*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 | D5.2 Architecture design and validation (L1 and L2 avatars) |
| Workpackage | WP5: Cognitive Architecture |
| Nature | D (Deliverable) |
| Dissemination Level | PU - Public |
| Date | 20/05/2024 |
| Status | Draft v1.0 |
| Editor(s) | Marco Coraggio (CRdC), Francesco De Lellis (CRdC), Antonio Spallone (CRdC), Antonio Grotta (CRdC), Mario di Bernardo (CRdC) |
| Involved Institutions | CRdC |
| Document Description | This deliverable reports the revised design of the cognitive architectures to be used to drive L1 and L2 virtual humans in the SHARESPACE project. The proposed architectures are first described and then validated 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 an updated version of the algorithms and strategies first presented in Deliverable D5.1 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. We present the designs of both the L1 and L2 CA, focusing on the two Proofs of Principle of the project. After the presentation of the methodology that we followed in the development, we present experimental and numerical validation of the proposed architectures.

## 1.2 STRUCTURE OF THE DOCUMENT

The remainder of this document is organized as follows:

- **Section 2: State of the art** — This section reviews the latest advancements and methodologies in terms of infusion and recognition of emotion in artificial movement, and of artificial movement in group motor tasks.
- **Section 3: Role and assumptions of L1, L2 cognitive architectures —** This part reviews the roles and underlying assumptions of the L1 and L2 CAs.
- **Section 4: L1 cognitive architecture: —** This section focuses on the L1 CA, detailing its design and operation.
- **Section 5: L2 cognitive architecture —** This part provides an in-depth look at the design and validation of the L2 CA, presenting designs valid for both Proofs of Principle.
- **Section 6: Phase estimation in 3D —** This section introduces a new methodology for phase estimation within three-dimensional contexts, relevant to L2 CAs exploiting the phase domain.
- **Section 7: Conclusions and future work**: In the final section, we draw conclusions based on the preceding discussions and outline the directions for future research and development.

# 2 STATE OF THE ART

This section updates the state of the art presented in Deliverable D5.1, pertaining to the analysis and synthesis of emotion and intention in motion, and on the use of cognitive architectures in group motor tasks to achieve specific goals.

Human societies are fundamentally based on group interactions (Homans, 1951). As we advance towards a future increasingly intertwined with virtual and augmented reality, people might be represented by digital avatars. These avatars are sometimes partially controlled by cognitive architectures (CAs) (Langley et al. 2009) (Laird et al., 2017), which are advanced control systems designed to enhance interaction with humans and other avatars.

A crucial characteristic of cognitive architectures is their ability to enable avatars to display the emotions and intentions necessary for effective interaction and collaboration. Several studies have explored the impact of emotions on motor tasks. Babajanyan et al. (2022) examined a *pick and place* task, revealing how hidden intentions can be discerned from emergent behavioral patterns during collaborative efforts to achieve a shared objective. Moreover, Calabrese et al. (2021) demonstrated

that subtle variations in body kinematics could be categorized into three distinct patterns for accurate leadership identification and classification. Scaliti et al. (2023) showed the feasibility of discerning human intentions from recorded kinematic data when individuals are tasked with specific assignments.

Moreover, several studies, such as Atkinson et al. (2004) and Llobera (2022), have shown that human observers can recognize emotions from body movements, suggesting the potential for communicating emotions through motion. This has led to the development of socially aware robotic systems that use kinematic redundancy to encode emotions in movements (Claret et al., 2017) and trajectory planning that incorporates emotional aspects, drawing on *Laban movement analysis* (LMA) (Lourens et al., 2010). For instance, Masuda and Kato (2010) used LMA to adapt arbitrary movements of humanoid robots to add target emotions, while Claret et al. (2017) provided an inverse kinematic strategy to convey specific emotions in humanoid robot movements exploiting kinematic redundancy. Additionally, virtual reality has been used to study emotion encoding in body kinematics, minimizing emotion misclassification in exergame scenarios (Lombardi et al., 2021a). A common limitation of existing approaches to generating motion encoding of emotions (e.g., Claret et al., 2017; Masuda and Kato, 2010) is that the presence of emotions is assessed only qualitatively.

In motor tasks, several cognitive architectures have been developed to enhance coordination [with possible application to sports (Neumann, 2018) and rehabilitation (Howard, 2017)]. The *mirror game* is a paradigmatic motor task where participants are instructed to execute motions that are both “synchronized and interesting” (Noy et al., 2011). For example, Zhai et al. (2014) combined the Haken-Kelso-Bunz (HKB) model (Haken et al., 1985) with a nonlinear feedback controller to enable an avatar to lead or follow a person's movements in the mirror game. Later, Zhai et al. (2016) identified a model describing the motions of two players and used it to design a cognitive architecture capable of improvising joint movements with a person. A shift towards model-free approaches was introduced by Lombardi et al. (2021b), who employed a Q-learning algorithm to enable an avatar to interact with a human playing the mirror game in various configurations. Multiplayer extensions to the mirror game have also been studied, where people in a group perform some joint motor task, e.g., synchronizing the oscillatory motion of their fingers. Alderisio et al. (2017) modeled this interaction using Kuramoto oscillators. Subsequently, Lombardi et al. (2019, 2021a) presented reinforcement learning-based cognitive architectures to display a specific motor signature, a quantity uniquely identifying different individuals (Slowinski et al., 2016). However, to the best of our knowledge, schemes that alter a participant's motion to achieve specific goals, such as increasing coordination—which has been found beneficial for human well-being (Rennung and Göritz, 2016)—have not been studied in the context of these motor tasks.

# 3 ROLE AND ASSUMPTIONS OF L1, L2 COGNITIVE ARCHITECTURES

For the sake of clarity, we recall the roles of the L1 and L2 CAs. The L1 CA has to manipulate minimally the input motion of a participant it represents, so that the output motion replicates the input while compensating for digital alterations such as delays and noise. The L2 CA has to modify the input of a participant, referred to as the *assisted participant*, so that the altered motion achieves better values of one or more metrics of interest, while still being similar to the motion of the assisted participant.

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 $T \in \mathbb{R}_{>0}$, where $d \in \mathbb{N}_{\geq 1}$ 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 creation of such libraries is currently part of WP2: "Sensorimotor primitives of social interaction".

A4 The 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 inverse kinematic being left as part of the animation.

The input signals mentioned in Assumption A1 are stored in a circular buffer, which acquires new data points as the participant/virtual humans (VHs) move, contextually deleting old data points. In the context of tasks akin to the PoP of Amplification, 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 assisted human movements are possible in a task and decide which ones to perform and how. In the context of tasks akin to the PoP of Social Connectedness, A2 is used to select the modifications that the L2 CA should apply to the assisted participant's motion, whereas A3 is used to convert 3D motion to the phase domain and vice versa.

# 4 L1 COGNITIVE ARCHITECTURE

In the SHARESPACE project, L1 virtual humans are designed to mimic user movements with precision. To achieve an accurate portrayal of human movements, the L1 cognitive architecture (CA) needs to process data as it is received.

The design of the L1 CA remains unchanged from that presented in Deliverable D5.1. Specifically, the L1 CA must take into account the technical details of the sensors and computation modules used and the typical delays in data transmission, as outlined in Deliverable D1.7. Presently, body kinematics are captured using inertial motion units (IMUs), strategically placed to track the movements of most human joints. To counteract potential signal drift issues with IMUs (Ahmad et al., 2013), the data is integrated with information from cameras. Therefore, at this stage of development, the L1 CA does not require the integration of signal compensation mechanisms.

# 5 L2 COGNITIVE ARCHITECTURE

L2 virtual humans can make autonomous decisions to modify the movement of a human participant—whom we refer to as an *assisted participant*—to improve user experience during group tasks and amplify/attenuate user intentions or emotions encoded in their movement. In particular, the L2 cognitive architecture has to satisfy three crucial requirements:

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

These requirements present a multi-objective challenge that the L2 cognitive architecture must address. To tackle this challenge, the L2 architecture needs to carefully balance three key objectives: firstly, ensuring smooth information flow from the participant to the L2 virtual human (VH); secondly, implementing meaningful modifications to the original motion; and thirdly, optimizing computational efficiency. The specific design of the cognitive architecture (CA) depends heavily on the particular group task being undertaken. Therefore, to clarify the implementation of the L2 CA, we refer to the Proofs of Principle outlined in Deliverable D1.2. The PoP of Amplification serves a benchmark to showcase the ability of the L2 CA to amplify emotion in movement, whereas the PoP of Social Connectedness is used to demonstrate the capabilities of the L2 CA in facilitating the achievement of a certain shared goal (in this case, synchronization).

## 5.1 L2 CA FOR THE POP OF AMPLIFICATION

In the Proof of Principle of Amplification, the participants are asked to pass an object between each other to study how information encoded in their body kinematics is transmitted within a group (e.g., a chain of people). 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 an emotion—e.g., fear—through *kinematic encoding* (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.

Currently, as presented in Deliverable D2.1, *kinematic encoding* is performed through 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 as a representative motion primitive to be manipulated by the CA.

A set of experimentally acquired motion signals of people reaching an object was provided by UKE and used to populate a preliminary *movement primitives library* for this PoP. This serves as a labelled database of examples of speed profiles that *kinematically encode* fear (informative signals) and speed profiles that do not. In general, the library required by the L2 CA is depicted in Figure 1 and is intended to be a collection of signals that define the movement primitives to be executed in a given task.

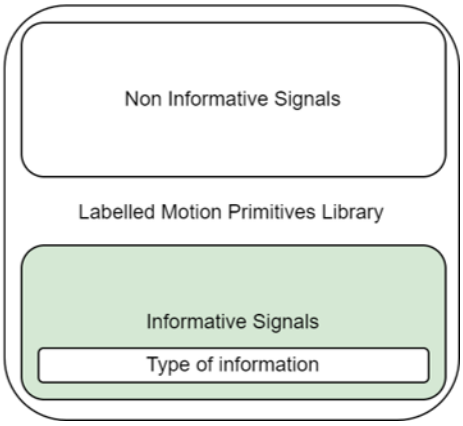


Figure 1. Schematic diagram of the movement primitives library.

More in general, each sample is a set of signals (specified by the *kinematic coding*) that are labeled to identify their properties. In particular, as the library has to contain movement primitives that cover the human movements required to complete the task (see Assumption A3), the labeling specifies:

- the type of task-dependent motion (e.g., reach to grasp, pass the object, return to rest).
- the presence of socially relevant information in the recorded signal (e.g., fear).

### 5.1.1 Working principles of the L2 cognitive architecture and main results

The main components required by the L2 cognitive architecture are described in Deliverable D5.1 and are schematically depicted in Figure 2.

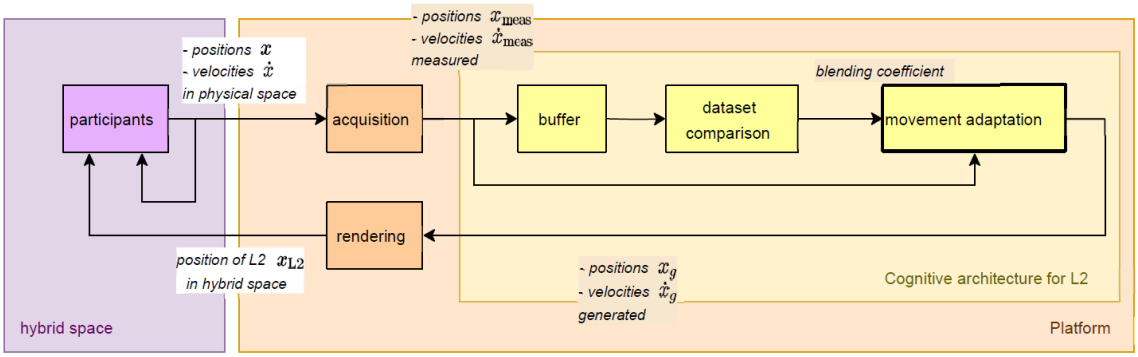


Figure 2. Block scheme of the proposed L2 cognitive architecture and its integration with other components being under development by other work packages (WP3: “Capturing”, WP4: “Rendering”).

With respect to the previous version of the L2 architecture (see Deliverable D5.1), in this version (v2.0), we:

1. modified the movement adaptation block to include a module that guarantees accurate object grasping also when the L2 CA modifies the human signals;
2. repurposed the architecture and retrained the L2 CA modules based on the findings of kinematic encoding reported in Deliverables D1.2 and D2.4.

Such upgrades were performed in collaboration with UKE which carried out the experimental campaign and assisted with data analysis.

The results of this collaboration produced the new version of the L2 CA which is described in detail in this report and in a peer-reviewed article (De Lellis, 2024) to be published in IEEE Control Systems Letters, that is attached as an appendix to this report.

### 5.1.2 Main results and advancements

In this section, we report the main advancements obtained and a comparison with the previous iteration of the L2 cognitive architecture presented in Deliverable D5.1. For a more in-depth analysis, please refer to De Lellis et al. (2024).

To upgrade the modeling capabilities, we used a training dataset of samples collected from 11 subjects in Hamburg by UKE. This allows the architecture to deal with different ways of encoding fear from different subjects. Compared to the previous iteration, the resulting dataset increased in size, with a total of 458 samples, 197 labeled as encoding fear and 261 as not encoding fear. Each sample records the 3D velocity and position of a participant's wrist at a 100 Hz sampling rate. However, during the training stage, we only use the velocity signals, as they carry the most information, according to Deliverable D2.1. From this dataset, we exclude roughly 30% of the samples, which we do not use for training but only for validation purposes to test model generalization.

The training procedure follows the same logic presented in Deliverable D5.1. However, we also trained an extra module based on Reinforcement Learning to guarantee object grasping by imposing a terminal distance between the end-effector and the object to be no greater than 20 mm. For numerical details concerning this module, we refer to De Lellis et al. (2024).

To validate the online control strategy, we replay samples from the validation dataset. The numerical results of the experiments are summarized in Figure 3 below, where a human velocity signal not encoding fear is transformed into one that encodes fear, according to the approximate encoding function.

