Applied Intelligence (2026) 56:141
https://doi.org/10.1007/s10489-026-07100-9

Leveraging systems and control theory for social robotics: a model-based behavioral control approach to human-robot interaction

Maria L. Morão Patrício[1] · **Anahita Jamshidnejad**[1,2]

Received: 10 March 2025 / Accepted: 6 January 2026

Abstract

Social robots (SRs) are increasingly expected to assist in healthcare, education, and companionship, thereby addressing the growing need for personalized and affordable health and social care. However, sustaining long-term user engagement remains a major challenge for SRs, largely due to their limited understanding of human mental states. Accordingly, we leverage a recently introduced mathematical dynamic model of human perception, cognition, and decision-making for behavioral control of SRs. By identifying the parameters of this model and deploying it within a model-based behavioral steering system, SRs can autonomously adapt their actions to evolving user mental states, enhancing long-term engagement and personalization. To achieve this, we introduce the first integration of a systems-theoretic cognitive model into a closed-loop predictive behavioral control framework for SRs, formulated as a constrained multi-objective optimization problem that enables transparent, cognition-aware adaptation. In experiments with $\mathbf{10}$ participants interacting with a Nao robot across three chess puzzle sessions ($\mathbf{45}$ to $\mathbf{90}$ minutes each), the identified model achieved a mean squared error (MSE) of $\mathbf{0.067}$ (i.e., $\mathbf{1.675\%}$ of the maximum possible MSE) in tracking beliefs, goals, and emotions of participants, and increased engagement by $\mathbf{16\%}$ ($\boldsymbol{p = 0.009}$) compared to a model-free baseline. Post-interaction participant questionnaires further confirmed the perceived engagement and awareness of the model-based controller. Overall, the framework provides a practical pathway toward SRs that autonomously adapt to users in real time, sustain long-term engagement, and ultimately deliver more effective and personalized assistance in domains such as healthcare, education, and companionship.

Keywords Mathematical dynamic model of mental states · Adaptive cognition-aware social robots · Model-based control

1 Introduction

With the growth and aging of the world population, an increasing number of individuals require targeted assistance. However, health and social care systems often struggle to meet this demand, due to limited resources and financial inaccessibility [1, 2]. Social Robots (SRs), i.e., robots designed to interact with humans in social contexts, offer a promising solution by providing complementary, personalized, and cost-effective assistance to vulnerable users, thereby alleviating the burden on caregivers [1–3].

Over the past decades, SRs have been deployed in healthcare, for example, in post-stroke and gait rehabilitation [4–6], and to assist individuals with cognitive impairment [7, 8]. They have also been deployed in education, both for tutoring and guiding educational games [9–12], and for promoting skill development in children with autism spectrum disorder [13–15]. Furthermore, SRs serve as companions and assistive tools for older adults [16–18].

In assistive contexts, in order to be effective and beneficial, SRs must interact with their users for a prolonged period of time [3, 4, 16, 17, 19, 20]. However, most SRs struggle to keep their users engaged in the long term. This

✉ Maria L. Morão Patrício
m.l.moraopatricio@tudelft.nl

Anahita Jamshidnejad
a.jamshidnejad@tudelft.nl

[1] Control and Operations Department, Delft University of Technology, Kluyverweg 1, Delft 2629 HS, The Netherlands

[2] Intelligent Systems Department, Delft University of Technology, Van Mourik Broekmanweg 5, Delft 2628 XE, The Netherlands

Published online: 09 March 2026

prompts users to abandon using the robots over time [14, 17, 20].

A major challenge in sustaining long-term user engagement lies in the limited social and natural interaction capabilities of current SRs [20, 21].

To address this issue, state-of-the-art SRs typically display human-like behaviors, such as empathy [22], eye contact [13], and creativity [23], with the goal of maintaining user interest over time. While these behaviors can successfully engage users initially, they are often undermined by the novelty effect, i.e., users are temporarily entertained by unfamiliar behaviors but gradually lose interest as these behaviors become familiar through repetition [17, 19].

Consequently, existing approaches fail to sustain long-term engagement. This persistent challenge, recognized for decades yet remaining unresolved, is closely linked to the behavior of SRs, since user engagement over time depends on how varied, responsive, and adaptable those behaviors are [19–21]. This emphasizes the necessity for developing behavioral control approaches for SRs that overcome current limitations in sustaining long-term user engagement.

State-of-the-art controllers of SRs are typically designed for specific applications or tasks. This task-specific design, which stems from underlying behavioral control approaches of SRs, limits their generalizability across contexts [21]. Current methods predominantly rely on rule-based systems, which struggle to handle unforeseen situations, or on machine learning approaches, which, in the SR domain, face data scarcity that forces simplified interactions and constrained behavioral repertoires [4, 14, 24]. As a result, robot behavior often becomes repetitive, predictable, and insufficiently responsive to user input, which can reduce the perceived social presence of the robot and ultimately hinder long-term user engagement [3, 25].

Consequently, this leads to the fundamental question of which aspects of human social interaction SRs fail to replicate. During social interactions, humans naturally model each other's mental states, such as beliefs, desires, and emotions [8]. Furthermore, they dynamically leverage this ability to predict one another's actions and to adjust their own behavior accordingly, a manifestation of Theory of Mind (ToM) [26]. Nevertheless, despite prior work recognizing the potential benefits of such capabilities for SRs, few existing systems comprehensively implement these capabilities [3, 8, 27].

Accordingly, we propose a behavioral control framework for SRs grounded in explicit mathematical modeling of user mental states and their dynamic evolution over time. These models are used to adapt the robot's behavior in real time to optimize interaction-related objectives (e.g., maximize engagement) while respecting constraints (e.g., ensuring a positive emotional state). This approach aims to enable SRs to behave in a more human-like, engaging, and personalized manner, thereby promoting sustainable long-term interactions in Human-Social-Robot Interactions (HSRIs).

The remainder of this paper is organized as follows: Section 1.1 describes the main contributions of this paper, and Section 1.2 reviews relevant work. Section 2 of this article describes the approaches and mathematical changes applied to Mathematical Model of Mind proposed in [28] to make it suitable for HSRIs. A case-study where the framework is applied is given in Section 3, and the results in Section 4. Finally, Section 5 summarizes the conclusions taken in this study and the recommendations for the future.

For convenience, Table 1 lists the abbreviations frequently used throughout this paper.

Table 1 List of abbreviations frequently used in the paper

Abbreviation	Full term
SR	Social Robot
HSRI	Human-Social-Robot Interaction
ToM	Theory of Mind
MMM	Mathematical Model of Mind
MBC	Model-Based Controller
RBC	Rule-Based Controller
RL	Reinforcement Learning
NN	Neural Network
MSE	Mean Squared Error

1.1 Main contributions

In order to advance social robotics and improve the interactions between Social Robots (SRs) and humans, this paper offers the following contributions:

- **First deployment of systems-and-control-theoretic methods for cognitive Human-Social-Robot Interactions (HSRIs)** We leverage, for the first time, a *dynamic, control-theoretic* model of Theory of Mind (ToM) — the Mathematical Model of Mind (MMM) introduced in a recent work [28] — and integrate it into the behavioral control system of SRs. This enables personalization and real-time adaptation of SR behaviors to the evolving mental states of humans, such as beliefs, goals, and emotions. Leveraging MMM for SRs required adapting systems-and-control-theoretic methods to a controlled process that is inherently non-physical and unconventional: human cognition. Unlike physical systems and engineered processes, mental states are not directly measurable, evolve on multiple timescales, and are not governed by deterministic physical laws. Addressing these intrinsic characteristics makes the integration and adaptation of control principles non-trivial and establishes a bridge between traditional systems and control theory and human-centered domains, including social robotics.

