;
if $v_t \geq 0$, **then** $v_t^{\mathrm{norm}} = v_t / A_t^{v\geq 0}$, **else** $v_t^{\mathrm{norm}} = v_t / A_t^{v<0}$;

$\theta_t = \mathrm{atan2}(-v_t^{\mathrm{norm}}, p_t^{\mathrm{norm}})$;

Algorithm 1: estimation of angular phases from linear motions

"Motion computer" block: The "motion computer" block takes as input the phases $\theta_i(k)$ of all VHs/participants and outputs an *ideal* phase of the L2 VH, namely $\theta_{i^*,\mathrm{ideal}}(k)$, where i^* is the index of the participant whose motion is going to be altered by the L2 CA; the quantity is called ideal in the sense that it will have to undergo further manipulations before it is passed to the rendering block. The block operates as follows. $\theta_{i^*,\mathrm{ideal}}(k)$ is generated using a modified version of the model (5.1), where the cognitive architecture acts on the natural frequency $\omega_{i^*}(k)$ of the L2 VH, which means we use the following equation to generate θ_{i^*}:

$$\theta_{i^*}(k+1) = \theta_{i^*}(k) + \Delta t\left(\omega_{i^*}(k) + \Delta\omega_{i^*}(k) + \frac{c}{N}\sum_{j\in\mathcal{N}_{i^*}} a_{i^*j}\sin\left(\theta_j(k) - \theta_{i^*}(k)\right)\right) \quad (5.3)$$

where $\Delta\omega_{i^*}(k)$ is the variation of the natural frequency of the human that drives the L2 architecture, k is discrete time, and Δt is the time step of a discretization of (5.1).

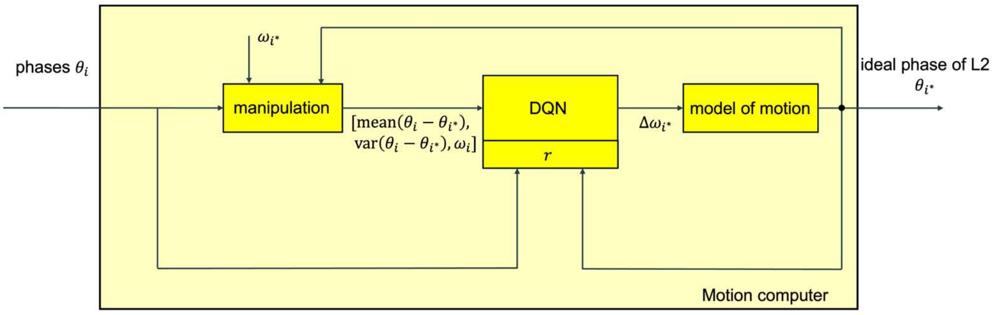

Figure 11: Schematic of the generation process of the ideal phase $\theta_{i^*,ideal}$

Figure 11 shows an in-depth description of the "motion computer" block, concerning the details of how the phase $\theta_{i^*,\text{ideal}}$ is generated. In particular, the estimated phases $\theta_{i^*}(k)$ and $\theta_i(k)$ and the natural frequency $\omega_{i^*}(k)$ are the inputs of the "manipulation block", which operates as an aggregator. Indeed, it outputs the array $[\text{mean}(\theta_i(k) - \theta_{i^*}(k)), \text{var}(\theta_i(k) - \theta_{i^*}(k)), \omega_{i^*}(k))]$, used as input to the "DQN" block. Specifically, we use the mean and variance to scale groups of different numbers of participants. In the "DQN" block, the Deep Q Network (DQN) is implemented, which is a well-known Reinforcement Learning (RL) algorithm (the details can be found in [(Mnih et al., 2015)]). At each time step k, the algorithm outputs the variation of the natural frequency $\Delta\omega_{i^*}$ that maximize the metric of interest, namely the order parameter r. $\Delta\omega_{i^*}$ will be used as input to the "model of motion" block. This block outputs the ideal phase θ_{i^*} using the equation in (5.1). Further detail on the training of this block is provided below in Section 5.2.3.

"Position model" block*:* The phase θ_{i^*} generated by the "motion computer" block is transformed into a linear motion $x_{i^*,\text{meas}}$ by the block "position model" with the transformation

$$x_{i^*}(k) = A(k)\cos(\theta_{i^*}(k)) \quad (5.4)$$

and then used as an input (with the data acquired from the hybrid space) to the "blending" block.