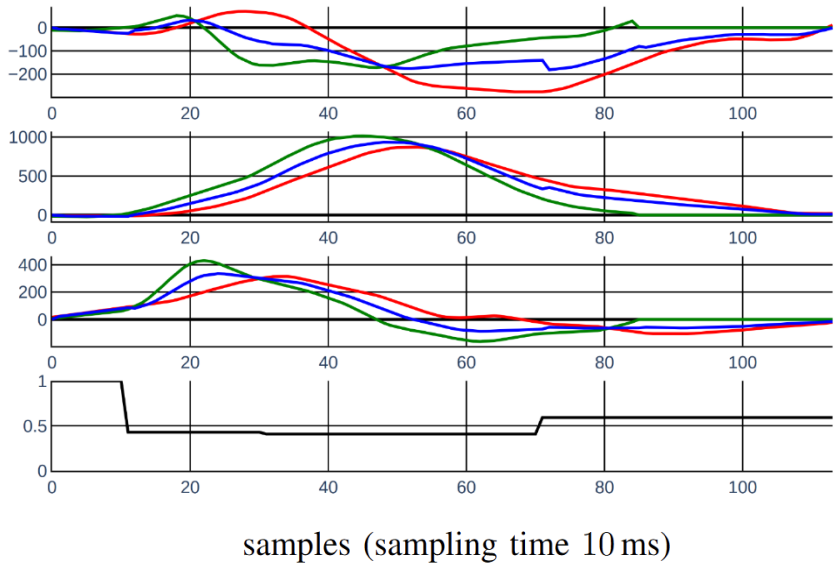


Figure 3. Representative online alteration of a human motion not encoding fear(red signal) the 3D space, whose components are denoted as x, y, z. The green line is motion sample extracted from the motion primitive library of a motion encoding fear. The alteration of the origin human signal produced by the L2 CA is depicted in a blue solid line. The bottom panel shows the blending decision taken by the architecture every 10 samples.

For a comprehensive view of the results of our approach, we applied the CA to each of the 65 experimentally obtained human movements not encoding fear from the validation dataset. Using a blending coefficient computed online, the architecture achieves an 89% success rate, with the constraint on the final position (not enforced in this case) being satisfied in 24% of cases. However, we also note that the final distance at the end of the performed movement stays at an average value of 31 mm. When applying this solution in combination with the reinforcement learning module, the final position constraint satisfaction increased to 90%. However, the success rate for classifying the presence of fear decreased to 65%.

The results from our experimental validation demonstrate the effectiveness of our approach. However, there is a clear tradeoff between the accuracy of emotional encoding and adherence to endpoint position constraints. This tradeoff arises because the problem becomes a multi-objective optimization task when additional constraints on the end-effector position are introduced and needs to be taken into account depending on the task defined in the proof of principles and scenarios (see Deliverable D1.2).

### 5.1.3 Implementation and reproducibility

The numerical simulation has been carried out with Python and the code is provided on GitHub.[1] The training algorithms have been built using TensorFlow 2. 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 $T_s$ = 10 ms; during deployment, the L2 CA compares the data stored in the buffer with the movement primitives library every 10 samples obtained. In this context, a single iteration can take up to about 2 ms to perform the search of the movement primitives library, and a RAM usage below 1 GB. However, these time lengths and storage requirements are expected to increase when the library grows larger: this relation will be further investigated in the future implementations of the L2 CA. We refer to De Lellis (2024) for further technical details.

## 5.2 L2 CA FOR THE PoP OF SOCIAL CONNECTEDNESS

In this PoP, participants are asked to perform a periodic motor task and contextually synchronize their motion. Specifically, participants stand in a circle, facing each other so that a certain predetermined graph of communication is established, and are asked to move their arms back and forth along an axis parallel to the ground, towards the center of the group. A detailed description of this PoP is provided in Deliverable D1.2.

The role of the L2 CA is to alter the motion of the assisted participant, so as to increase the coordination level of the group, while maintaining the altered motion close to that of the assisted participant, so that the latter still recognizes the altered motion as natural and realistic and as being their own.

In Deliverable D5.1, we introduced a preliminary design of the L2 CA for this PoP. In the rest of this section, we present a refined implementation of this CA, and validate it numerically.

[1] https://github.com/FrancescoDeLellis/L2_Cognitive_Architecture_PoP_Amplification

### 5.2.1 Mathematical model

For the sake of completeness, we briefly review the mathematical model considered for the design of the CAs for this PoP. In Alderisio et al. (2017), it was shown that the dynamics of people performing periodic group motion akin to that of this PoP can be modeled by a Kuramoto network model, that is

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

where $\theta_i(t) \in \mathbb{S}^1$ is the phase of participant $i$, with $\mathbb{S}^1 = \mathbb{R}/2\pi\mathbb{Z}$ being the *1-sphere* (loosely, the periodic interval $[0,2\pi]$), $\omega_i(t) \in \mathbb{R}$ is the natural frequency of the $i$-th participant, $c \in \mathbb{R}_{>0}$ is a scalar coupling strength, $N$ is the number of participants, $a_{ij}$ 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.

The level of synchronization of the group is measured through the order parameter $r: \mathbb{S}^N \to [0,1]$, given by (Calabrese et al., 2022)

$$r(t) := \frac{1}{N}\left|\sum_{i=1}^{N} e^{j\theta_i(t)}\right|. \tag{2}$$

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 the average of $r$ over the whole interaction.

### 5.2.2 Design of the L2 CA for the PoP of Social Connectedness

In Deliverable D5.1, we presented a preliminary scheme of the implementation of the L2 CA, where a *blender* module combined the position of an assisted participant (L0) with a reference position, with the goal of improving synchronization. To facilitate deployment in the SHARESPACE Scenarios, where position signals might be more complicated, here we present and validate a revised scheme, where the blending happens in the phase domain; see Figure 4. Below, we explain in more detail each of the blocks in the schematic diagram in Figure 4 and the modifications we made with respect to the previous design, reported in Deliverable D5.1.

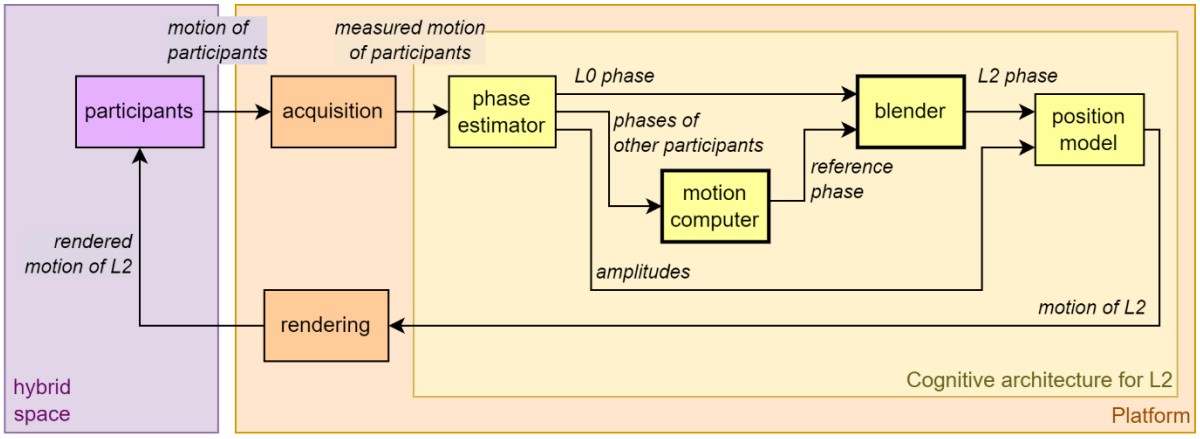


Figure 4. Block scheme of the L2 cognitive architecture for the PoP of Social Connectedness.

***"Phase estimator" block*** The "phase estimator" block takes as input the discretized linear motions of the VHs/participants and outputs the estimated phases $\theta_i$ of all VHs/participants and the estimated

amplitudes of their motion, in the sense of ( 1 ). In this PoP, we use an estimation algorithm (reported below as Algorithm 1) that is a slightly revised version of that presented in Deliverable D5.1. It assumes that the position signal of each participant alternates one maximum, one crossing of zero, one minimum, one crossing of zero, and so on. The previous implementation resulted in occasional erroneous updates of the amplitudes when a point with zero velocity but non-zero acceleration was found; the issue was resolved by adding a check on the value of the acceleration. Moreover, we verified that a fast low-pass filter significantly improves phase estimation performance (see Section 5.2.3.1).

**Algorithm 1:** Phase and amplitude estimation

**Input:** Positions $p_{t-1}$, $p_t$; velocities $v_{t-1}$, $v_t$; acceleration $a_t$; amplitudes $A_{t-1}^{p>0}$, $A_{t-1}^{p<0}$, $A_{t-1}^{v>0}$, $A_{t-1}^{v<0}$.

**Output:** phase $\theta_t$; amplitudes $A_t^{p>0}$, $A_t^{p<0}$, $A_t^{v>0}$, $A_t^{v<0}$.

1 **if** $p_{t-1} < 0 \wedge p_t \geq 0 \wedge v_t < 0$, **then** $A_t^{v>0} = |v_t|$, **else** $A_t^{v>0} = A_{t-1}^{v>0}$;
2 **if** $p_{t-1} \geq 0 \wedge p_t < 0 \wedge v_t > 0$, **then** $A_t^{v<0} = |v_t|$, **else** $A_t^{v<0} = A_{t-1}^{v<0}$;
3 **if** $v_{t-1} \geq 0 \wedge v_t < 0 \wedge a_t < 0$, **then** $A_t^{p>0} = |p_t|$, **else** $A_t^{p>0} = A_{t-1}^{p>0}$;
4 **if** $v_{t-1} < 0 \wedge v_t \geq 0 \wedge a_t > 0$, **then** $A_t^{p<0} = |p_t|$, **else** $A_t^{p<0} = A_{t-1}^{p<0}$;
5 **if** $p_t \geq 0$, **then** $p_t^{\text{norm}} = p_t / A_t^{p>0}$, **else** $p_t^{\text{norm}} = p_t / A_t^{p<0}$;
6 **if** $v_t \geq 0$, **then** $v_t^{\text{norm}} = v_t / A_t^{v>0}$, **else** $v_t^{\text{norm}} = v_t / A_t^{v<0}$;
7 $\theta_t = \text{atan2}(-v_t^{\text{norm}}, p_t^{\text{norm}})$;

Algorithm 1. Phase estimation from linear motion.

***"Motion computer" block*** This block takes as input the phases $\theta_i$ of all VHs/participants and outputs a *reference phase* for the L2 VH, say $\theta_{\text{ref}}$, with the assumption that, if executed, $\theta_{\text{ref}}$ would be more beneficial in terms of achieving the goal considered (here, increasing coordination) with respect to the phase of the assisted participant. Here we compute $\theta_{\text{ref}}$ as the phase provided by an L3 CA, which is based on reinforcement learning, and whose implementation details can be found in Deliverable D5.4.

***"Blender" block*** This block generates the phase of the L2 VH, say $\theta_{\text{L2}}$ by combining the reference phase $\theta_{\text{ref}}$ and the assisted participant's phase $\theta_{\text{L0}}$. The working principle of the blender is to allow limited alteration to the assisted participant's phase. Moreover, the alteration can be larger if executed for short time periods, so as to allow the L2 CA to compensate for quick disrupting movement in the assisted participant's motion—such as those induced by aversive stimuli—while avoiding the participant feeling the L2's motion to be detached from their own. To achieve this, we set the output phase of the L2 CA, i.e., $\theta_{\text{L2}}$, as

$$\theta_{\text{L2}} = \theta_{\text{L0}} + \text{sat}_\lambda(\theta_{\text{ref}} - \theta_{\text{L0}}), \tag{3}$$

where $\text{sat}_\lambda(x) : \mathbb{R} \to [-\lambda, \lambda]$ is a saturation function with a (time-varying) threshold $\lambda(t) \in \mathbb{R}_{>0}$, limiting the amount of alteration performed over the participant's phase $\theta_{\text{L0}}$, and is given by

$$\text{sat}_\lambda(x) = \begin{cases} \lambda, & x > \lambda, \\ x, & -\lambda \leq x \leq \lambda, \\ -\lambda, & x < -\lambda. \end{cases} \tag{4}$$

(We also verified correct operation when using as saturation the differentiable function $x/(1 + |x/\lambda|^p)^{1/p}$). The saturation threshold $\lambda$ is dependent on the current conditions. In particular, $\lambda$ is adapted in real time to attain a certain time-varying value $\lambda_\infty$, obtained as (see Figure 5)

$$\lambda_\infty = \psi\left(\frac{\theta_{\text{L2}} - \theta_{\text{L0}}}{e_{\max}}\right) \cdot (\lambda_{\max} - \lambda_{\min}) + \lambda_{\min}, \tag{5}$$

where $\psi : \mathbb{R} \to [0,1]$ is given by

$$\psi(x) = \begin{cases} 0, & |x| \geq 1, \\ 1 - |x|, & |x| < 1. \end{cases}$$

(We also verified correct operation with a *bump function* having $\exp\left(1 - \frac{1}{1-|x|^2}\right)$ in place of $1 - |x|$).

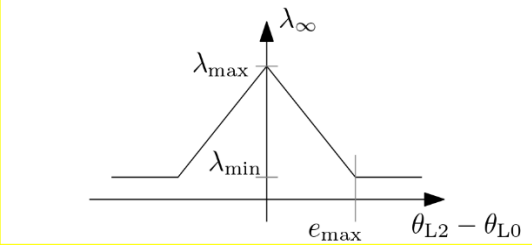


Figure 5. Function that sets the steady state value ($\lambda_\infty$) of the threshold $\lambda$.