- **Development of a model-based behavioral steering system for SRs** We introduce a model-based predictive control framework that optimizes robot actions over anticipated trajectories of user mental states, while respecting constraints that impact the HSRI, such as safety and social norms. The control objective is to maximize key criteria of effective HSRIs, including the engagement and emotional well-being of users, as well as the sustainability of HSRIs. To achieve this, we introduced several technical adaptations bridging classical control theory and social robotics, including: (i) specifying and parameterizing the mathematical formulations of perception, cognition, and decision-making processes of MMM to enable practical deployment; (ii) developing procedures to identify user-specific parameters under limited data; and (iii) embedding the identified model in a model-based behavioral controller. Collectively, these developments address central limitations of rule-based and model-free controllers by enabling more transparent, adaptive, and socially sustainable robot behaviors in real-world HSRIs.
- **Empirical validation through real-life long-term HSRIs** We conducted experiments with 10 participants in extended chess puzzle sessions with a Nao robot. The results demonstrated a 16% increase in the engagement level of users (based on self-reported boredom reduction, $p = 0.009$), as well as reduced intent to quit (i.e., -7.7%, $p = 0.032$) and skip puzzles (i.e., -8.9%, $p = 0.029$). The model also achieved a high estimation accuracy (i.e., mean squared error of 0.067). Post-interaction questionnaires completed by the participants confirmed that perceived personalization and awareness have increased with the robot that leveraged MMM. These results validate the feasibility and effectiveness of cognition-aware closed-loop control for improving the social awareness and responsiveness of SRs in real-world settings.

1.2 Related work

Most state-of-the-art research on steering the behavior of SRs relies on model-free approaches, including learning-based and rule-based systems. Among learning-based methods, Reinforcement Learning (RL) [14, 24, 27, 29, 30] Neural Network (NN) [4, 8, 31, 32], and data-driven approaches [5] have been applied to steer the behavior of SRs.

In social robotics, RL is typically used to learn high-level interaction policies from experience. However, most applications rely on simple algorithms that operate in low-dimensional, discrete state/action spaces and that often produce static or minimally adaptive policies. For example, [29] used RL to personalize the supportive behaviors of a tutor SR, but since the RL policy converged to a single fixed behavior per user, the adaptability of the SR remained very limited. Similarly, [14] applied RL to adjust the difficulty of educational math games and the feedback that was provided by the robot during HSRIs using performance-based rewards. The state space, however, was restricted to the ten predefined games, which in practice led the RL agent to learn one optimal action per game, again limiting its adaptability. Bagheri et al. [24] used RL to select among four empathetic utterance categories based on the user's emotion and personality. However, the state space was limited to four discrete emotion labels combined with a fixed personality type per participant, resulting in limited adaptation during interaction. More recently, more sophisticated RL frameworks were introduced into social robotics. Patacchiola and Cangelosi [27] proposed an actor-critic RL algorithm for trust-based learning, where the inputs consisted of audio and visual features. Meanwhile, [30] developed a batch deep RL algorithm that optimized user engagement by controlling smiling and nodding behaviors of a SR, with the state space consisting of speech features (such as pitch and intensity) extracted from the user's audio. Despite using more advanced RL methods, both approaches relied on measurable cues for state representation and reward design, rather than on explicit models of users' mental states.

The application of NNs in social robotics has been largely restricted to perception tasks rather than behavior control [4, 8, 31, 32]. Filippini et al. [31] used a multilayer perceptron to estimate affective states of users in real time from visible and thermal images, with the NN output feeding into a simple rule set for behavioral control. Similarly, [24] applied an NN-based emotion detection system to generate the state input to the low-dimensional RL agent described in the previous paragraph. Lee et al. [4] combined a feedforward NN with a rule-based module to assess the movement quality of patients. The output of these modules was used by a SR acting as a rehabilitation coach to provide corrective feedback. Finally, [8] relied on off-the-shelf NNs modules for speech-to-text processing and sentiment and intention recognition. Overall, these studies show that NN are primarily used to infer task performance and affective states of users. Nonetheless, the adaptation of SR behavior to these states is typically handled by simple rule-based systems or by machine learning agents, and is rarely grounded in multi-dimensional, dynamic models of user mental state.

More recently, researchers have applied NNs to the generation of robot behavior itself. For example, [33] proposed a natural language interface for home assistance SRs, where a large language model translates user commands into behavior trees that control robot actions. While the system adapts the behavior trees in response to execution failures

and environment constraints, it does not consider or adapt to user mental states. Similarly, [34] developed a transformer-based system that learns social interaction behaviors, such as handshakes, from human-human motion data. In particular, the system generates SR movements by refining its future motions based on the predicted motions of the human partner, enabling natural and coordinated physical interactions. While this system demonstrates the importance of adapting to the physical behavior of users, it does not adapt to their mental states, which is essential in HSRIs.

Additionally, [5] proposed a non-neural data-driven approach that clusters behaviors of human caregivers to generate SR interaction strategies for coaching and therapeutic contexts. In general, most machine learning controllers of SRs neither model the (dynamics of) user mental states nor act in accordance with their evolution over time.

Rule-based approaches are another widely applied control strategy in social robotics [4, 8, 13, 31, 35, 36]. These approaches typically adapt SR behavior based on rules depending on interaction-specific metrics (e.g., task performance [4, 8, 13], user physical state [8, 35], and directly observable user affective states [31, 36]), rather than modeling the mental states of the users or their dynamics. Several studies implemented highly constrained rule sets that specify only a few discrete conditions and corresponding robot behaviors. For example, [13] adjusted game difficulty for children with autism based on fixed performance thresholds, increasing the level when accuracy exceeded 75% and decreasing it when below 25%; [35] regulated the intensity of dance movements proposed to the user according to whether the user's heart rate was above or below a target range; [36] activated one of the four pre-programmed behaviors for a SR based on the age and anxiety level of the user; [31] selected between two actions (i.e., telling a story or singing a song) for a SR depending on the user's engagement. More structured designs include combining a finite state machine with parameterized if-then rules for rehabilitation exercises [4], and adapting the communication style of the SR using a rule-based system informed by variables such as user sentiment, intent, and extraversion [8]. Essentially, rule-based systems remain dominant in practice, but their reliance on fixed or context-specific variables hinders adaptability and provides limited modeling of user mental states.

Several of these studies, including [8, 24, 31], and [36], employ affective computing methods, as they rely on emotion or sentiment recognition to guide robot behavior. However, these affective cues are typically used reactively and fail to capture the temporal evolution or inter-dynamics of affective and cognitive states of users.