"Blender" block: This block takes as input $x_{i^*,\text{ideal}}$ and $x_{i^*,\text{meas}}$ (directly from the "acquisition" block) and outputs x_i*, which is a combination of the two inputs. x_{i^*} is chosen so as to ensure that x_{i^*} is never too different from $x_{i^*,\text{meas}}$, otherwise other participants might have the impression they are not interacting with participant i^*, but rather with an autonomous virtual human. The first implementation of this block we consider is a saturation function; namely,

$$x_{i^*} = x_{i^*,\text{meas}} + \text{sat}\left(x_{i^*,\text{ideal}} - x_{i^*,\text{meas}}\right) \quad (5.5)$$

where, given some bound $S^+ > 0$ and $S^- < 0$,

$$sat(\xi) = \begin{cases} S^+, & \xi > S^+ \\ \xi, & S^- \le \xi \le S^+ \\ S^-, & \xi < S^+ \end{cases} \quad (5.6)$$

5.2.3 Training of the architecture

To train the motion computer block in the proposed architecture, we would need a huge amount of data on how participants react to at least most of the possible actions of the L2 VH, in a variety of conditions. However, that would amount to possibly many hundredths or thousands of experiments, which is unfeasible in practice. Hence, we train the L2 VH by letting it interact with synthetic agents, modelling human behavior through the Kuramoto oscillators equation in (5.1). The fact that the input to the L2 VH (more precisely, the DQN block in Figure 11) is an aggregation of the phases of the VHs the L2 interacts with–that is, mean and variance–allows flexibility of the L2 VH; therefore, it suffices to train the L2 CA by having it interact with just two oscillators, rather than with any possible number of oscillators. Then, to present a wide variety of situations to the CA, in each episode of the training, we change the natural frequencies of the two agents, extracting them from a uniform distribution in the interval $\left[\boldsymbol{\omega_m} - \frac{\Delta\omega}{2}, \boldsymbol{\omega_m} + \frac{\Delta\omega}{2}\right]$, where $\boldsymbol{\omega_m}$ is the mean frequency of a group of people performing the oscillatory task in isolation, and $\boldsymbol{\Delta\omega} = \boldsymbol{c}$, where c is the coupling gain in (5.1) (in the literature the value of *c* has been taken equal to 1.25 [Alderisio et al., 2017]). The width of the interval is chosen so to allow the possibility of synchronization, according to the theory of synchronization of Kuramoto oscillators [Bullo, 2022, Theorem 17.9].

The numerical validation of the proposed approach is currently under investigation and will be reported in the next deliverable (D5.4) together with the design of the L3 CA.

6 Conclusions and Future Works

In this deliverable, we presented the first implementation of the AI-driven Cognitive Architectures (CAs) that decide the motion behavior of the virtual humans in the SHARESPACE project. After reviewing the state of the art and the autonomy levels of virtual humans in SHARESPACE in Sections 2 and 3, respectively, in Section 4, we discussed the processing that the CA needs to perform in order to drive L1 virtual humans. In particular, the CA will at this stage just compensate possible delays in the inputs and perform minimal filtering to reduce possible noise in the inputs. Next, in Section 5, we described in detail the design, the training, and the implementation of the L2 Cognitive Architecture. Namely, we provided two different implementations of the L2 CA, one to be used in tasks more similar to the PoP of Amplification and the other to be used in the PoP on Social Connectedness.

The L2 CA for the PoP of Amplification aims at altering the motion of a participant so that it (does/does not) encodes certain information (that can be read by human participants). The CA works by recording and altering portions of movement of a participant. In particular, the CA blends the movement to be altered with another in the motion primitive library, which contains the information that has to be added/removed to the original motion. The way in which the second movement is selected and the way in which the blending procedure is done was explained in detail in Section 5.1.3. Numerical validation provided evidence of the effectiveness of this approach in amplifying a movement to encode kinematic information.

The L2 CA for the PoP on Social Connectedness has the role of altering the movement of a participant with the aim to improve social coordination in the group. The implementation we proposed works by modifying in real time the frequency of the oscillatory motion of a participant, while keeping the variation contained, in order to avoid detachment from the original motion.