In practice, $\lambda_\infty$ (which is the maximum allowed alteration) is large when $|\theta_{L2} - \theta_{L0}|$ is small, and $\lambda_\infty$ is small when $|\theta_{L2} - \theta_{L0}|$ is large. If $\lambda$ were set at all times equal to $\lambda_\infty$, the maximum saturation allowed (i.e. the maximum value $\lambda$ could take) would be given by the intersection of the function in ( 5 ) with the bisector (i.e., the curve $\lambda_\infty = \theta_{L2} - \theta_{L0}$). However, both to ensure $\lambda$ varies smoothly and to allow a behavior where quick large alterations are possible, we make set $\lambda$ as the solution to the first order piecewise-smooth asymptotically stable system

$$\frac{d\lambda(t)}{dt} = \begin{cases} -\kappa^+(\lambda - \lambda_\infty), & \lambda \leq \lambda_\infty, \\ -\kappa^-(\lambda - \lambda_\infty), & \lambda > \lambda_\infty, \end{cases} \qquad \lambda(0) = \lambda_{\max}, \tag{6}$$

where $\kappa^+, \kappa^- \in \mathbb{R}_{>0}$ are two coefficients; moreover, we select $\kappa^- > \kappa^+$, so that $\lambda$ can quickly be reduced if the system determines that the alteration to the participant's motion must be limited. Note that ( 5 ) and ( 6 ) together allow relatively large alterations to $\theta_{L0}$ only for a short time. Specifically, when $|\theta_{ref} - \theta_{L0}|$ stays small, then $\lambda_\infty$ is high [see ( 5 )], and $\lambda$ reaches $\lambda_\infty$ (close to $\lambda_{\max}$) (with a time constant $1/\kappa^+$), meaning that large (future) alterations are allowed. Conversely, if $|\theta_{ref} - \theta_{L0}|$ grows large, $\lambda_\infty$ is reduced [see ( 5 )], and $\lambda$ quickly reaches $\lambda_\infty$ (close to $\lambda_{\min}$) (with a time constant $1/\kappa^-$).

### 5.2.3 Validation of the L2 CA for the PoP of Social Connectedness

#### *5.2.3.1 Validation of the phase estimation algorithm for 1D motion*

We start by validating Algorithm 1, used to extract online the phase of a 1D oscillating motion. In Figure 6, we report a real experimental position signal, with the offline phase estimated through the Hilbert transform. To test the robustness of the Algorithm, we selected a position signal containing portions where it is constant, due to acquisition issues. In Figure 7 and Figure 8 we report the different performances of the revised Algorithm 1 presented here, including a low-pass filter with cutting frequency at 40 Hz (applied before the algorithm), compared to the previous version of the algorithm, as reported in Deliverable 5.1. Both figures show significantly better performance of the revised algorithm, and relatively small error when compared to the offline estimation performed with the Hilbert transform. The simulation code is available on GitHub.[2]

[2] https://github.com/SINCROgroup/online_phase_estimator_1d.

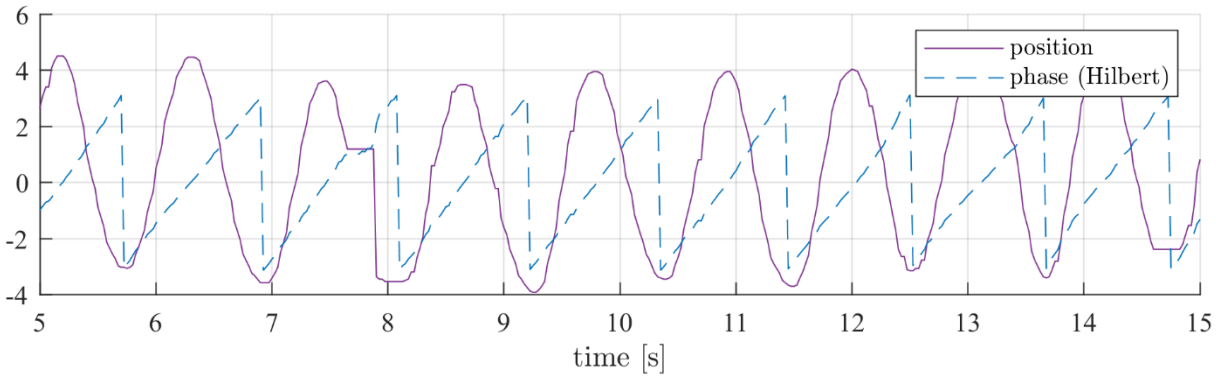


Figure 6. Offline estimation via the Hilbert transform of the phase of a real position signal.

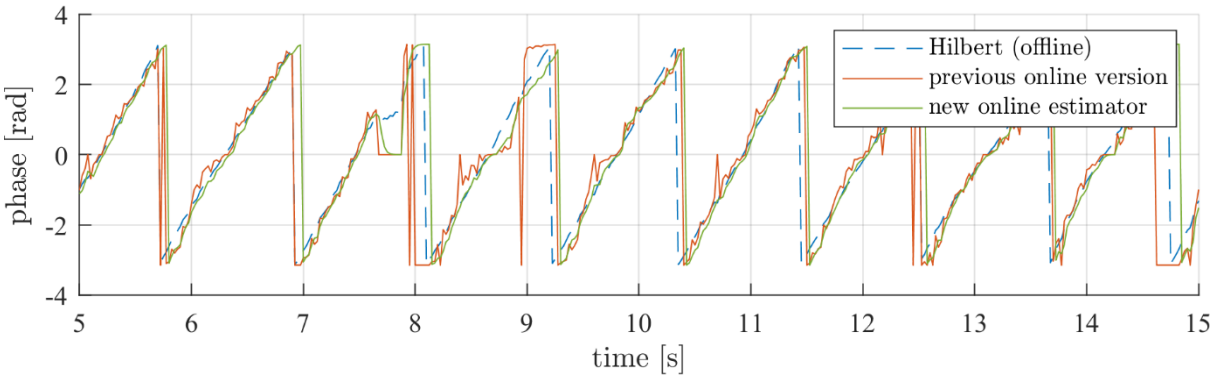


Figure 7. Phase estimation of the revised Algorithm 1 presented here (including a low-pass filter applied before the algorithm) compared to the previous version in Deliverable D5.1.

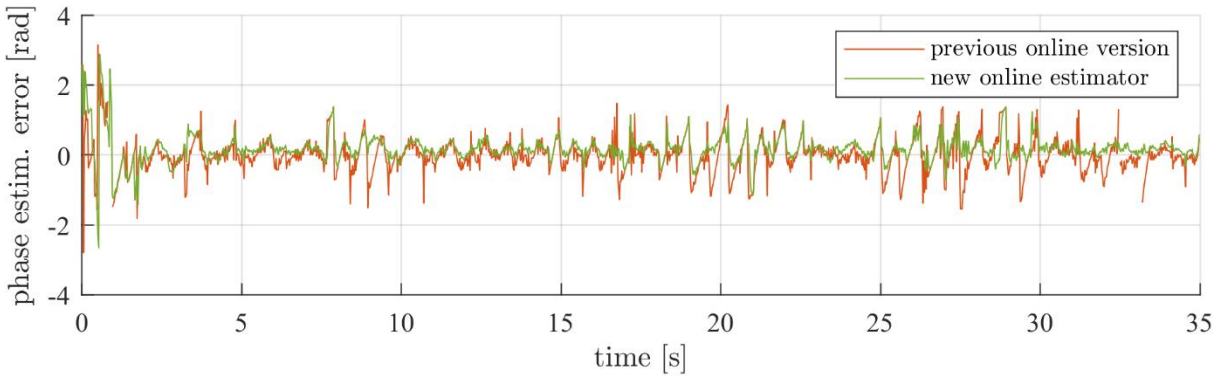


Figure 8. Phase estimation error of the revised Algorithm 1 presented here (including a low-pass filter applied before the algorithm) compared to the previous version in Deliverable D5.1.

### *5.2.3.2 Validation of the blending mechanism*

To validate the L2 CA, we deploy it in simulation over 3 different groups, parametrized using real data. To perform the simulations, we use a slight variation of model ( 1 ), that is

$$\dot{\theta}_i(t) = \omega_i(t) + \frac{c}{N}\sum_{j=1}^{N} a_{ij} \sin\left(\tilde{\theta}_j(t) - \tilde{\theta}_i(t)\right), \qquad \forall i \in \{1, \dots, N\},$$

where $\tilde{\theta}_i = \theta_i$ if participant $i$ is *not* the assisted participant, whereas $\tilde{\theta}_i = \theta_{\mathrm{L2}}$ if parcitipant $i$ is the assisted participant. This modification models the fact that all participants (including the assisted one) see the L2 VH's phase in place of that of the assisted participant. Furthermore, the model is discretized

via forward Euler with time sampling set to $0.01$ s, all phases initialized to $0$ rad, and participants are connected in a ring configuration. To parametrize the models, we picked the natural frequencies and the coupling gain from Calabrese et al. (2022) and Alderisio et al. (2017), that were estimated from experimental data in similar oscillatory group motion. The reference motion ($\theta_{\mathrm{ref}}$) was generated by an L3 CA (presented in Deliverable D5.4).[3] When deciding which simulated participant was to be substituted with an L2, we verified that if an L3 CA was substituted with the participant, the order parameter would increase, to ensure that the L2 is blending the assisted participant's motion with one that can be more beneficial to the group.

The parameters associated with each group are reported in Table 2.

| Group # | $N$ | $c$ | $\omega$ | Assisted participant: | Simulation time |
|---|---|---|---|---|---|
| Group 1: (Calabrese et al. 2022) | 5 | 1.6 | [3.04, 6.36, 3.34, 9.91, 5.21] | #4 | 10 s |
| Group 2: (Alderisio et al., 2017; group 1 in the paper) | 7 | $\frac{1.25}{N} \approx 0.18$ | [4.2568, 4.3143, 4.6691, 4.2951, 4.3623 2.9433, 4.2184] | #6 | 300 s |
| Group 3: (Alderisio et al., 2017; group 2 in the paper) | 7 | $\frac{4.4}{N} \approx 0.63$ | 2.7151, 2.9299, 4.0344 2.1476, 3.9117, 3.7429, 3.2827 | #6 | 300 s |

Table 2. Parameter values used for the validation of the L2 CA for the PoP of Social Connectedness.

As parameters for the L2 CA, we used $\lambda_{\max} = \pi/2$, $\lambda_{\min} = 0.3\,\pi/2$, $e_{\max} = 0.3\,\pi/2$, $\kappa^{-} = 2.5$, $\kappa^{-} = 1$. The simulation code is available on GitHub.[4]

In Figure 9., we report the order parameter observed with and without the addition of L2 to Group 1 (obtained performing separate simulations), noting an improvement in the coordination level when the latter is used. In Figure 10., we report the phase of the assisted participant ($\theta_{\mathrm{L0}}$), the reference phase ($\theta_{\mathrm{ref}}$), and the L2 blended phase ($\theta_{\mathrm{L2}}$): when $\theta_{\mathrm{L0}}$ and $\theta_{\mathrm{ref}}$ are close, the L2 CA selects the latter, whereas when the two are far apart, $\theta_{\mathrm{L2}}$ remains close to $\theta_{\mathrm{L0}}$, compatibly with the intended design goal.

In Table 3, we report the value of the order parameter obtained in all 3 groups, noting an improvement in all cases. Importantly, we verified that if the values of the parameters $\lambda_{\min}$ and $\lambda_{\max}$ used in the L2 CA are increased, a higher coordination improvement is observed, as the L2 CA can choose a phase, $\theta_{\mathrm{L2}}$, that is in general closer to $\theta_{\mathrm{ref}}$,. However, doing so might conflict with the requirement of keeping $\theta_{\mathrm{L2}}$ close to $\theta_{\mathrm{L0}}$, to avoid the possibility that the assisted participant does not recognize the altered motion as their own. Therefore, we advise against picking values of $\lambda_{\min}$ and $\lambda_{\max}$ larger than those used here.

[3] The action space of the L3 CA was $[-0.5, -0.4, \dots, 0.5]$. The frequencies were bounded in $[-15,\ 15]$ rad/s.

[4] https://github.com/SINCROgroup/l2_ca_pop_sync.

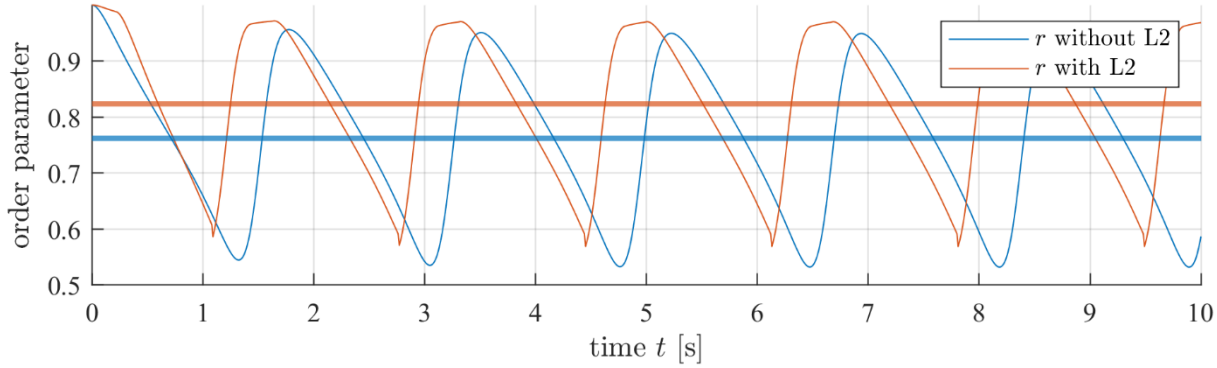


Figure 9. Comparison of the order parameter with and without L2 in Group 1.

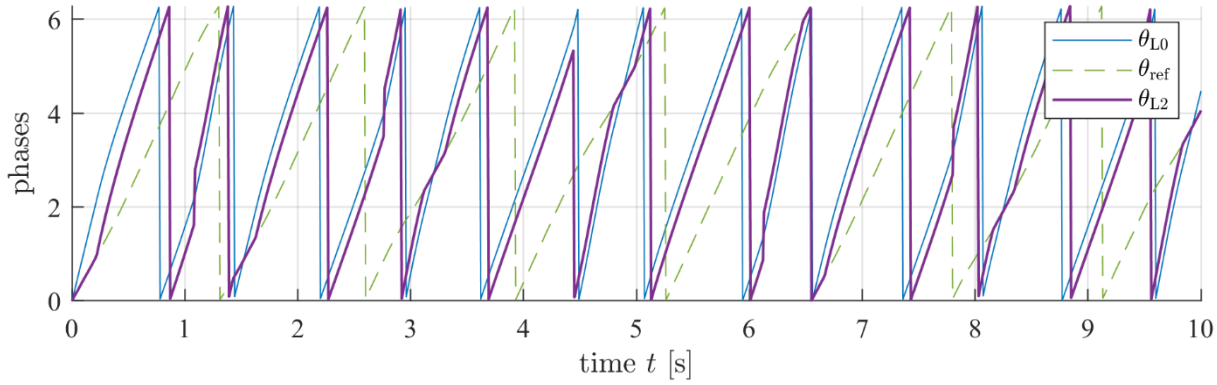


Figure 10. Group 1. Phase of the assisted participant ($\theta_{L0}$), reference phase ($\theta_{ref}$), and L2 blended phase ($\theta_{L2}$).

| Group # | Order parameter without L2 | Order parameter with L2 (variation) |
|---|---|---|
| 3 | 0.76 | 0.82 (+8.14%) |
| 1 | 0.47 | 0.49 (+5.59%) |
| 2 | 0.61 | 0.64 (+4.80%) |

Table 3. Order parameter observed in the numerical validation of the L2 CA for the PoP of Social Connectedness.