Beyond affective computing, cognitive modeling approaches are typically used to generate the robot's behavior through models or architectures inspired by cognitive processes. Within these approaches, one line of research develops biological models of the brain [27, 37], where the SR behavior is generated through a brain-inspired architecture. This line of work has primarily been proposed for enabling SRs to learn trust [27] or to distinguish false beliefs [37].

Similarly, cognitive architectures have been used to structure the behavior of SRs, typically through hierarchical or modular systems that integrate perception, internal states, and task or context knowledge to generate coherent and context-aware actions [8, 11, 38–40]. Nevertheless, these architectures model the internal processes and generate the behavior of the SR, not that of the user.

Some cognitive architectures, such as [8, 11], and [40], have recently incorporated behavioral adaptation to user states, including mental states. For example, [8] proposed a cognitive architecture, intended for SRs assisting older adults, that primarily models robot cognition but includes a reactive component that adjusts robot behavior according to user sentiment, intent, and extraversion. However, these user variables are treated as inputs to the architecture, rather than being explicitly modeled. Similarly, [11] proposed a cognitive architecture for SRs that supported children's learning in problem-solving tasks. The architecture modeled the beliefs, desires, and intentions of the SR while adapting its behavior according to indicators such as the learner's performance and emotions. These indicators were inferred from inputs (e.g., speech, gestures, facial expressions) and incorporated into the robot's beliefs through rule-based processing, rather than by representing the learner as an autonomous agent within the architecture. Cao et al. [40] presented a general-purpose therapeutic architecture that models the robot's affect and behavior, adapting them to the personality, preferences, and mood of users — which were predefined, detected via sensors, or set by a therapist, rather than explicitly modeled within the architecture. Overall, these architectures extensively model the internal cognitive processes of the robot, but treat user-related variables as static inputs. As a result, behavioral adaptation remains reactive to the observed user states, whereas modeling the dynamics of the internal processes and mental states of users enables predictive, context-aware adaptation.

To address this limitation, some researchers have proposed that SRs should implement Theory of Mind (ToM), the capability of rational agents to be aware of and to reason about the perception, beliefs, desires, and decision-making of other rational agents [26, 41]. Scassellati [41] poses that humanoid robots should exhibit ToM, and since then, some studies have applied the concept of ToM for SRs [27, 37, 42–44]. For example, [43] enabled SRs to represent that different individuals may hold distinct and even conflicting

perspectives and beliefs. Additionally, [44] developed two computational frameworks for storytelling interactions, in which the storyteller and listener agents each model and respond to the other's mental states: the storyteller infers and seeks to sustain the listener's engagement, while the listener estimates and influences the storyteller's belief about its engagement. More recently, [37] and [42] explored approaches for equipping SRs with the ability to model and respond to others' false beliefs.

However, these works focus mainly on one particular aspect of ToM, such as belief or engagement, rather than providing a comprehensive model of user cognition or decision-making, leaving a gap in fully understanding and predicting user behavior.

To overcome these limitations, in our earlier work [28], we have proposed a mathematical framework that models the perception, cognition, and decision-making of humans, following the core ToM principles. In this paper, we refer to this model as MMM. In particular, we leverage the MMM to be used by SRs to track and act on the mental states of their human users, by integrating the model within a control system that is suitable for the purposes of HSRIs.

In contrast to existing behavior control methods for SRs, our framework directly addresses the main limitations of these methods and introduces key advantages: It ensures adaptability and personalization by explicitly modeling the user's mental states, including their evolution and inter-dynamics, and by continuously adjusting the behavior of the SR based on this model. It achieves predictivity by optimizing robot actions over anticipated trajectories of user cognition, moving beyond the short-term reactivity typical of affective computing approaches. The generalizability of the framework stems from the domain-independent design of MMM, whereas rule-based and learning-based methods are often tailored to specific setups and are trained on limited datasets. Grounding MMM in psychology and cognitive science enhances data efficiency by embedding prior knowledge about human cognition, which is typically missing in learning-based approaches. Moreover, explicitly modeling user mental states provides transparency and explainability that are often lacking in machine learning approaches, while the control-theoretic foundation ensures robustness by incorporating dynamic constraints, such as safety and social considerations.

2 Methodology

In this section, we describe our proposed model-based control framework for SRs. In particular, we explain how the dynamic mathematical model of perception, cognition, and decision-making introduced in [28], i.e., the Mathematical Model of Mind (MMM),, is leveraged to be used in a model-based control framework for SRs.

In [28], a path diagram, which illustrates variables that are connected via processes and weighted linkages, has been proposed to provide a white-box representation for the inter-dynamics of the (invisible) mental states and their contribution to the (visible) actions of humans, based on the Theory of Mind (ToM). The variables represented in the path diagram can be either static variables (i.e., variables without a memory or impact from their previous realizations) and dynamic variables (i.e., variables with a memory that are impacted by the past realizations). The static variables of the path diagram are real-life data, perceived data, rationally perceived knowledge, intention, and action, and the dynamic variables include beliefs, goals, emotions, perceived knowledge, and biases.

The path diagram is then formulated in [28] as a mathematical model, which we refer to as MMM, using systems theory. MMM includes three modules: perception, which models how real-life data is perceived by humans to impact their mental states; cognition, which represents the dynamic evolution of the mental states of humans; and decision-making, which determines the actions of the humans based on their mental states.

The dynamic processes of MMM have been developed in the state space framework. Thus, they are represented in terms of a set of inputs, outputs, state variables (i.e., dynamic variables that represent the internal mental states at any given time and that evolve according to a differential or difference equation), and dynamic auxiliary variables (i.e., dynamic variables that are additionally included in the model to allow for a realistic and precise bridging of the different modules and state variables). All the dynamic variables (i.e., beliefs, goals, and emotions, which are the state variables, as well as the biases and the perceived knowledge, which are the dynamic auxiliary variables) are updated within the cognition module.

The three modules of perception, cognition, and decision-making are sequentially integrated, i.e., the output variable of the perception module is injected as input into the cognition module, which updates all the dynamic variables of the model. Then, the updated state variables beliefs and goals are injected into the decision-making module, which outputs the action. Note that although the state variable emotion does not directly impact the decision-making processes, it does impact the updated state variables beliefs and goals via their inter-dynamics.

The state space representation of MMM allows the model to be embedded within control-theoretic approaches, which in turn enables steering SRs transparently with guarantees on performance and constraint satisfaction. Next, we describe how the MMM is leveraged for incorporation into

Table 2 List of nomenclature associated with MMM. For readability, the list is divided in four groups: time steps, variables, functions, and parameters and sets