The next steps of the project concerning the development of the CA are the following:

For the L2 CA for the PoP of Amplification:

- Given a more comprehensive analysis of what information in the avatars kinematic (i.e., position/velocity of specific human body parts) needs to be encoded in their motion (as will be described in D2.1 and D2.4), we will implement the proposed CA and verify that it is still able to amplify all the required signals.
- We will validate numerically the CA with a larger movement primitive library to refine the design.
- The CA will then be validated in experiments that will be carried out in WP2 (D2.7).

For the L2 CA for the PoP on Social Connectedness:

- Compare the implementation presented here with others where the amplitude of the oscillatory motion of the participant is changed, and/or a lag/lead in phase is introduced.
- Perform more extensive numerical validation of the L2 CA.
  Validate the CA in experiments to be carried out on the PoP (D2.9).

7 REFERENCES

N. Ahmad, R. A. R. Ghazilla, N. M. Khairi, and V. Kasi, “Reviews on various inertial measurement unit (imu) sensor applications,” International Journal of Signal Processing Systems, vol. 1, no. 2, pp. 256–262, 2013.

F. Alderisio, G. Fiore, R. N. Salesse, B. G. Bardy, and M. di Bernardo, “Interaction patterns and individual dynamics shape the way we move in synchrony,” Scientific Reports, vol. 7, no. 1, 2017.

D. Babajanyan, G. Patil, M. Lamb, R. W. Kallen, and M. J. Richardson, “I know your next move: Action decisions in dyadic pick and place tasks,” in 44th Annual Meeting of the Cognitive Science Society: Cognitive Diversity, pp. 563–570, 2022.

F. Bullo, “Lectures on network systems”, ed. 1.6. Kindle Direct Publishing, 2022.

C. Calabrese, M. Lombardi, E. Bollt, P. De Lellis, B. G. Bardy, and M. Di Bernardo, “Spontaneous emergence of leadership patterns drives synchronization in complex human networks,” Scientific Reports, vol. 11, no. 1, p. 18379, 2021.

H. Haken, J. A. S. Kelso, and H. Bunz, “A theoretical model of phase transitions in human hand movements,” *Biol. Cybern.*, vol. 51, no. 5, pp. 347–356, 1985

M. Lombardi, D. Liuzza, and M. di Bernardo, “Deep learning control of artificial avatars in group coordination tasks,” in Proc. of 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), pp. 714–719, 2019.

M. Lombardi, D. Liuzza, and M. di Bernardo, “Dynamic input deep learning control of artificial avatars in a multi-agent joint motor task,” Frontiers in Robotics and AI, vol. 8, 2021.

M. Lombardi, D. Liuzza, and M. di Bernardo, “Using learning to control artificial avatars in human motor coordination tasks,” IEEE Transactions on Robotics, vol. 37, no. 6, pp. 2067–2082, 2021.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, “Human-level control through deep reinforcement learning,” Nature, vol. 518, no. 7540, pp. 529–533, 2015.

L. Noy, E. Dekel, and U. Alon, “The mirror game as a paradigm for studying the dynamics of two people improvising motion together,” Proceedings of the National Academy of Sciences, vol. 108, no. 52, pp. 20 947–20 952, 2011.

A. Pikovskij, M. Rosenblum, and J. Kurths, Synchronization: A Universal Concept in Nonlinear Sciences. Cambridge: Cambridge Univ. Press, 2003.

E. Scaliti, K. Pullar, G. Borghini, A. Cavallo, S. Panzeri, and C. Becchio, “Kinematic priming of action predictions,” Current Biology, 2023.

G. Turri, A. Cavallo, L. Romeo, M. Pontil, A. Sanfey, S. Panzeri, and C. Becchio, “Decoding social decisions from movement kinematics,” iScience, vol. 25, no. 12, p. 105550, 2022.

C. Zhai, F. Alderisio, P. Słowiński, K. Tsaneva-Atanasova, and M. di Bernardo, “Design of a virtual player for joint improvisation with humans in the mirror game,” PLOS ONE, vol. 11, no. 4, p. e0154361, 2016.

C. Zhai, F. Alderisio, P. Słowiński, K. Tsaneva-Atanasova, and M. di Bernardo, “Design and Validation of a Virtual Player for Studying Interpersonal Coordination in the Mirror Game,” IEEE Transactions on Cybernetics, vol. 48, no. 3, pp. 1018–1029, 2018.