# 6 PHASE ESTIMATION IN 3D

The L2 CA presented in Section 5.2, designed to operate in the PoP of Social Connectedness, manipulates phases, as an abstraction of motion. To apply this CA in the SHARESPACE Scenarios, modules are required to transform arbitrary motion to the phase domain and vice versa. Hence, in this section, we consider a quasi-periodic trajectory in 3D and present and validate an original method to estimate online the phase [defined, e.g., in Kralemann et al. (2008)] of a point on the trajectory and to reconstruct the phase associated to any point on the trajectory. This module will be essential in extending the cognitive architectures used in the PoP of Social Connectedness (which operate in the phase space) onto the Scenarios, particularly those where periodic motion is performed, such as Health and possibly Art.

## 6.1 FROM POSITION TO PHASE

In Figure 11, we portray a block scheme representation of the method to extrapolate phase from position.

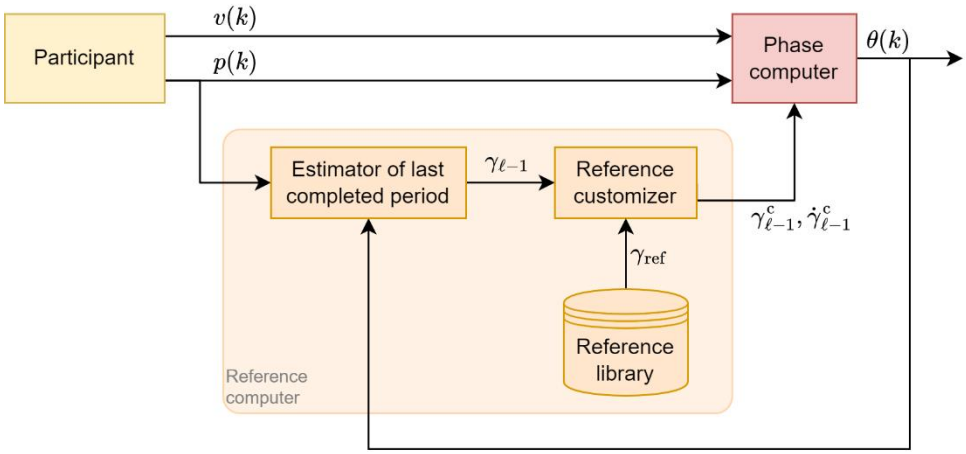


Figure 11. Block scheme of the Phase estimator for 3D space periodic motion.

We denote by $p(k) \in \mathbb{R}^3$ the position at time instant $k \in \mathbb{Z}$ of a point on the body of a person performing a motion and by $v(k) \in \mathbb{R}^3$ the velocity of that point. In Figure 11, the "phase computer block" takes as input the motion $[p_i(k)\ \ v_i(k)]$ and outputs the estimated phase $\theta(k)$. The estimation exploits a reference motion $\gamma_{\text{ref}} \in \mathbb{R}^{3\times N}$ ($N$ denotes a number of samples in time), representing a reference motion that the person should attempt to follow, and which is stored in a pre-existing movement primitives library.

### 6.1.1 Assumptions

The online 3D phase estimation method requires a series of assumptions, i.e.:

1. A reference motion $\gamma_{\text{ref}}$ is available (with $\theta = 0$ rad being associated to the beginning of the motion).
2. The person's periodic motion describes a curve that has a shape which does not significantly differ from that of the reference motion $\gamma_{\text{ref}}$ (except for possible translation, rotation, and scaling).
3. The initial part of the person's motion resembles the initial part of $\gamma_{\text{ref}}$ (in practice, this means the person's motion starts with phase equal to $0$ rad).
4. The person's motion does not significantly differ between two successive periods.

### 6.1.2 Algorithm

Here, we explain in detail the different blocks displayed in Figure 11. To do so, let us introduce some notation. We use index $\ell \in \mathbb{N}$ to denote the $\ell$-th time period. We let $k_{\text{P}\ell} \in \mathbb{N}$ denote the first time instant of the $\ell$-th recorded period. Hence, $M_\ell = k_{\text{P}(\ell+1)} - k_{\text{P}\ell}$ denotes the number of samples in the $\ell$-th recorded period and $\gamma_\ell \coloneqq [p(k_{\text{P}\ell}) \ \cdots \ p(k_{\text{P}\ell+1} - 1)] \in \mathbb{R}^{3\times M_\ell}$ is the trajectory executed by the person in that period. We denote by $\gamma^{\text{c}}_\ell \in \mathbb{R}^{3\times N}$ the *customized reference motion*, which is a transformed version of $\gamma_{\text{ref}}$ to make it similar to $\gamma_\ell$. The transformation is an affine one, thus admitting rotation, translation, and scaling; see Figure 12 for an example. We note that the phase estimation process will need to estimate $\gamma_1$ and $\gamma^{\text{c}}_1$, which happens at the end of the first period. Hence, the phase

cannot be estimated during the initial period, but its estimation starts in the next period. We are now ready to describe the purpose and operation of the blocks in Figure 11.

**"Reference computer" block** This macro-block has the purpose of computing $\gamma^{\mathrm{c}}_{\ell-1}$ (and the associated velocities)

a. The "*Estimator of last completed period*" block takes as input the motion $p(k)$ and outputs $\gamma_{\ell-1}$. To compute $\gamma_1$, it is necessary to identify $k_{\mathrm{P2}}$. To achieve this, we seek a segment of trajectory such that the motion starts repeating itself. Formally, we set $h \in \mathbb{N}$ and define the distance $d(k)$ between segments of motion of length $h \in \mathbb{N}_{>1}$ as

$$d(k) \coloneqq \frac{1}{h}\sum_{j=0}^{h-1} \|p(k_{P1}+j) - p(k-(h-1)+j)\|, \quad \forall\, k > k_{\mathrm{P1}} + h.$$

Let $k^*$ be the smallest time instant (larger than $k_{\mathrm{P1}} + h$ ) such that

$$d(k^*) < d(k), \qquad \forall k \in \{k^*-2, k^*-1, k^*+1, k^*+2\}.$$

Then we set $k_{\mathrm{P2}} = k^* - (h-1)$. Therefore, $\gamma_1$ is now available and will be used to estimate the phase of the motion in the second period ($\ell = 2$). For periods $\ell > 1$, $k_{\mathrm{P}\ell}$ is estimated differently; namely, as the time instant where the estimated phase (which is now available, as the estimation in period $\ell$ only requires knowledge of $\gamma_{\ell-1}$) crosses $2\pi$.

b. The "*Reference customizer*" block takes as input a period trajectory, say $\gamma_\ell$ and the reference trajectory ($\gamma_{\mathrm{ref}}$) and computes the customized reference trajectory $\gamma^{\mathrm{c}}_\ell$, by finding some appropriate translation, scaling and rotation to minimize the distance between $\gamma^{\mathrm{c}}_\ell$ and $\gamma_{\mathrm{ref}}$. The process, described below, exploits Arun et al. (1987); an example is portrayed in Figure 12.

   i. Resampling: Let $\gamma^{\mathrm{r}}_{\mathrm{ref},\ell}$ be a resampling of $\gamma_{\mathrm{ref}}$ obtained by linear interpolation, so that it has the same number of samples (i.e., $M_\ell$) of $\gamma_\ell$.

   ii. Translation: Let $\gamma^{\mathrm{t,r}}_{\mathrm{ref},\ell}$ and $\gamma^{\mathrm{t}}_\ell$ be translated versions of $\gamma^{\mathrm{r}}_{\mathrm{ref},\ell}$ and $\gamma_\ell$, so that the centroids of the former are at the origin.

   iii. Scaling: Let $\sigma$ denote a standard deviation operator and define

$$\sigma^x \coloneqq \frac{\sigma\left(\gamma^{\mathrm{t}}_{\ell,x}\right)}{\sigma\left(\gamma^{\mathrm{t,r}}_{\mathrm{ref},\ell,x}\right)}; \quad \sigma^y \coloneqq \frac{\sigma\left(\gamma^{\mathrm{t}}_{\ell,y}\right)}{\sigma\left(\gamma^{\mathrm{t,r}}_{\mathrm{ref},\ell,y}\right)}; \quad \sigma^z \coloneqq \frac{\sigma\left(\gamma^{\mathrm{t}}_{\ell,z}\right)}{\sigma\left(\gamma^{\mathrm{t,r}}_{\mathrm{ref},\ell,z}\right)},$$

   where the subscripts $x, y, z$ denote components of the motion along the homonymous axes. The scaled trajectory is then computed as

$$\gamma^{\mathrm{s,t,r}}_{\mathrm{ref},\ell} = \gamma^{\mathrm{t,r}}_{\mathrm{ref},\ell-1} \cdot [\sigma^x\ \sigma^y\ \sigma^z]^T$$

   iv. Rotation: Compute the covariance matrix $\sum_{i=1}^{M_\ell} q_i\, {p_i}^T$, where $q_i \in \gamma^{\mathrm{s,t,r}}_{\mathrm{ref},\ell}$ and $p_i \in \gamma^{\mathrm{t}}_\ell$. Perform singular value decomposition on $H$ obtaining $H = U\Sigma V^T$, and let the rotation matrix $R \coloneqq \pm VU^T$, with the sign decided so that $|R| > 0$ (to avoid reflection). Finally, compute $\gamma^{\mathrm{o,s,t,r}}_{\mathrm{ref},\ell} \coloneqq R \cdot \gamma^{\mathrm{s,t,r}}_{\mathrm{ref},\ell}$, with the result that now $\gamma^{\mathrm{o,s,t,r}}_{\mathrm{ref},\ell}$ has the same orientation, scaling, and position (of its centroid) of $\gamma^{\mathrm{t}}_\ell$.

   v. Reverse translation: Let $\gamma^{\mathrm{c}}_\ell$ be the translated version of $\gamma^{\mathrm{o,s,t,r}}_{\mathrm{ref},\ell}$ so that $\gamma^{\mathrm{c}}_\ell$ has its centroid corresponding to that of $\gamma_\ell$ (and retains the same orientation and scaling of $\gamma^{\mathrm{o,s,t,r}}_{\mathrm{ref},\ell}$).

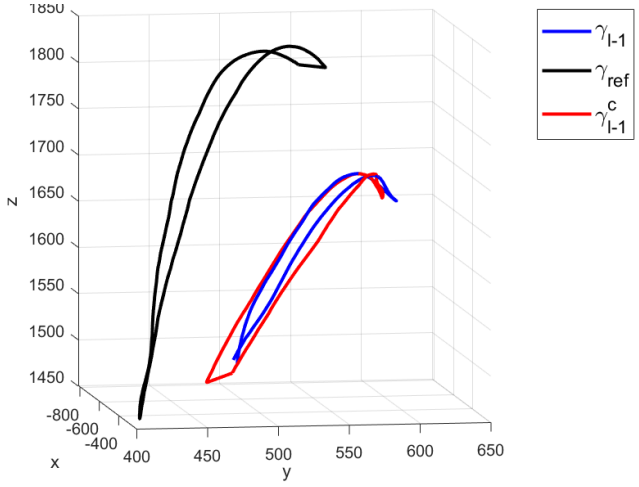


Figure 12. Example of reference customization.

**"Phase computer" block** This block takes as input the current position and velocity $[p(k), v(k)]$, the customized reference motion $\gamma^{c}_{\ell-1}$ and its derivatives $\dot{\gamma}^{c}_{\ell-1}$ (obtained via the backward finite difference method), and outputs the estimated current phase. The block calculates the distance between $[p(k), v(k)]$ and each point in $[\gamma^{c}_{\ell-1}, \dot{\gamma}^{c}_{\ell-1}]$ and identifies the index

$$i^*(k) = \arg\min_i \left\| \begin{bmatrix} p(k) \\ v(k) \end{bmatrix} - \begin{bmatrix} \gamma^{c}_{\ell-1}(j) \\ \dot{\gamma}^{c}_{\ell-1}(j) \end{bmatrix} \right\|, \qquad k \in [k_{P\ell}, k_{P\ell} - 1]$$

where $\|\cdot\|$ denotes the Euclidean norm. Then, we compute $\vartheta(k) = \frac{2\pi i^*(k)}{M_\ell}$.[5]

### 6.1.3 Validation (phase estimation)

We validated the 3D online phase estimation algorithm with real data provided by the UM team. Participants were instructed to move their arms back and forth in a periodic manner; the motion was recorded using the VICON platform. We selected $h = 10$ for our algorithm. The simulation code is available on GitHub.[6]

To assess the performance of our algorithm, we compared its online phase estimation with that obtained using a widespread offline algorithm utilizing *principal component analysis* (PCA) and the Hilbert transform. Specifically, the PCA is used to extract the principal direction where the informative content is the greatest; then the Hilbert transform is used to extract a phase from this 1D signal. A comparison between our online phase estimation (see Section 6.1.2) and the offline estimation algorithm (using PCA and the Hilbert transform) is depicted in Figure 13 and Figure 14, showing only limited error in the online estimation process with respect to the offline benchmark one.

[5] To simplify the computation of $i^*(k)$ (which has in general complexity $O(M_\ell)$), it is possible to provide $i^*(k-1)$ as a starting point in the search, and stop it once a local minimum (with a predetermined tolerance) is found.

[6] https://github.com/SINCROgroup/Phase_estimator3D.