Symbol	Definition
Time steps	
k^{P}	Discrete time step of perception module
k	Discrete time step of cognition and decision-making modules
Variables	
$u_i(k^{\mathrm{P}})$	The i^{th} element of real-life data at perception time step k^{P}
$y_i^{\mathrm{PA}}(k^{\mathrm{P}})$	The i^{th} element of perceived data at perception time step k^{P}
$y_i^{\mathrm{RR}}(k^{\mathrm{P}})$	The i^{th} element of rationally perceived knowledge at perception time step k^{P}
$x_i(k)$	State or dynamic auxiliary variable i at cognition time step k
$y_i^{\mathrm{RIS}}(k)$	The i^{th} element of the intentions at decision-making (or cognition) time step k
$y_i^{\mathrm{RAS}}(k)$	The i^{th} element of the actions at decision-making (or cognition) time step k
Functions	
$f_{ij}^{\mathrm{RR}}(\cdot)$	Function modeling the impact of $y_i^{\mathrm{PA}}(k^{\mathrm{P}})$ on $y_j^{\mathrm{RR}}(k^{\mathrm{P}})$ for every perception time step k^{P}
$f_i(\cdot)$	Function modeling the impact of $x_i(k)$ on $x_i(k+1)$ for every cognition time step k
$f_{ji}^{\mathrm{W}}(\cdot)$	Function modeling the impact of $x_j(k)$ on $x_i(k+1)$ for every cognition time step k
$f_{\ell i}^{\mathrm{Z}}(\cdot)$	Function modeling the impact of $y_\ell^{\mathrm{RR}}(k)$ on $x_i(k+1)$ for every cognition time step k
Parameters and sets	
$\boldsymbol{\theta}_{ij}^{\mathrm{RR}}$	Parameter vector parameterizing sub-process function $f_{ij}^{\mathrm{RR}}(\cdot)$
w_{ii}	Parameter weighing the self-impact of $x_i(k)$ on its evolved value $x_i(k+1)$
$w_{ji}(k)$	Weight representing the impact of $x_j(k)$ on $x_i(k+1)$
$\mathcal{X}_i$	Admissible set of all potential realizations of variable $x_i(k)$
$\overline{\mathcal{X}}_i$ and $\underline{\mathcal{X}}_i$	Subsets of $\mathcal{X}_i$ representing distinct ranges of the influenced variable $x_i(k)$
$\mathcal{X}_{j,m}$	Subsets of $\mathcal{X}_j$ representing distinct ranges of the influencing variable $x_j(k)$; for each subset $\mathcal{X}_{j,m}$ and each subset of the influenced variable ($\overline{\mathcal{X}}_i$ or $\underline{\mathcal{X}}_i$), the influence weight $w_{ji}(k)$ is constant
$\zeta_{j,m}$	Parameter defining $w_{ji}(k)$ for $x_j(k) \in \mathcal{X}_{j,m}$ and $x_i(k) \in \overline{\mathcal{X}}$
$\xi_{j,m}$	Parameter defining $w_{ji}(k)$ for $x_j(k) \in \mathcal{X}_{j,m}$ and $x_i(k) \in \underline{\mathcal{X}}$
θ_i^{RIS}	Neutral realization of intention $y_i^{\mathrm{RIS}}(k)$ (i.e., intention realized when beliefs and goals are zero)
$\theta_{ij}^{\mathrm{RIS}}$	Parameter weighing the influence of the j^{th} element of the beliefs on intention $y_i^{\mathrm{RIS}}(k)$ at cognition/decision-making time step k
$\theta_{i\ell}^{\mathrm{RIS}}$	Parameter weighing the influence of the ℓ^{th} element of the goals on intention $y_i^{\mathrm{RIS}}(k)$ at cognition/decision-making time step k

a control-theoretic method for steering the behavior of SRs. The model is represented in a discrete time framework.

2.1 Leveraging MMM for model-based control of SRs

We introduce improvements to the formulation of MMM [28] that make the resulting model suitable for adoption by control-theoretic methods in real-life implementations of Human-Social-Robot Interaction (HSRI).

Note that, according to the neuroscience findings, although perception, cognition, and decision-making are interconnected, their processes occur with different speeds [45]. More specifically, the perception processes (i.e., the transition of the external stimuli into perceived information) occur almost three times faster than the cognition and motor processes (i.e., the evolution of mental states and their transition into actions) [46]. Moreover, the perception depends on the external stimuli and should be updated every time a new external stimulus is received. However, the cognition evolves not only due to the perception but also under the impact of the inter-dynamics of the mental states. Therefore, the cognition may be updated with a frequency different from the frequency of capturing the external stimuli. Consequently, in the following discussions, we have used different discrete time steps for the perception (where the discrete time steps are indicated by k^{P}) and for the cognition and decision-making (where k denotes the corresponding discrete time steps).

The improvements made in different modules of MMM and the motivation for introducing them are explained next.

For clarity, the main symbols and notation employed in the subsequent discussion are summarized in Table 2.

Perception module Perception is the process of translating real-life data into an internal representation of captured information. Perception in MMM is mathematically modeled with three variables, i.e., real-life data, perceived data, and rationally perceived knowledge, and through two linked sub-processes, perceptual access and rational reasoning. The perceptual access is the transformation of the real-life data into perceived data, which is the data captured by the human. The rational reasoning is the transition of the perceived data into rationally perceived knowledge, which is knowledge that would ideally be reasoned by that individual when not biased by current mental states. Figure 1 illustrates these sub-processes and their input-output pairs.

In order to leverage MMM for model-based control of SRs when used in the context of HSRI, we propose the

following improvements (highlighted in remarks 1 and 2, and elaborated in the text) for the perception module of MMM.

Remark 1 **Specifying the perceptual access mathematically** In [28], the perceptual access sub-process is represented via a generic function $f^{\mathrm{PA}}(\cdot)$ without specification regarding its class or characterization. We fill this gap to enable deployment on SRs by representing the perceptual access as a filter where the output (i.e., perceived data) equals the input (i.e., real-life data) when the input is captured, and, otherwise, equals the last captured value. Our solution presents a transparent way to mathematically represent the perceptual access, enhancing the white-box nature of MMM, and facilitating its usage by SRs.

Suppose that $\mathcal{R}$ is an ordered set which includes all the real-life data in a given HSRI context. At the discrete time step k^{P}, the value of the elements of $\mathcal{R}$ are denoted by $u_i(k^{\mathrm{P}})$ with $i = 1, \ldots, |\mathcal{R}|$. Moreover, $y_i^{\mathrm{PA}}(k^{\mathrm{P}})$, with $i = 1, \ldots, |\mathcal{R}|$, is the i^{th} element of the output of the perceptual access (i.e., i^{th} element of the perceived data) at time step k^{P}, which is determined based on $u_i(k^{\mathrm{P}})$. Whenever the human captures the i^{th} element of real-life data, then $y_i^{\mathrm{PA}}(k^{\mathrm{P}})$ is replaced by the value of this element, $u_i(k^{\mathrm{P}})$. Otherwise, the value of $y_i^{\mathrm{PA}}(k^{\mathrm{P}})$ keeps its most recent value. The mathematical representation of the perceptual access is:

$$y_i^{\mathrm{PA}}(k^{\mathrm{P}}) = \begin{cases} u_i(k^{\mathrm{P}}), & \text{if } u_i(k^{\mathrm{P}}) \text{ was captured} \\ y_i^{\mathrm{PA}}(k^{\mathrm{P}}-1), & \text{otherwise} \end{cases} \tag{1}$$

Remark 2 **Decoupling & parameterizing the function that models the rational reasoning** In [28], the rational reasoning sub-process is represented via a generic function $f^{\mathrm{RR}}(\cdot)$, which remains abstract, without specifying its class or characterization. We address this gap for deployment on SRs, which should understand how humans reason about perceived data, by decoupling and parameterizing the function that maps the input of the sub-process (i.e., the perceived data) to its output (i.e., rationally perceived knowledge). Our solution further enhances the white-box nature of MMM, thus its transparency and traceability, and yields a practical and computationally efficient way for personalizing the perception process to users.