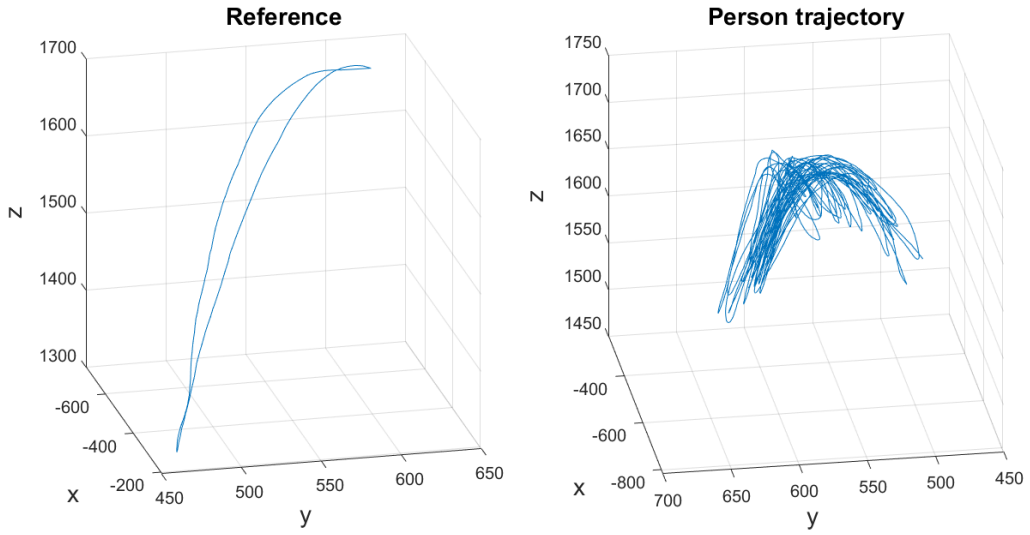


Figure 13. (Left) Reference trajectory $\gamma_{\mathrm{ref}}$ (extracted from the data provided by UM). (Right) Trajectory (position of a hand) executed by a person, captured using VICON by UM.

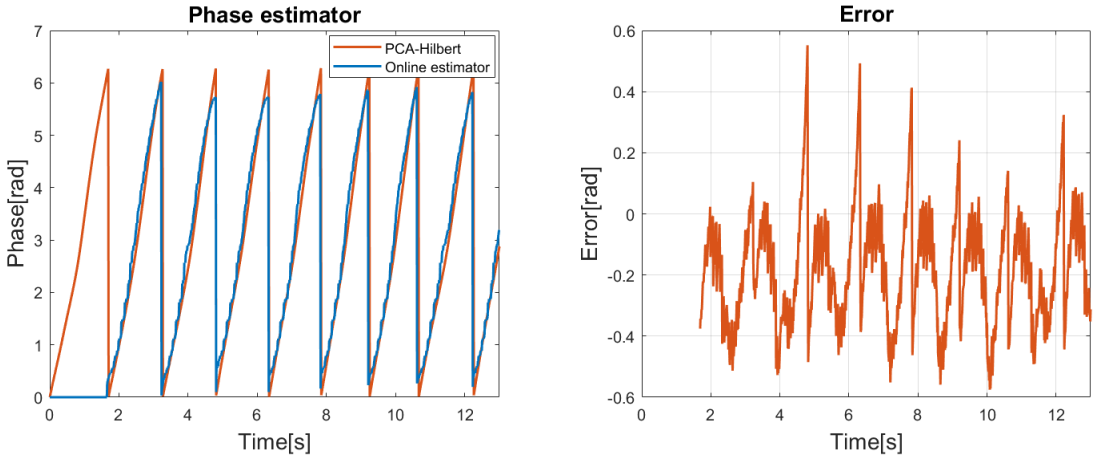


Figure 14. (Left) Phase estimations provided by our online algorithm and the offline PCA-Hilbert algorithm. (Right) Estimation error of the online estimation compared to the offline one (the first calibration period was neglected).

## 6.2 FROM PHASE TO POSITION

In Figure 15, we portray a block scheme representation of the method to reconstruct online a 3D position signal from a phase signal, given a reference trajectory.

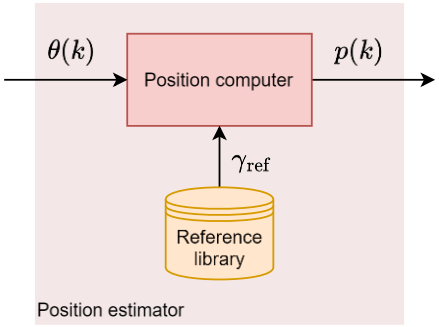


Figure 15. Block scheme of the online position estimator for 3D space periodic motion.

We denote by $\theta(k)$ the phase at time instant $k \in \mathbb{Z}$. In Figure 15, the "position estimator block" takes as input the phase $\theta(k)$ and outputs the estimated position $p(k) \in \mathbb{R}^3$ of a point in the 3D space, taken from a (possibly time-varying) reference motion $\gamma_{\text{ref}} \in \mathbb{R}^{3\times N}$ ($N$ denotes a number of samples in time), which is stored in a pre-existing library.

### 6.2.1 Algorithm

The **"Position computer"** block in Figure 15 takes as input the phase $\theta(k)$ and the reference motion $\gamma_{\text{ref}}$ and computes the position $p(k) \in \mathbb{R}^3$, which corresponds to the position on the reference motion associated with the phase $\theta(k)$. Formally, define the index $i(k) := \text{round}\left(\frac{N\,\theta(k)}{2\,\pi}\right)$, where $\text{round}: \mathbb{R} \to \mathbb{Z}$ denotes the rounding operation and $\theta(k) \in [0\,, 2\pi)$; the position is computed as

$$p(k) = \gamma_{\text{ref}}\big(i(k)\big).$$

### 6.2.2 Validation

Here we present a validation of the algorithm presented in Section 6.2.1. The phase signal $\theta(k)$ is generated through a first-order oscillator at a constant frequency ($\dot{\theta}(t) = \omega$). The reference trajectory $\gamma_{\text{ref}}$ was provided by UM. White noise was added to the position estimate to make the output more human-like (clearly, noise with different spectral content can also be used). The results are portrayed in Figure 16, showing the generation of a realistic position signal.

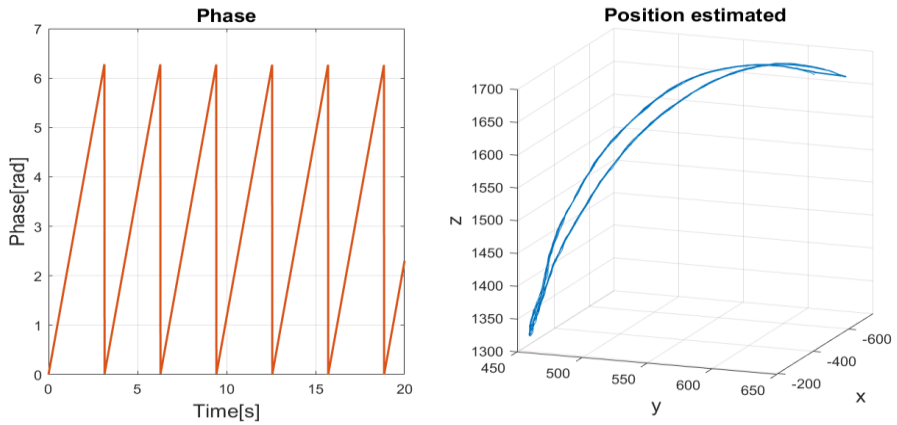


Figure 16. (Left) Phase trajectory. (Right) Estimated position on the reference trajectory $\gamma_{\text{ref}}$ (provided by UM).

# 7 CONCLUSIONS AND FUTURE WORK

In this deliverable, we presented revised versions of the cognitive architectures driving L1 and L2 virtual humans.

In particular, we confirmed the version of the L1 CA introduced in Deliverable D5.1, which does not need the integration of signal compensation mechanisms.

Concerning the L2 CA, we presented two revised implementations, one for each Proof of Principle. The two implementations share a common main design, i.e., the use of a so-called blender module that combines the motion of a person with some reference motion, designed to achieve a specific goal (displaying certain emotions in the PoP of Amplification and increasing coordination in the PoP of Social Connectedness). Then, the two implementations differ on how the reference motion is obtained, which is a task-specific operation. We presented successful validation of both implementations: namely, a numerical and experimental one for the PoP of Amplification and a numerical one for the PoP of Social Connectedness.

Moreover, we presented an online method to estimate the phase of a multidimensional position signal and to reconstruct a position in space starting from a phase. This methodology will be crucial in the extension of the technologies developed for the PoP of Social Connectedness to the Scenarios. The methodology was validated numerically exploiting real recorded motion data.

The next steps involve:

- Experimental testing of the L2 CA for both PoPs. For the L2 CA for PoP of Amplification, it is necessary to have real people evaluate the motion generated by the CA, ensuring it conveys the intended emotion or information, alongside assessments by the learned encoding model. The L2 CA for the PoP of Social Connectedness must be deployed in a group setting to confirm that it yields an observable increase in synchronization metrics.
- Extension to the Scenarios. The L2 CAs must be employed in the SHARESPACE Scenarios. Currently, we envision using the L2 CA for the PoP of Amplification in the Sports scenario, and the L2 CA for the PoP of Social Connectedness in the Health scenario, with potential applications in the Art scenario as well. A critical task in this process is determining which body joints (assumed to be a small number) the CA should modify. Such a decision will be made in consultation with other consortium partners.

# 8 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.

K. S. Arun, T. S. Huang and S. D. Blostein, "Least-Squares Fitting of Two 3-D Point Sets," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-9, no. 5, pp. 698-700, 1987.

A. Atkinson, W. Dittrich, A. Gemmell, and A. Young, “Emotion perception from dynamic and static body expressions in point-light and full-light displays,” Perception, vol. 33, no. 6, pp. 717–46, 2004.

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.

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.

C. Calabrese, B.G. Bardy, P. De Lellis, M. di Bernardo, “Modeling frequency reduction in human groups performing a joint oscillatory task,” Front. Psychol., 12, 2022

J.-A. Claret, G. Venture, and L. Basanez, “Exploiting the robot kinematic redundancy for emotion conveyance to humans as a lower priority task,” International Journal of Social Robotics, vol. 9, no. 2, pp. 277–292, 2017.

F. De Lellis, M. Coraggio, N. C. Foster, R. Villa, C. Becchio, M. di Bernardo, “Data-driven architecture to encode information in the kinematics of robots and artificial avatars” IEEE Control Systems Letters, to appear (doi: 10.1109/LCSYS.2024.3416071), 2024

F. D. Lellis, M. Coraggio, G. Russo, M. Musolesi and M. di Bernardo, "Guaranteeing Control Requirements via Reward Shaping in Reinforcement Learning," IEEE Transactions on Control Systems Technology, to appear (doi: 10.1109/TCST.2024.3393210), 2024

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.

G.C Homans, “The Human Group”, Routledge & Kegan Paul Ltd., London, 1st edition, 1951.

M.C. Howard, “A meta-analysis and systematic literature review of virtual reality rehabilitation programs”, Computational Human Behaviour, 70, 317–327, 2017.

B. Kralemann, L. Cimponeriu, M. Rosenblum, A. Pikovsky, and R. Mrowka, “Phase dynamics of coupled oscillators reconstructed from data,” Phys. Rev. E, vol. 77, no. 6, p. 066205, 2008.

J. E. Laird, C. Lebiere, and P. S. Rosenbloom, “A standard model of the mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics,” AI Magazine, vol. 38, no. 4, 2017.

P. Langley, J.E. Laird, and S. Rogers, “Cognitive architectures: Research issues and challenges”, Cognitive Systems Research, 10(2), 141–160, 2009.

J. Llobera, V. Jacquat, C. Calabrese, and C. Charbonnier, “Playing the mirror game in virtual reality with an autonomous character,” Scientific Reports, vol. 12, no. 1, p. 21329, 2022.

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, 2021a.

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, 2021b.

T. Lourens, R. Van Berkel, and E. Barakova, “Communicating emotions and mental states to robots in a real time parallel framework using Laban movement analysis,” Robotics and Autonomous Systems, vol. 58, no. 12, pp. 1256–1265, 2010.

M. Masuda and S. Kato, "Motion rendering system for emotion expression of human form robots based on Laban movement analysis," 19th International Symposium in Robot and Human Interactive Communication, 2010, pp. 324-329,.

D.L. Neumann, R.L. Moffitt, P.R. Thomas, K. Loveday, D.P. Watling, C.L. Lombard, S. Antonova, and M.A. Tremeer, “A systematic review of the application of interactive virtual reality to sport”, Virtual Reality, 22(3), 183–198, 2017.

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.

M. Rennung, and A.S. Göritz, “Prosocial Consequences of Interpersonal Synchrony: A Meta-Analysis”, Zeitschrift für Psychologie, 224(3), 168–189, 2016.

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

C. Zhai, F. Alderisio, K. Tsaneva-Atanasova, and M. di Bernardo, “A novel cognitive architecture for a human-like virtual player in the mirror game”, In Conf. Proc. 2014 IEEE Int. Conf. Syst. Man Cybern., 754–759, 2014

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.

# 9 APPENDIX

In this appendix we enclose the peer-reviewed scientific paper De Lellis et al. (2024), which provides an in-depth description of the L2 cognitive architecture for the PoP of Amplification (see Section 5.1). The paper results from the collaboration between CRdC and UKE and is to appear on the IEEE Control Systems Letters and to be presented at the 2024 IEEE Conference on Decision and Control.

This article has been accepted for publication in IEEE Control Systems Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2024.3416071

# Data-driven architecture to encode information in the kinematics of robots and artificial avatars

Francesco De Lellis[1], Marco Coraggio[2], Nathan C. Foster[3], Riccardo Villa[3], Cristina Becchio[3], Mario di Bernardo[1,2]

***Abstract*— We present a data-driven control architecture designed to encode specific information, such as the presence or absence of an emotion, in the movements of an avatar or robot driven by a human operator. Our strategy leverages a set of human-recorded examples as the core for generating information-rich kinematic signals. To ensure successful object grasping, we propose a deep reinforcement learning strategy. We validate our approach using an experimental dataset obtained during the reach-to-grasp phase of a pick-and-place task.**

***Index Terms*— Data-driven control, Machine learning, Human-in-the-loop control**

## I. Introduction

MOVEMENT encodes significant information about both the external characteristics of objects and the internal states of the mover, such as intentions and expectations [1]. For instance, the way an individual reaches towards an object can reveal the mover's anticipations regarding its weight [2] and their specific intentions, such as intending to pour or drink from a cup or bottle [3]. Thus, analyzing movement patterns allows us to infer others' internal states [1], [4]. However, accurately decoding or interpreting this information is challenging due to the variability in movement kinematics and the observer's ability to distinguish between informative and non-informative variations.

Experimental studies have shown that naive human observers can decode or read some, but not all, information from movement kinematics [1], with the potential to overlook informative variations [3]. During social interactions, individuals naturally modify their movement kinematics to make their actions more interpretable to others [5], highlighting the importance of conveying information clearly through movement. Therefore, for robots and virtual reality (VR) avatars to improve interactions with humans, it is essential to accurately encode information in their kinematics, making the encoded information more accessible and interpretable.

In this Letter, we present a data-driven control architecture capable of adjusting human-like movement kinematics to encode emotional states. The study focuses on the pick-and-place task, a common scenario in social robotics [6], specifically examining the reach-to-grasp movement. By combining live kinematics with a database of movements using AI tools, the proposed architecture enables real-time adjustments to the motion of avatars in VR or tele-operated robots in extended reality. This approach facilitates the encoding of information, making the movements of these entities more comprehensible and effective in social interactions.

[1]Department of Electrical Engineering and Information Technology, University of Naples Federico II, Naples, Italy

[2]Scuola Superiore Meridionale, Naples, Italy

[3]Department of Neurology, University Medical Center Hamburg-Eppendorf, Hamburg, Germany

This work was supported by the EU Research Project SHARESPACE (EU HORIZON-CL4-2022-HUMAN-01-14). For more info see www.sharespace.eu

### A. State of the art

The concept of emotion has been integrated into dynamical and control systems in diverse ways, reflecting its complexity. This integration spans decision-making architectures and attempts at mimicking cognitive processes, illustrating the variety of approaches in current research. In decision-making, a framework was developed in [7] combining model-based control and model-free reinforcement learning (RL), informed by cognitive science insights into emotional responses. Similarly, emotion's role in mimicking cognitive processes is evident in reinforcement learning strategies for recurrent neural network parameter tuning presented in [8], where emotional state-modulated reward functions enhance learning and control.

Studies have shown that human observers can recognize emotions from body movements, suggesting the feasibility of communicating emotions through motion [9], [10]. Indeed, efforts to interact with humans have led to socially aware robotic systems that communicate emotions, using kinematic redundancy to encode emotions in movements [11], and trajectory planning that incorporates emotional aspects, drawing on Laban Movement Analysis (LMA) [12]; for instance, in [13], it was shown that LMA could be used to adapt arbitrary movements of a humanoid robot to add target emotions, while in [11], an inverse kinematic strategy was provided to convey specific emotions in a humanoid robot movements exploiting kinematic redundancy. Moreover, VR has been used to study emotion encoding in body kinematics, minimizing emotion misclassification in exergame scenarios [14].

A commmon limitation of existing approaches to generate motion encoding of emotions (e.g., [11], [13]) is that the presence of emotions is assessed only *qualitatively*. These methods, such as LMA, are subject to personal preferences, style, and movement signature, making emotion classification not always reliable [15]. To the best of our knowledge, we propose the first control architecture for encoding desired emotions in movement kinematics from human demonstra-

tions, that exploits a *quantitative encoding model* to assess the presence of emotion.

## II. Control of Body Movement to Express Emotion

Given a movement by a person, say $H$, another human or computer-based evaluator, say $L$, can observe such movement and decide whether $H$ was experiencing a certain emotion or not while performing it. If $H$ was actually experiencing the emotion *and* this was successfully recognized by $L$, we say that the movement *encodes* the emotion, and that $L$ could *decode* the movement. Given a human movement not encoding a certain emotion, we consider the problem of altering it to produce a motion that encodes the desired emotion (or conversely, altering it to not encode the emotion if the original motion did encode it). Such a motion could then be used as a reference motion for an avatar or robot tele-operated by the human so as for it to express (or not) the target emotion.

Before formalizing the problem mathematically, we review a few useful definitions and notation.

### A. Preliminaries

*Notation:* When applied to vectors, $\|\cdot\|$ denotes the Euclidean norm. $\odot$ denotes the Hadamard product. $\mathcal{B}^{\mathcal{A}}$ denotes the set of functions from set $\mathcal{A}$ to set $\mathcal{B}$.

Let $T \in \mathbb{N}_{>0}$ be the *(maximum) duration* of the movements being considered, $\mathcal{T} \coloneqq \{1, \dots, T\}$ be the corresponding *time window*, $\Delta t \in \mathbb{R}_{>0}$ be a *sampling time*, and $\rho \in \mathbb{N}_{>0}$ be the *number of degrees of freedom* being considered (e.g., $\rho = 3$ if a single point is being sensed in the 3D space). Let $\mathcal{X} \coloneqq (\mathbb{R}^{\rho})^{\mathcal{T}}$ be the sets of movement signals considered, letting $p \in \mathcal{X}$ and $v \in \mathcal{X}$ be the position and velocity associated to the same movement, the constraint holds that

$$v(t) = \frac{p(t+1) - p(t)}{\Delta t}, \quad \forall t \in \{1, \dots, T-1\}. \tag{1}$$

Moreover, we assume that all movements eventually stop; hence, we let $\delta_{\text{vel}} \in \mathbb{R}_{>0}$ be a small threshold, and define the *terminal instant* associated to a movement with velocity $v$ as the time instant $t_{\text{term}}(v) \in \mathcal{T}$ such that $\|v(t)\| \leq \delta_{\text{vel}}$ for all $t \geq t_{\text{term}}(v)$.

We let $\text{dist} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ denote the *distance* between two signals, induced by the $L^2$ norm, that is

$$\text{dist}(x_1, x_2) \coloneqq \sqrt{\sum_{t=1}^{T} \|x_1(t) - x_2(t)\|^2}.$$

Moreover, given a subset of signals $\mathcal{S} \subseteq \mathcal{X}$, we define the *$\mathcal{S}$-projection* operator $\text{proj}_{\mathcal{S}} : \mathcal{X} \to \mathcal{S}$, as[1]

$$\text{proj}_{\mathcal{S}}(x_{\mathcal{X}}) \coloneqq \arg\min_{x_{\mathcal{S}} \in \mathcal{S}} \text{dist}(x_{\mathcal{X}}, x_{\mathcal{S}}). \tag{2}$$

[1]The time needed to compute $\text{proj}_{\mathcal{S}}$ in (2) grows at most linearly with respect to the cardinality of $\mathcal{S}$.

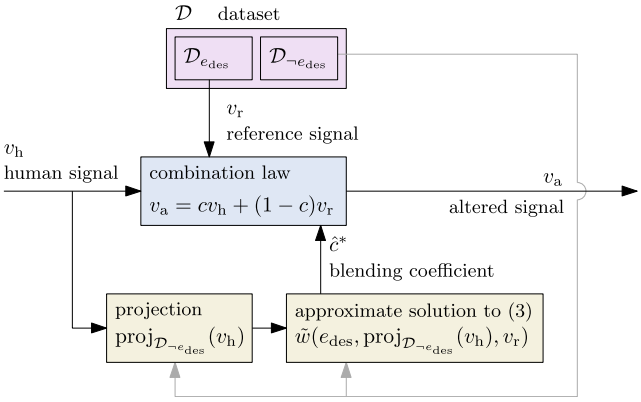


Fig. 1. Block scheme of the proposed solution to solve Problem 3.

### B. Problem statement

We denote by $p_{\text{h}}, p_{\text{a}} \in \mathcal{X}$ and $v_{\text{h}}, v_{\text{a}} \in \mathcal{X}$ the position and velocity signals associated to the *original human motion* to be modified and its *altered* version, respectively.

The *encoding function* $\varepsilon : \mathcal{X} \to \{0, 1\}$ associates to a velocity signal the *encoding level* of that movement, that is 1 if the emotion is encoded and 0 otherwise (note that $\varepsilon$ corresponds to an *encoding model*, as described in [3]). Finally, we let $e_{\text{des}} \in \{0, 1\}$ be a desired value for the encoding level. We aim to solve the optimization problem

$$\min_{v_{\text{a}} \in \mathcal{X}} \ \text{dist}(v_{\text{h}}, v_{\text{a}}), \tag{3a}$$

$$\text{s.t.} \ \ \varepsilon(v_{\text{a}}) = e_{\text{des}}. \tag{3b}$$

The difficulty in solving (3) lies in the size of the decision space $\mathcal{X}$. To overcome this issue and solve (3), we propose a data-driven control strategy represented in Figure 1.

## III. A Data-Driven Solution Framework

We assume the availability of a *dataset* $\mathcal{D} \subset \mathcal{X}$, where each data sample, say $v_{\text{d}} \in \mathcal{D}$, is a human velocity signal with known encoding level $\varepsilon(v_{\text{d}})$ of the target emotion. Given a desired encoding level $e_{\text{des}}$, we further suppose that there exists a non-trivial partition $(\mathcal{D}_{e_{\text{des}}}, \mathcal{D}_{\neg e_{\text{des}}})$ of $\mathcal{D}$, where

$$\mathcal{D}_{e_{\text{des}}} \coloneqq \{v \in \mathcal{D} \mid \varepsilon(v) = e_{\text{des}}\}, \tag{4}$$

$$\mathcal{D}_{\neg e_{\text{des}}} \coloneqq \{v \in \mathcal{D} \mid \varepsilon(v) \neq e_{\text{des}}\}. \tag{5}$$

We then proceed according to the following steps.

1) We train a feedforward neural network to approximate the encoding function $\varepsilon$, using the labelled examples in $\mathcal{D}$, and denote by $\hat{\varepsilon} : \mathcal{X} \to \{0, 1\}$ the resulting approximation.
2) We choose the *reference velocity signal*, say $v_{\text{r}}$, as the signal in the dataset such that
$$v_{\text{r}} = \text{proj}_{\mathcal{D}_{e_{\text{des}}}}(v_{\text{h}}). \tag{6}$$
3) We compute the output transformed velocity signal $v_{\text{a}}$ as the following combination of the velocity signal of the human participant $v_{\text{h}}$ and the reference velocity selected from the dataset. Specifically, we set
$$v_{\text{a}} = cv_{\text{h}} + (1-c)v_{\text{r}}, \tag{7}$$

where $c \in \mathcal{C}$ is a *blending coefficient*, with $\mathcal{C}$ being an appropriate discretization of the interval $[0, 1]$.

4) The position of the modified kinematics is then computed by inversion of (1), enforcing $p_{\mathrm{a}}(1) = p_{\mathrm{h}}(1)$.

Consider now the revised optimization problem

$$\max_{c \in \mathcal{C}} \quad c, \tag{8a}$$

$$\text{s.t.} \quad \hat{\varepsilon}(v_{\mathrm{a}}) = e_{\mathrm{des}}. \tag{8b}$$

In the next Lemma, we show that it is possible to solve (8) in order to solve Problem (3), when (7) is assumed.

**Lemma 1.** *Assume that the encoding function $\varepsilon$ is approximated by some $\hat{\varepsilon}$ such that*

$$\forall v \in \mathcal{X}, \quad \begin{cases} \varepsilon(v) \geq \hat{\varepsilon}(v), & \text{if } e_{\mathrm{des}} = 1, \\ \varepsilon(v) \leq \hat{\varepsilon}(v), & \text{if } e_{\mathrm{des}} = 0. \end{cases} \tag{9}$$

*Then, when $v_{\mathrm{a}}$ is computed from* (7)*, the solution $c^*$ to Problem* (8) *yields a $v_{\mathrm{a}}$ that is optimal for Problem* (3)*.*

*Proof.* Using (7), we have

$$\begin{aligned} \mathrm{dist}(v_{\mathrm{h}}, v_{\mathrm{a}}) &= \|v_{\mathrm{h}} - v_{\mathrm{a}}\| = \|v_{\mathrm{h}} - c v_{\mathrm{h}} - (1-c) v_{\mathrm{r}}\| \\ &= \|(1-c)(v_{\mathrm{h}} - v_{\mathrm{r}})\| = |1-c| \, \|(v_{\mathrm{h}} - v_{\mathrm{r}})\|. \end{aligned} \tag{10}$$

Then, as $v_{\mathrm{r}}$ is selected according to (6), from (10), and recalling that $0 \leq c \leq 1$, it is immediate to see that minimizing $\mathrm{dist}(v_{\mathrm{h}}, v_{\mathrm{a}})$ in (3) corresponds to maximizing $c$. Additionally, for a signal $v_{\mathrm{a}}^*$ that satisfies (8b), constraint (3b) holds as a consequence of (9). Indeed, recall that $\varepsilon : \mathcal{X} \to \{0, 1\}$; if $e_{\mathrm{des}} = 1$, we have $\varepsilon(v_{\mathrm{a}}^*) \geq \hat{\varepsilon}(v_{\mathrm{a}}^*) = e_{\mathrm{des}} = 1$ and thus $\varepsilon(v_{\mathrm{a}}^*) = 1$; on the other hand, if $e_{\mathrm{des}} = 0$, we obtain $\varepsilon(v_{\mathrm{a}}^*) \leq \hat{\varepsilon}(v_{\mathrm{a}}^*) = e_{\mathrm{des}} = 0$ and thus $\varepsilon(v_{\mathrm{a}}^*) = 0$. □

### A. Offline solution

We start by proposing a heuristic solution to Problem (8) that can be used when the human movement signals to be altered have already been acquired and are available offline.

Let $w : \{0, 1\} \times \mathcal{X} \times \mathcal{D}_{e_{\mathrm{des}}} \to \mathcal{C}$ be the (unknown) *solution function*. Given some desired encoding level $e_{\mathrm{des}}$, human velocity signal $v_{\mathrm{h}}$, and reference velocity signal $v_{\mathrm{r}}$, it yields the solution $c^*$ to (8), i.e., such that $w : (e_{\mathrm{des}}, v_{\mathrm{h}}, v_{\mathrm{r}}) \mapsto c^*$.