The perceived data is injected as input into the rational reasoning sub-process to generate as output the rationally perceived knowledge (see Fig. 1). Thus, the outputs $y_i^{\mathrm{PA}}(k^{\mathrm{P}})$ of the perceptual access are the inputs to the rational reasoning. The output of the rational reasoning, i.e., rationally perceived knowledge, is composed of elements $y_j^{\mathrm{RR}}(k^{\mathrm{P}})$, where $j = 1, \ldots, |\mathcal{K}|$ and $\mathcal{K}$ is the ordered set that embeds all potential realizations of the rationally perceived knowledge in the given HSRI context.

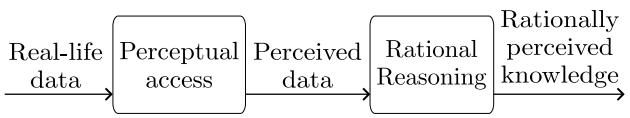


Fig. 1 Perception module

We consider HSRIs that require selective attention from humans (e.g., a particular educational or entertaining joint task by the SR and the human). Since the human focuses on a specific stimulus or input at a time while suppressing the others, each piece of perceived data affects rationally perceived knowledge independently. This allows us to decouple the input-output mapping (i.e., the mapping that shows the impact of input $y_i^{\mathrm{PA}}(k^{\mathrm{P}})$ on output $y_j^{\mathrm{RR}}(k^{\mathrm{P}})$) in the mathematical modeling of the rational reasoning. Considering the additivity condition (which allows for simplicity and efficiency of the computations) on this input-output mapping, we have:

$$y_j^{\mathrm{RR}}(k^{\mathrm{P}}) = \sum_{i=1}^{|\mathcal{P}|} f_{ij}^{\mathrm{RR}}(y_i^{\mathrm{PA}}(k^{\mathrm{P}}); \boldsymbol{\theta}_{ij}^{\mathrm{RR}}), \quad \forall j \in \mathcal{K} \tag{2}$$

In (2), the sub-process is modeled as the summation of various parameterized functions, each mapping one input of the sub-process to its corresponding output. More specifically, the contribution of the i^{th} element $y_i^{\mathrm{PA}}(k^{\mathrm{P}})$ of $\mathcal{P}$ through the rational reasoning process into the j^{th} element $y_j^{\mathrm{RR}}(k^{\mathrm{P}})$ of $\mathcal{K}$ (i.e., the j^{th} element of the rationally perceived knowledge at time step k^{P}) is modeled by the function $f_{ij}^{\mathrm{RR}} : \mathcal{P} \to \mathcal{K}$. Note that the function $f_{ij}^{\mathrm{RR}}(\cdot)$ is defined as parameterized, i.e., $\boldsymbol{\theta}_{ij}^{\mathrm{RR}}$ is a parameter vector that should be identified per user for all indices i and j. This allows for more flexibility in the mathematical model and for the ease of personalization per user.

Remark 3 Sets $\mathcal{P}$, and $\mathcal{K}$ are defined as ordered sets, because the inputs and outputs in (1) and (2) need to be distinguishable to be associated to the corresponding parameterized function of the model.

Cognition module Cognition refers to the processes that use the output of the perception process, i.e., the rationally perceived knowledge, to update the mental states that will determine the decision-making of the human. In [28], using a state space framework, the state and dynamic auxiliary variables (shown by $x(k)$ and distinguished by subscript indices) of the cognitive module are updated at each discrete time step $k+1$, based on their values at the previous discrete time step k. Note that different indices do not only refer to different dynamic variables (e.g., belief versus emotion) but also

to different elements for one variable (e.g., different beliefs that play a role in the given HSRI context). We have:[1]

$$x_i(k+1) = f_i\left(x_i(k)\right) + \sum_{i \neq j} w_{ji}(k) x_j(k) + \sum_{\ell \in \mathcal{K}} z_{\ell i}(k) y_\ell^{\mathrm{RR}}(k) \tag{3}$$

where the first term on the right-hand side of (3) models the impact of the previous realization of the variable, i.e., of $x_i(k)$, on its updated realization, i.e., on $x_i(k+1)$. The second term on the right-hand side of (3) represents the inter-dynamics of the state and dynamic auxiliary variables, with $w_{ji}(k)$ the relative weight showing the impact of variable $x_j(k)$ (called the influencing variable) on updated variable $x_i(k+1)$ (called the influenced variable) at discrete time step *k*. The third term on the right-hand side of (3) represents the influence of those inputs that are external to the cognitive module, with $z_{\ell i}$ the influence of the output of the perception module, the rationally perceived knowledge, on the dynamic variable perceived knowledge (see Figs. 1 and 2).

Remark 4 Note that (2) estimates $y_\ell^{\mathrm{RR}}(k^{\mathrm{P}})$ more frequently than the update frequency of the cognition module. Thus, in (3) $y_\ell^{\mathrm{RR}}(k)$ is the most recent value of $y_\ell^{\mathrm{RR}}(k^{\mathrm{P}})$ that has been estimated prior to time step *k*.

The weights $w_{ji}(k)$ in (3) are, in general, functions of both the influencing and influenced variables at the current time step *k*, as these interactions occur within the cognition module and are state-dependent. In contrast, the weight $z_{\ell i}(k)$ only depends on the influencing variable, since the rationally perceived knowledge acts as an external input to the cognition module, without a closed-loop feedback from the cognitive variables. Accordingly:

$$w_{ji}(k) = f_{ji}^{\mathrm{W}}\left(x_i(k), x_j(k)\right) \tag{4a}$$

$$z_{\ell i}(k) = \begin{cases} f_{\ell i}^{\mathrm{Z}}\left(y_\ell^{\mathrm{RR}}(k)\right), & x_i(k) \text{ is perceived knowledge} \\ 0, & \text{otherwise} \end{cases} \tag{4b}$$

with $f_{ji}^{\mathrm{W}}(\cdot)$ denoting the function that estimates the weights $w_{ji}(k)$ based on the values of the influencing and influenced variables at each time step, and $f_{\ell i}^{\mathrm{Z}}(\cdot)$ denoting the function that estimates $z_{\ell i}(k)$ based solely on the influencing variable, i.e., the rationally perceived knowledge $y_\ell^{\mathrm{RR}}(k)$.

[1] Note that indices used in different sections of the paper (e.g., *i* and *j*) should be treated as local variables, i.e., there is no relevance between these indices used, for instance, in the mathematical discussions for the cognition module and the same indices used elsewhere in the mathematical discussions for the perception or decision-making modules.

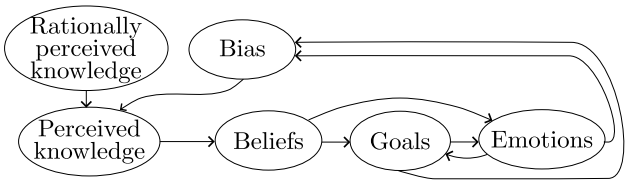


Fig. 2 Cognition module

We propose the following improvements (highlighted in remarks 5 and 6, and elaborated in the text) to leverage the cognition module of MMM for real-life applications in HSRI.