Define the *restricted solution function* $\tilde{w} : \{0, 1\} \times \mathcal{D}_{\neg e_{\mathrm{des}}} \times \mathcal{D}_{e_{\mathrm{des}}} \to \mathcal{C}$ as a restriction of $w$. We assume the finite cardinalities of $\tilde{w}$'s domain and codomain are small enough for $\tilde{w}$ to be computed by enumeration in reasonable time.[2] The blending coefficient $c$ solving (8) can then be computed as follows. If the input signal $v_{\mathrm{h}}$ already encodes the desired information, i.e., $\hat{\varepsilon}(v_{\mathrm{h}}) = e_{\mathrm{des}}$, then we set $c = 1$ as no modification is needed. Otherwise, if $\hat{\varepsilon}(v_{\mathrm{h}}) \neq e_{\mathrm{des}}$, we obtain $c$ by projecting $v_{\mathrm{h}}$ onto $\mathcal{D}_{\neg e_{\mathrm{des}}}$, computing

$$c = \tilde{w}(e_{\mathrm{des}}, \mathrm{proj}_{\mathcal{D}_{\neg e_{\mathrm{des}}}}(v_{\mathrm{h}}), v_{\mathrm{r}}),$$

which yields the suboptimal blending coefficient solving (3) approximately. As shown below, unlike $w$, $\tilde{w}$ can be computed using $\mathrm{proj}_{\mathcal{D}_{\neg e_{\mathrm{des}}}}$. A block scheme reviewing the strategy is reported in Figure 1.

[2]If this is not the case, i.e., $\mathcal{D}$ is relatively large, an approximator such as a feedforward neural network can be used to approximate $\tilde{w}$.

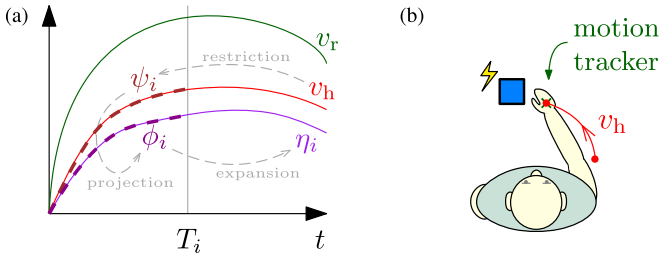


Fig. 2. (a) Example of the trajectories, projections, restrictions, and expansions described in Sec. III-B. (b) Representation from above of the data collection setup described in Sec. V-A.

### B. Online computation

Next, we propose a solution that generates the altered signal $v_{\mathrm{a}}$ in real-time, as the human velocity signal $v_{\mathrm{h}}$ is being measured. To do so, we introduce the *restriction* and *expansion* operators as follows.

For any *reduced time duration* $\tau \in \{1, \dots, T\}$, let $\mathcal{T}_\tau = \{1, \dots, \tau\}$ be a *reduced time window*. We let the *$\tau$-restriction*, denoted by $\cdot|_\tau$, be the restriction of a signal in $\mathcal{X}$ to $\mathcal{T}_\tau$. By extension, for any $\mathcal{S} \subseteq \mathcal{X}$, $\mathcal{S}|_\tau := \{s|_\tau\}_{s \in \mathcal{S}}$ is the set of all signals in $\mathcal{S}$ restricted to $\mathcal{T}_\tau$. Moreover, we define the *$(\tau, \mathcal{S})$-expansion* operator, denoted by $\mathrm{expa}_\tau^{\mathcal{S}} : \mathcal{S}|_\tau \to \mathcal{S}$, yielding the inversion of $\tau$-restriction with respect to $\mathcal{S}$. Namely,

$$\forall \zeta \in \mathcal{S}|_\tau, \quad \mathrm{expa}_\tau^{\mathcal{S}}(\zeta) := s \in \mathcal{S} \text{ s.t. } \zeta = s|_\tau.$$

**Remark 1.** *The operator $\mathrm{expa}_\tau^{\mathcal{S}}$ is well defined if $\mathcal{S}$ does not contain elements whose $\tau$-restriction is the same, i.e., if*

$$\nexists s_1, s_2 \in \mathcal{S} : s_1 \neq s_2, \, s_1|_\tau = s_2|_\tau. \tag{11}$$

*Alternatively, if more than one element in $\mathcal{S}$ is associated to the same restriction, then we define $\mathrm{expa}_\tau^{\mathcal{S}}(\zeta) := s_{\imath^*}$, where the index $\imath^*$ is chosen as $\imath^* := \min \imath(\mathcal{Z})$; $\mathcal{Z}$ being the set of elements mapped to the same restriction, defined as $\mathcal{Z} := \{s \in \mathcal{S} \mid s|_\tau = \zeta\}$ and $\imath : \mathcal{S} \to \mathbb{R}$ an arbitrary indexing function.*

We are now ready to present the online implementation of our proposed control strategy. We assume the blending coefficient is time-varying, and with slight abuse of notation denote it as the function $c : \mathcal{T} \to \mathcal{C}$, and set $c(t) = 0$ for the first $T_0 < T$ time steps. Define $\Delta T \in \mathbb{N}_{>1}$ and the time instants $T_i := T_0 + i\Delta T$, $\forall i \in \mathcal{I}$, where $\mathcal{I} := \{1, \dots, \lfloor \frac{T - T_0}{\Delta T} \rfloor\}$. At each of these instants we measure the current available portion of the human velocity signal, that is $\psi_i := v_{\mathrm{h}}|_{T_i}$ (see Figure 2a). Next, we project $\psi_i$ onto the dataset, obtaining $\phi_i := \mathrm{proj}_{\mathcal{D}_{\neg e_{\mathrm{des}}}|_{T_i}}(\psi_i)$ (c.f. (4)). Finally, we recover the expansion of $\phi_i$ as $\eta_i := \mathrm{expa}_{T_i}^{\mathcal{D}_{\neg e_{\mathrm{des}}}}(\phi_i)$. Now, since $\eta_i \in \mathcal{D}_{\neg e_{\mathrm{des}}}$, it is possible to compute the blending coefficient as

$$c(t) = \tilde{w}(e_{\mathrm{des}}, \eta_i, v_{\mathrm{r}}), \quad \forall t \in \{T_{i-1} + 1, \dots, T_i\}. \tag{12}$$

Moreover, as the whole $v_{\mathrm{h}}$ is unavailable when the first segments of $v_{\mathrm{a}}$ must be generated, we select $v_{\mathrm{r}} = \mathrm{proj}_{\mathcal{D}_{e_{\mathrm{des}}}}(\eta_0)$.

## IV. Enforcing Terminal Conditions

When a movement is carried out to perform some task, often some terminal constraints on the modified kinematics must be

This article has been accepted for publication in IEEE Control Systems Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2024.3416071

fulfilled. This is done so as to match the terminal conditions of the human movement. For instance, when the movement is performed with the goal to reach an object, in Problem (3) we have the additional constraint that the altered movement terminates sufficiently close to the human one.

Then, given the human velocity signal $v_\mathrm{h}$, with position signal $p_\mathrm{h}$, we extend Problem (8) as

$$\max_{c\in\mathcal{C}} \quad c, \tag{13a}$$
$$\text{s.t.} \quad \varepsilon(v_\mathrm{a}) = e_\mathrm{des}, \tag{13b}$$
$$\|p_\mathrm{h}(t_\mathrm{term}(v_\mathrm{h})) - p_\mathrm{a}(t_\mathrm{term}(v_\mathrm{h}))\| \leq \delta_\mathrm{pos}, \tag{13c}$$

where $\delta_\mathrm{pos} \in \mathbb{R}_{>0}$ is a small threshold. Because of the new constraint (13c), in principle there is no guarantee that there exists a solution to Problem (13). Hence, it might be required to relax some of the constraints.

### A. Heuristic solution to the problem of reaching an object

To provide a heuristic solution to Problem (13), we generate the altered velocity signal as

$$v_\mathrm{a} = cv_\mathrm{h} + (1-c)v_\mathrm{r} + v_u, \tag{14}$$

with $v_u \in \mathcal{X}$ being a correction term, given by

$$v_u = k_u \alpha \odot (p_\mathrm{h} - p_\mathrm{a}), \tag{15}$$

where $k_u \in \mathbb{R}_{>0}$ is a control gain, $p_\mathrm{h}(t) - p_\mathrm{a}(t)$ can be understood as the integral error on velocity, and $\alpha : \mathcal{T} \to \{0,1\}^\rho$ yields a vector of boolean variables, used to switch on or off the contribution of the correction term on specific degrees of freedom, and is computed via reinforcement learning.

In particular, following [16], we use a Deep Q-Network approach (DQN), with state $\xi(t) = [p_\mathrm{h}(t)^\mathsf{T} \;\; p_\mathrm{a}(t)^\mathsf{T} \;\; v_\mathrm{h}(t)^\mathsf{T} \;\; v_\mathrm{a}(t)^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{4\rho}$ and action $\alpha(t) \in \{0,1\}^\rho$. The reward function is defined according to the procedure described in [17], which allows to provide a guarantee for the satisfaction of (13c), provided that the cumulative reward obtained is large enough. Namely, let

$$g(t) \coloneqq \begin{cases} 1, & \text{if } \|p_\mathrm{h}(t) - p_\mathrm{a}(t)\|_2 \leq \delta_\mathrm{pos}, \\ 0, & \text{otherwise.} \end{cases}$$

The reward at time $t$ is selected as

$$r(t) = -k_r \|p_\mathrm{a}(t) - p_\mathrm{h}(t)\|_2 - k_\alpha \|\alpha(t)\|_\infty + r^\mathrm{c}(t). \tag{16}$$

where $k_r, k_\alpha \in \mathbb{R}_{>0}$ and

$$r^\mathrm{c}(t) = \begin{cases} r^\mathrm{c}_\mathrm{in}, & \text{if } g(t) = 1, \\ r^\mathrm{c}_\mathrm{exit}, & \text{if } g(t) = 0 \text{ and } g(t-1) = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

where $r^\mathrm{c}_\mathrm{in}, r^\mathrm{c}_\mathrm{exit} \in \mathbb{R}$ are chosen following Algorithm 1 in [17]. In (16), the term containing $\alpha$ is used to avoid applying the correction term $v_u$ when unnecessary. The objective of the reinforcement learning model is to maximize $\sum_{t=1}^{T} \gamma^t r(t)$ (see, e.g., [18]), where $\gamma \in [0,1]$ is the *discount factor*.

During the learning phase, we run episodic tasks selecting each time a different $v_\mathrm{h} \in \mathcal{D}_{\neg e_\mathrm{des}}$ and compute the reference profile $v_\mathrm{r} \in \mathcal{D}_{e_\mathrm{des}}$ as well as the blending coefficient as described in Sections III-A and III-B.

## V. Validation

To validate our AI-based control architecture focused on encoding fear in human movements during a reach-to-grasp phase without actual fear, we define velocity signal $v$ with $\varepsilon(v) = 0$ indicating no fear and $\varepsilon(v) = 1$ indicating fear, setting $e_\mathrm{des} = 1$.[3] Our approach, detailed in Sec. III, utilizes a dataset $\mathcal{D}$ comprising velocity signals both with and without fear ($\mathcal{D}_{e_\mathrm{des}}$ and $\mathcal{D}_{\neg e_\mathrm{des}}$, respectively) for this purpose. Below, in Sec. V-A, we describe the experimental setup used to collect the dataset $\mathcal{D}$; the training of the control architecture being detailed in Secs. V-B, V-C, while numerical validation over a reserved part of the dataset being presented in Sec. V-D.

### A. Data collection

For data collection, we tracked 11 naive participants reaching towards and grasping one of two sensorized cubes of identical size and weight but different colors (one blue, the other yellow). Movements were tracked using a near infra-red optical motion capture system (Vicon Motion Systems Ltd, frame rate 100 Hz); see Figure 2b. To induce fearful responses, participants received an unpleasant electrical stimulation as part of an adaptation to a standard fear conditioning (threat learning) protocol [19]. Specifically, an unpleasant electrodermal stimulation was delivered upon touching one cube (e.g., the yellow one) in 33% of the trials, with the color of the cube causing stimulation counterbalanced among participants.[4] The recorded movement trajectories were classified as encoding fear ($\mathcal{D}_{e_\mathrm{des}}$) when participants reached for the cube paired with stimulation, and as not encoding fear ($\mathcal{D}_{\neg e_\mathrm{des}}$) when participants reached for the other cube. Movements recorded before any stimulation (baseline) were also included in $\mathcal{D}_{\neg e_\mathrm{des}}$. Threat learning was assessed retrospectively with a post-experimental questionnaire, which verified that: 1) participants experienced the unpleasant stimulus (11 participants out of 11), and 2) they recognized the pairing between the stimulus and the cube (10 participants out of 11).

The resulting dataset $\mathcal{D}$ consists of 458 samples, with 197 labeled as encoding fear and 261 as not encoding fear. Each sample records the 3D velocity and position of a participant's wrist at a 100 Hz sampling rate.

### B. Training of the approximate encoding function $\hat{\varepsilon}$

To compute the restricted solution function $\tilde{w}$ as discussed in Sec. III-A, we train a neural classifier to approximate the encoding function $\varepsilon$, where the classifier inputs a velocity signal $v \in \mathcal{X}$ and outputs its encoding level $\varepsilon(v)$. We use a feedforward neural network with an input layer of 60 nodes (representing the 20 points per each axis of the velocity signal

[3] Conversely, removal of fear information is achieved by setting $e_\mathrm{des} = 0$.

[4] The experimental protocol has received approval by the ethical committee (Ethikkommission bei der Ärztekammer Hamburg). All participants received written information about the purpose of the study and the electrodermal stimulation. They were informed that they were allowed to withdraw from the experiment at any time. Written informed consent was obtained by all participants prior to the experimental session. Electrodermal stimulation was well tolerated by participants. No discomfort or adverse effects were reported by participants or noticed by the experimenter during the calibration procedure, nor during or after the experimental session.

This article has been accepted for publication in IEEE Control Systems Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2024.3416071

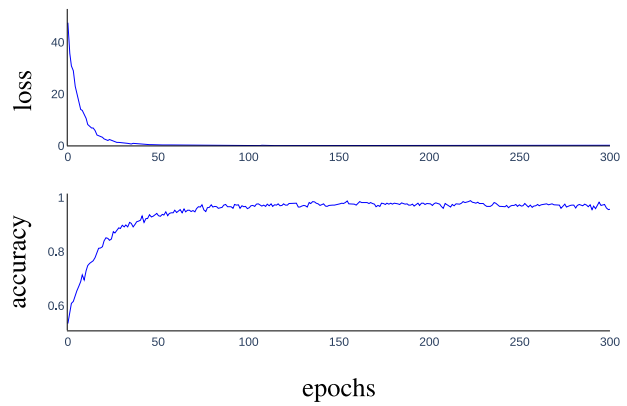


Fig. 3. Relevant quantities concerning the training of the approximate encoding function (see Sec. V-B). Results are computed as an average of the 5 sessions used in cross validation.

available), a hidden layer of 200 neurons with ReLU activation functions, and a sigmoid-activated output layer. Input signals are resampled to 20 points per axis after applying restriction and cubic interpolation. Training employs the Adam optimizer with a learning rate of 0.001 and dropout on the hidden nodes to prevent overfitting. To fully validate, we perform a 5-fold cross-validation with a 70-30 split for training and validation. Training average performances are depicted in Figure 3.

To minimize misclassification's impact, we apply a clipping layer on the classifier's output, categorizing signals as encoding fear for $\varepsilon(v) \in (0.9, 1]$, not encoding fear for $\varepsilon(v) \in [0, 0.1)$, and discarding unclassified signals with $\varepsilon(v) \in [0.1, 0.9]$. The model achieves 87% accuracy for fear signals and 83% for non-fear signals on the validation dataset, with misclassifications at minimum with 3% and 1.5%, respectively.

Post-validation, the network with the highest accuracy on the validation set is selected, identifying two specific sets for final evaluation: $\mathcal{D}^{\text{val}}_{\neg e_{\text{des}}}$ with 65 samples not encoding fear and $\mathcal{D}^{\text{val}}_{e_{\text{des}}}$ with 73 samples encoding fear.

### C. Computation of $\tilde{w}$ and training of the RL module to enforce terminal conditions

Next, we compute $\tilde{w}$ exhaustively (see Sec. III-A). To do so, we set $\mathcal{C}$ to be the discretization of $[0, 1]$, with 50 equally spaced values, including 0 and 1, letting $\Delta c := 1/50$, and follow Algorithm 1. Figure 5 illustrates the computed values of $\tilde{w}$ for all velocity signals in the dataset.

To train the RL agent described in Sec. IV, we run $E = 1100$ episodes, each lasting 200 steps with a DQN algorithm. The DQN approach used consists in a Deep Neural Network made of two hidden layers of 128 nodes with ReLU activation function. The output layer consists of 8 nodes with linear activation function. Furthermore, at the end of each episode, we copy the weights of the neural approximator in a second target network with same structure. We select the discount factor $\gamma = 0.99$, learning rate 0.001 and set random exploration probability to 1 with a decreasing factor of 0.995 applied at the end of each episode. The value of the integral action $k_u$ is set to 6 in (15). Moreover, the reward (16) uses

**Algorithm 1:** Computation of restricted solution function

**Input:** Datasets $\mathcal{D}_{e_{\text{des}}}$ and $\mathcal{D}_{\neg e_{\text{des}}}$; $\mathcal{C} = \{0, \Delta c, 2\Delta c, \dots, 1\}$; approximate encoding function $\hat{\varepsilon}$.

**Output:** Restricted solution function $\tilde{w}$.

```
for v_ne ∈ D_{¬e_des} do
    for v_r ∈ D_{e_des} do
        c ← 1;   is_done ← false;
        while ¬is_done do
            v_a ← c v_ne + (1 − c) v_r;                ▷ Exploiting (7)
            if ε̂(v_a) = e_des ∨ c = 0;                 ▷ e_des = 1
             then
                is_done ← true;   w̃(1, v_ne, v_r) ← c;
            else c ← c − Δc;
```

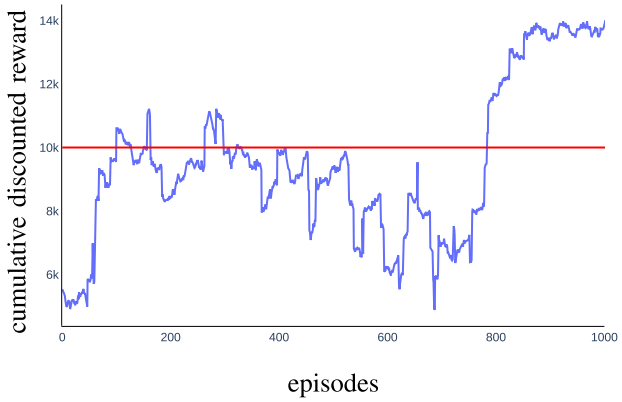


Fig. 4. Moving average of 100-sample cumulative discounted rewards per episode. Red line: threshold value $\boldsymbol{\sigma = 10000}$, defined in [17]. The agent surpasses the threshold after 800 episodes, indicating successful constraint enforcement (13c) on training examples in $\boldsymbol{\mathcal{D}_{\neg e_{\text{des}}}}$.

coefficients $k_{\text{r}} = -0.01$, $k_\alpha = -10$ and the correction terms (17) is shaped to enforce constraint (13c) with $\delta_{\text{pos}} = 20\,\text{mm}$, thus obtaining $r^{\text{c}}_{\text{in}} = 273$ and $r^{\text{c}}_{\text{exit}} = -101953$. Consequently, the evaluation of the fulfillment of condition (13c) is encoded in the discounted cumulative rewards, as explained in [17]. The results of this training procedure is depicted in Figure 4.

### D. Validation of the control strategy

We validate the online control strategy introduced in Sec. III-B using the validation set $\mathcal{D}^{\text{val}}_{\neg e_{\text{des}}}$. The numerical simulation is run with the same sampling time as the motion capture system (see Sec. V), i.e., 0.01 s.

In Figure 6, we showcase an example where a human velocity signal $v_{\text{h}}$, not encoding fear, is transformed into $v_{\text{a}}$ that encodes fear, as per the approximate encoding function $\hat{\varepsilon}$. A more extensive validation, omitted here for the sake of brevity, demonstrated that our approach, when applied to each of the 65 experimentally obtained human movements in $\mathcal{D}^{\text{val}}_{\neg e_{\text{des}}}$ using a blending coefficient computed online, is able to achieve an 89% success rate, with the constraint on the final position (not enforced in this case) being satisfied in 24%. However,

This article has been accepted for publication in IEEE Control Systems Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2024.3416071

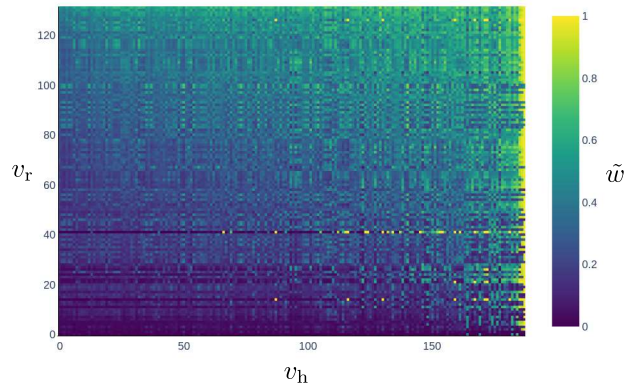


Fig. 5. Values of the restricted solution function $\tilde{w}(\mathbf{1}, \boldsymbol{v}_\mathbf{h}, \boldsymbol{v}_\mathbf{r})$ for $\boldsymbol{v}_\mathbf{h} \in \boldsymbol{\mathcal{D}}_{\neg e_\mathbf{des}}$ and $\boldsymbol{v}_\mathbf{r} \in \boldsymbol{\mathcal{D}}_{e_\mathbf{des}}$ (see Sec. III-A).

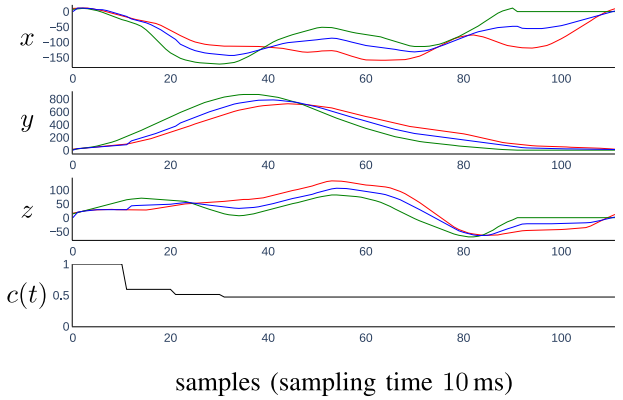


Fig. 6. Representative online alteration of a human motion signal in the 3D space, whose components are denoted as $x$, $y$, $z$. The red line is $\boldsymbol{v}_\mathbf{h}$, the green line is $\boldsymbol{v}_\mathbf{r}$, and the blue line is $\boldsymbol{v}_\mathbf{a}$.

the final distance at end of the preformed movement stays on a average value of 31 mm. Applying solution (14) with an artificially added action, as outlined in Sec. V-C, increased the final position constraint satisfaction to 90%. However, the success rate for classifying "fear" decreased to 65%.

## VI. Conclusions

We presented a data-driven architecture capable of producing information-rich kinematic signals to convey specific emotions. A database of recorded human *reach-to-grasp* kinematics was leveraged to encode emotional information. Additionally, a reinforcement learning module was proposed to enforce a constraint on the terminal position for object grasping.

Validation results demonstrate the effectiveness of our approach. However, there is a clear tradeoff between the accuracy of emotional encoding and adherence to endpoint position constraints. This tradeoff arises because the problem becomes a multi-objective optimization task when additional constraints on the end-effector position are introduced.

We emphasize that our strategy is a first step towards quantifiable emotion encoding: applying the cognitive architecture to enhance avatar and/or robotic motion experimentally will be a future research direction within the EU Research Project Sharespace (`http://sharespace.eu`).

## References

[1] C. Becchio, A. Koul, C. Ansuini, C. Bertone, and A. Cavallo, "Seeing mental states: An experimental strategy for measuring the observability of other minds," *Physics of life reviews*, vol. 24, pp. 67–80, 2018.

[2] J. Podda, C. Ansuini, R. Vastano, A. Cavallo, and C. Becchio, "The heaviness of invisible objects: Predictive weight judgments from observed real and pantomimed grasps," *Cognition*, vol. 168, pp. 140–145, 2017.

[3] E. Scaliti, K. Pullar, G. Borghini, A. Cavallo, S. Panzeri, and C. Becchio, "Kinematic priming of action predictions," *Current Biology*, vol. 33, no. 13, pp. 2717–2727.e6, 2023.

[4] J. P. Gallivan, C. S. Chapman, D. M. Wolpert, and J. R. Flanagan, "Decision-making in sensorimotor control," *Nature Reviews Neuroscience*, vol. 19, no. 9, pp. 519–534, 2018.

[5] J. W. Strachan, A. Curioni, M. D. Constable, G. Knoblich, and M. Charbonneau, "Evaluating the relative contributions of copying and reconstruction processes in cultural transmission episodes," *Plos one*, vol. 16, no. 9, p. e0256901, 2021.

[6] A. Mörtl, T. Lorenz, and S. Hirche, "Rhythm patterns interaction-synchronization behavior for human-robot joint action," *PloS one*, vol. 9, no. 4, p. e95195, 2014.

[7] X. Huang, W. Wu, and H. Qiao, "Connecting model-based and model-free control with emotion modulation in learning systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 8, pp. 4624–4638, 2021.

[8] X. Huang, W. Wu, H. Qiao, and Y. Ji, "Brain-inspired motion learning in recurrent neural network with emotion modulation," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 4, pp. 1153–1164, 2018.

[9] A. Atkinson, W. Dittrich, A. Gemmell, and A. Young, "Emotion perception from dynamic and static body expressions in point-light and full-light displays," *Perception*, vol. 33, no. 6, pp. 717–46, 2004.

[10] J. Llobera, V. Jacquat, C. Calabrese, and C. Charbonnier, "Playing the mirror game in virtual reality with an autonomous character," *Scientific Reports*, vol. 12, no. 1, p. 21329, 2022.

[11] J.-A. Claret, G. Venture, and L. Basañez, "Exploiting the robot kinematic redundancy for emotion conveyance to humans as a lower priority task," *International Journal of Social Robotics*, vol. 9, no. 2, pp. 277–292, 2017.

[12] T. Lourens, R. van Berkel, and E. Barakova, "Communicating emotions and mental states to robots in a real time parallel framework using Laban movement analysis," *Robotics and Autonomous Systems*, vol. 58, no. 12, pp. 1256–1265, 2010.

[13] M. Masuda and S. Kato, "Motion rendering system for emotion expression of human form robots based on laban movement analysis," in *19th International Symposium in Robot and Human Interactive Communication*, 2010, pp. 324–329.

[14] 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.

[15] U. Bernardet, S. Fdili Alaoui, K. Studd, K. Bradley, P. Pasquier, and T. Schiphorst, "Assessing the reliability of the Laban movement analysis system," *PloS one*, vol. 14, no. 6, p. e0218179, 2019.

[16] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[17] F. De Lellis, M. Coraggio, G. Russo, M. Musolesi, and M. di Bernardo, "Guaranteeing control requirements via reward shaping in reinforcement learning," *IEEE Transactions on Control Systems Technology*, 2024, doi: 10.1109/TCST.2024.3393210.

[18] F. De Lellis, M. Coraggio, G. Russo, M. Musolesi, and M. di Bernardo, "CT-DQN: Control-tutored deep reinforcement learning," in *Proceedings of The 5th Annual Learning for Dynamics and Control Conference (L4DC 2023)*, vol. 211. PMLR, 2023, pp. 941–953.

[19] A. Olsson and E. A. Phelps, "Learned fear of "unseen" faces after pavlovian, observational, and instructed fear," *Psychological science*, vol. 15, no. 12, pp. 822–828, 2004.