Remark 5 **Incorporating the different update frequencies of state or dynamic auxiliary variables** Not all dynamic variables in MMM are, in general, updated with the same frequency. This difference should thus be incorporated in the update equation (3). Accordingly, we propose to represent the first term, i.e., $f_i\left(x_i(k)\right)$, as a weight w_{ii} multiplied by the most recent realization of the variable, $x_i(k)$. While each weight w_{ii} models how fast a variable responds to its last realization, the relative values of all the weights in (3) incorporate the difference of the update frequencies for different variables.

In MMM [28], the representation of function $f_i(\cdot)$, as well as the relative update frequencies of the dynamic variables were not discussed. As the processes that result in the formation of beliefs[2] are included in the perception module, and the perception processes are faster than the cognition processes [46], the beliefs are in general updated more frequently than the emotions and goals. Accordingly, we propose:

$$f_i(x_i(k)) = w_{ii} x_i(k) \tag{5}$$

By adjusting the value of w_{ii}, one models how intensely the variable $x_i(k)$ changes per discrete time step (i.e., how quickly the variable responds to its previous realization). Moreover, the relative value of w_{ii} and $w_{ji}(k)$ in (3) incorporates the difference in the frequency of the self-impact and the impact by other variables.

For the beliefs, in particular, we use $w_{ii} = 0$ since the beliefs per time step depend on the current stimuli, and are generated by the perceived knowledge (see Fig. 2). Any internal impacts that may affect the beliefs (e.g., the impact of the most recent goals and emotions) are incorporated

[2] Beliefs correspond to the internal representations of stimuli, arising from the perception of a rational agent, and thus vary in time as the perception does. Knowledge (e.g., moral values, political beliefs, etc.) that is fixed or that varies very slowly is deemed pieces of general world knowledge of the agent.

through the impact of the auxiliary variable bias on the perceived knowledge (see Fig. 2)

If the weight w_{ii} tends to 1, it leads to instability of (3) [47]. To ensure the stability of the model, all weights w_{ii} and w_{ji} should be chosen as to assure that all the eigenvalues of the corresponding matrix W (i.e., a matrix with weights w_{ii} in its diagonal element positions (i, i) and weights w_{ij} with $i \neq j$ in positions (i, j)) are within the unit circle [47].

Remark 6 **Modeling the inter-dynamics of mental variables via piecewise-constant functions** In (4a), as is given in [28], $f_{ji}^{\mathrm{W}}(\cdot)$ is presented as generic and abstract, while specific characteristics in formulating $f_{ji}^{\mathrm{W}}(\cdot)$ for meeting desired criteria by MMM remain vague. We address this gap by proposing a mathematical representation for $f_{ji}^{\mathrm{W}}(\cdot)$ that provides a balanced trade-off between computational efficiency and adequate flexibility for illustrating the interpersonal differences of humans, that is always striven for in HSRIs. Accordingly, we propose to define $f_{ji}^{\mathrm{W}}(\cdot)$ as a piecewise function with respect to the influenced variable $x_i(k)$ and influencing variable $x_j(k)$, where these constant values should be identified per human.

Suppose that $\mathcal{X}_i$ and $\mathcal{X}_j$ are the admissible sets of all the potential realizations of the state/auxiliary variables x_i and x_j, respectively. Moreover, $\overline{\mathcal{X}}_i \subseteq \mathcal{X}_i$ and $\underline{\mathcal{X}}_i \subseteq \mathcal{X}_i$, with $\overline{\mathcal{X}}_i \cup \underline{\mathcal{X}}_i = \mathcal{X}_i$ and $\overline{\mathcal{X}}_i \cap \underline{\mathcal{X}}_i = \emptyset$, include two distinct ranges for the influenced variable. For instance, they may be interpreted as ranges of undesirable and desirable status for the variable (e.g., spectrum of feeling bored and spectrum of feeling interested for the emotion state variable "boredom"). We also consider $\mathcal{X}_{j,1}, \ldots, \mathcal{X}_{j,n} \subseteq \mathcal{X}_j$ with $\cup_{m=1}^{n} \mathcal{X}_{j,m} = \mathcal{X}_j$ and $\mathcal{X}_{j,m} \cap \mathcal{X}_{j,o} = \emptyset$ for $m, o = 1, \ldots, n$ and $m \neq o$ as distinct ranges of the influencing variable, where in each of these sub-sets the impact of the influencing variable on the influenced variable may be considered as constant. Then for $x_j(k) \in \mathcal{X}_{j,m}$ with $m = 1, \ldots, n$, we propose the following representation for the function $f_{ji}^{\mathrm{W}}(\cdot)$:

$$w_{ji}(k) = \begin{cases} \zeta_{j,m}, & x_i(k) \in \overline{\mathcal{X}}_i \\ \xi_{j,m}, & x_i(k) \in \underline{\mathcal{X}}_i \end{cases} \tag{6}$$

where $\zeta_{j,m}$ and $\xi_{j,m}$ are fixed parameters that should be identified per user of the SR for all values of the indices j and m. Note that (6) provides a simple, easily implementable, and computationally efficient formulation for real-life applications on the model for SRs, while any potential nonlinearity in the inter-dynamics of the influencing and influenced variables will properly be represented by proper tuning of the number n of subsets $\mathcal{X}_{j,m}$.

In practical implementations, the subsets $\mathcal{X}_{j,m}$ may be defined either based on theoretical thresholds or derived empirically from data. For example, the subsets may correspond to conceptually meaningful ranges, such as low, medium, and high levels of an emotion or belief, or they can be obtained through data-driven partitioning methods, such as quantile-based segmentation or clustering, during model identification. The choice of the relevant approach depends on the availability of domain knowledge and data, as theory-based partitioning offers greater interpretability but requires more design effort.

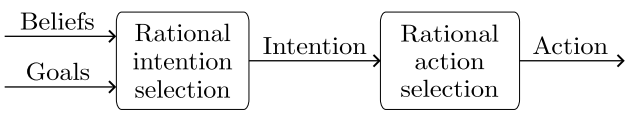


Fig. 3 Decision-making module

Decision-making module Decision-making is the process of deciding about actions, depending on the mental states, namely beliefs and goals. The decision-making process in MMM has been modeled via two sub-processes that are linked through a static auxiliary variable, called the intention. These two sub-processes are the rational intention selection (i.e., the conversion of the beliefs and goals into intentions) and the rational action selection (i.e., the conversion of the intentions into actions). Figure 3 shows a simplified illustration of the decision-making module in MMM.

In order to be able to model the decision-making of humans in real-life HSRIs using MMM, we propose the following improvements (highlighted in remark 7 and elaborated in the text).

Remark 7 **Parameterized formulation of rational intention selection based explicitly on beliefs and goals** In MMM [28], an explicit formulation based on the beliefs and goals for the rational intention selection sub-process has not been provided. We address this gap by proposing an affine parameterized formulation that outputs the values for the intensity of intentions, based on the values realized for the beliefs and goals. This formulation provides a functional way to individually identify the impact of the state variables on the intention, and to estimate their weighted summation to obtain the intentions. This yields an explainable, straightforward, and easily personalizable model for representing the rational intention selection of humans for SRs.

Suppose that $y_i^{\mathrm{RIS}}(k)$ is used to show the i^{th} element of the intention at time step k, with $i \in \mathcal{I}$ and $\mathcal{I}$ a set including all integer indices for the intention elements. In fact, per time step k, the rational intention selection sub-process outputs vector $\boldsymbol{y}^{\mathrm{RIS}}(k)$ that includes $y_i^{\mathrm{RIS}}(k)$ for all $i \in \mathcal{I}$. Element $y_i^{\mathrm{RIS}}(k)$ quantifies the strength of intention i that depends on

the realized values of the beliefs and goals at time step k that influence that intention. For each $i \in \mathcal{I}$ we have:

$$y_i^{\mathrm{RIS}}(k) = \sum_{j \in \mathcal{B}} \theta_{ij}^{\mathrm{RIS}} x_j(k) + \sum_{\ell \in \mathcal{G}} \theta_{i\ell}^{\mathrm{RIS}} x_\ell(k) + \theta_i^{\mathrm{RIS}} \tag{7}$$

where $\mathcal{B}$ and $\mathcal{G}$ are the sets of all integer indices for the beliefs and goals, respectively. Moreover, $\theta_{ij}^{\mathrm{RIS}}$ and $\theta_{i\ell}^{\mathrm{RIS}}$ are parameters that relatively weigh the influence of, respectively, one's beliefs and goals. The parameter θ_i^{RIS} quantifies the (fixed) neutral intention (i.e., the intention that would be realized when beliefs and goals are zero) on the i^{th} intention.

Finally, the rational action selection sub-process receives $\boldsymbol{y}^{\mathrm{RIS}}(k)$ as input, and outputs vector $\boldsymbol{y}^{\mathrm{RAS}}(k)$, which includes all elements of the action taken by the human at time step k. Each element of the action vector is simply a binary variable, with 1 implying that the action is selected, and 0 implying that the action is excluded. The process of transitioning the quantified strength of the intentions into binary actions via rational action selection depends on various practical aspects of the HSRI setup, such as the level of abstraction of the actions (e.g., whether only one or multiple actions are required to fulfill an intention) and any constraints that restrict the actions. In general, this process is defined depending on the HSRI setup.

2.2 Identification of the model

The formulations given in Section 2.1 for MMM show that the model includes parameters that need to be identified (preferably personalized for each user) using approaches based on, e.g., gradient descent [48] or genetic algorithms [49]. These parameters of MMM arise from the theoretical formalization of the model, which is grounded in cognitive and psychological theories, such as ToM [28]. In particular, these theories describe how mental states interact with each other, how they are formed through perception, and how they lead to actions via decision-making processes. The mathematical formulation of MMM expresses these mechanisms through parameterized functions, where the parameters determine how strongly variables influence one another and, in the case of the cognition module, how quickly the affected states respond to those influences. MMM has been presented comprehensively so that it can be adopted in different contexts involving HSRIs. To achieve this, it includes multiple parameters. For a particular context, however, not all the modeled inter-dynamics, and thus parameters, are relevant. Therefore, a crucial consideration in the identification of MMM for a particular context is to avoid redundancy of the parameters, particularly because these parameters mostly correspond to real-life concepts related to the mental processes.

This redundancy particularly occurs when the structure of the model is too complex for the size of the dataset that is available and relevant in a particular context to train the parameters. While performing longer and wider ranges of interactions may allow the collection of more data, this is often not desirable/possible in HSRIs. Moreover, larger datasets do not necessarily guarantee to circumvent the issue, especially when the collected data is not diverse enough for the complexity level of the model.

Hence, based on the context and the concepts that define the realizations of the variables, linkages, and parameters of MMM, one may need to gradually, and systematically, simplify the model structure. While this is context-dependent, we propose two main steps that together allow evaluating and reducing the redundancies in the model every time it is simplified for a given context. This simplification, which includes removing some linkages, variables, and parameters, alongside these two steps can ensure proper identification of the MMM. These steps include parameter redundancy evaluation, given in Algorithm 1, and a multi-stage identification procedure with warm-start, explained via Algorithms 2–4.

Evaluating the presence of redundant parameters in MMM In order to evaluate whether the model needs to be further simplified, the parameters that are unequivocally identifiable in the current structure of MMM are established using Algorithm 1. The MMM will be simplified by gradual, systematic removal of variables and linkages whose corresponding parameters have not been identified unequivocally. This continues until all remaining parameters are unequivocally identifiable.

Note that the identification procedure involves solving generally non-convex optimization problems (i.e., minimizing the error of the values that are generated by the model with regard to the corresponding values collected from real-life HSRIs). Therefore, the values that are returned by the optimizer may correspond to local, rather than global, optima. To circumvent the issue properly, the identification optimization is performed n^{run} independent times, each time with different starting values for the parameters. Then, the n^{opt} (where $n^{\mathrm{opt}} \leq n^{\mathrm{run}}$) sets of parameters that yielded the least realized value for the error function in the validation stage are chosen. In case the standard deviation of the

n^{opt} chosen values per parameter is small enough (i.e., less than a small threshold), the parameter is deemed identified.

Algorithm 1 Procedure to determine the parameters of MMM that are unequivocally identified.

Local variables
r: Index of current identification run
p: Index of parameter
n^{run}: Number of independent identification runs
n^{opt}: Number of optimal identification runs considered in assessment
n^{par}: Number of parameters considered
$\theta_{p,r}$: Identified value of parameter of index p in run r
$\boldsymbol{\theta}_p^*$: Vector of dimension n^{opt} including the identified values of parameter of index p in the n^{opt} runs with minimum optimization cost

Procedure
Run identification procedure n^{run} times, each time with different, random starting values for all n^{par} parameters
$\boldsymbol{r}^* \leftarrow$ indices of n^{opt} runs with minimum optimization cost in the validation stage
for $p \in \{1, ..., n^{\text{par}}\}$ **do**
  $\boldsymbol{\theta}_p^* \leftarrow \theta_{p,r} \quad \forall r \in \boldsymbol{r}^*$
  $\sigma_p \leftarrow$ standard deviation of $\boldsymbol{\theta}_p^*$
  if $\sigma_p \leq$ threshold **then**
    θ_p was unequivocally identified
  else
    θ_p was not unequivocally identified
  end if
end for

Multi-stage optimization with warm-start After collecting measurements for the beliefs, goals, emotions, and actions of a person, for the sake of simplicity, the identification optimization may be decoupled for the perception, cognition, and decision-making modules. By knowing the beliefs, goals, and actions (i.e., the inputs and outputs of the decision-making module, as it is illustrated in Fig. 3), it is straightforward to decouple this module from the other two modules. Contrarily, the perception and cognition modules are less trivial to decouple since the output of the perception module that is then injected into the cognition module is an auxiliary variable, and thus cannot be measured, e.g., by asking people about it in the course of HSRIs.

Therefore, we propose dividing the identification procedure into multiple optimization stages. To achieve this, we introduce two approaches, A and B, to perform the identification optimization. These approaches are given in Algorithms 3 and 4. Both approaches start with a preliminary identification of the parameters of the perception module, given by Algorithm 2, whose outcome is then used as a warm-start. In this case, the perception module of the model is decoupled from the cognition module and is solely considered (called the *decoupled configuration* in Fig. 4). In the *decoupled configuration*, the inputs and outputs are, respectively, real-life data and rationally perceived knowledge. Since the values of the rationally perceived knowledge are not measurable, the pre-identification for the *decoupled configuration* is performed under the assumption that each element of the rationally perceived knowledge for time step k is the same as the corresponding element of the belief for the same time step k (i.e., assuming the bias is null).

Algorithm 2 Generating the warm-start for the parameters of the perception module.

Local variables
r: Index of current identification run
r^*: Index of run with minimum optimization cost
n^{run}: Number of independent runs
$\boldsymbol{\theta}_r^{\text{per}}$: Vector containing identified values for parameters of perception module in run r
$\boldsymbol{\theta}^{\text{per-ws}}$: Vector containing warm-start of parameters of perception module
$y_\ell^{\text{RR}}(k)$: Realized value of element ℓ of rational reasoning at time step k
$x_\ell(k)$: Realized value of element ℓ of belief at time step k
$\mathcal{B}$: Set of all integer indices of beliefs.

Procedure
Assume $y_\ell^{\text{RR}}(k) = x_\ell(k) \quad \forall \ell \in \mathcal{B}$
for $r \in \{1, ..., n^{\text{run}}\}$ **do**
  Randomly initialize $\boldsymbol{\theta}_r^{\text{per}}$
  Optimize $\boldsymbol{\theta}_r^{\text{per}}$ within *decoupled configuration*
end for
$\boldsymbol{\theta}^{\text{per-ws}} \leftarrow \boldsymbol{\theta}_{r^*}^{\text{per}}$ for run r^* with minimum optimization cost

After obtaining a warm-start for the parameters of the perception module, one of the following two approaches is adopted to identify the entire model.

Approach A The perception and cognition modules are simultaneously identified, considering the *coupled configuration* shown in Fig. 4. The starting values in the identification optimization for the parameters of the perception module are those obtained in the pre-identification stage explained above. The starting values for the parameters of the cognition module are randomly selected. Approach A is described in detail in Algorithm 3.

Algorithm 3 Model identification with warm-start - Approach A.

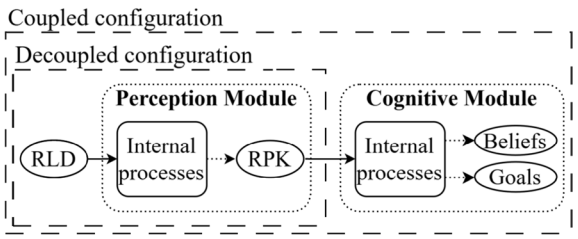


Fig. 4 The two configurations used in the two-stage identification of the ToM model (RLD stands for real-life data and RPK for rationally perceived knowledge). The *coupled configuration* includes both the perception and cognition modules, while the *decoupled configuration* includes only the perception module

Local variables
r: Index of current identification run
n^{run}: Number of identification runs
$\boldsymbol{\theta}^{\text{per-ws}}$: Vector containing warm-start of parameters of perception module obtained in Algorithm 2
$\boldsymbol{\theta}_r^{\text{per}}$: Vector containing identified values for parameters of perception module in run r
$\boldsymbol{\theta}_r^{\text{cog}}$: Vector containing identified values for parameters of cognition module in run r
Procedure
1: **for** $r \in \{1, ..., n^{\text{run}}\}$ **do**
2: $\quad \boldsymbol{\theta}_r^{\text{per}} \leftarrow \boldsymbol{\theta}^{\text{per-ws}}$ (initialization)
3: $\quad$ Randomly initialize $\boldsymbol{\theta}_r^{\text{cog}}$.
4: $\quad$ Optimize $\boldsymbol{\theta}_r^{\text{per}}$ and $\boldsymbol{\theta}_r^{\text{cog}}$ within *coupled configuration*
5: **end for**
6: $\boldsymbol{\theta}_{r^*}^{\text{per}}, \boldsymbol{\theta}_{r^*}^{\text{cog}} \leftarrow \boldsymbol{\theta}_r^{\text{per}}, \boldsymbol{\theta}_r^{\text{cog}}$ for run r with minimum optimization cost

Approach B Before identifying all parameters of the perception and cognition modules together, a pre-identification for the parameters of the cognition module is performed considering the *coupled configuration* in Fig. 4. At this stage, the parameters of the perception module are fixed to the values obtained in the warm-start, and only the parameters of the cognition module are identified (see steps 1–5 of Algorithm 4). Afterward, both perception and cognition modules are warm-started by the values of their pre-identified parameters and a final identification optimization is performed on both modules (see steps 7–12 of Algorithm 4).

Algorithm 4 Model identification with warm-start - Approach B.

Local variables
r: Index of current identification run
n^{run}: Number of identification runs
$\boldsymbol{\theta}^{\text{per-ws}}$: Vector containing warm-start of parameters of perception module obtained in Algorithm 2
$\boldsymbol{\theta}_r^{\text{per}}$: Vector containing identified values for parameters of perception module in run r
$\boldsymbol{\theta}_r^{\text{cog}}$: Vector containing identified values for parameters of cognition module in run r
$\boldsymbol{\theta}_1^{\text{cog}*}, \ldots, \boldsymbol{\theta}_{n^{\text{opt}}}^{\text{cog}*}$: Vectors containing identified values for parameters of cognition module in the n^{opt} runs with the minimum optimization cost until step 6
Procedure
1: **for** $r \in \{1, ..., n^{\text{run}}\}$ **do**
2: $\quad \boldsymbol{\theta}_r^{\text{per}} \leftarrow \boldsymbol{\theta}^{\text{per-ws}}$ (initialization)
3: $\quad$ Randomly initialize $\boldsymbol{\theta}_r^{\text{cog}}$
4: $\quad$ Optimize $\boldsymbol{\theta}_r^{\text{cog}}$ with *coupled configuration*
5: **end for**
6: $\boldsymbol{\theta}_1^{\text{cog}*}, \ldots, \boldsymbol{\theta}_{n^{\text{opt}}}^{\text{cog}*} \leftarrow \boldsymbol{\theta}_r^{\text{cog}}$ of n^{opt} runs with smallest realized values for optimization cost
7: **for** $r \in \{1, \ldots, n^{\text{opt}}\}$ **do**
8: $\quad \boldsymbol{\theta}_r^{\text{per}} \leftarrow \boldsymbol{\theta}^{\text{per-ws}}$ (initialization)
9: $\quad \boldsymbol{\theta}_r^{\text{cog}} \leftarrow \boldsymbol{\theta}_r^{\text{cog}*}$ (initialization)
10: $\quad$ Optimize $\boldsymbol{\theta}_r^{\text{per}}$ and $\boldsymbol{\theta}_r^{\text{cog}}$ with *coupled configuration*
11: **end for**
12: $\boldsymbol{\theta}_{r^*}^{\text{per}}, \boldsymbol{\theta}_{r^*}^{\text{cog}} \leftarrow \boldsymbol{\theta}_r^{\text{per}}, \boldsymbol{\theta}_r^{\text{cog}}$ for run r with optimal optimization cost

On the one hand, approach A has the advantage of being comprised of less steps, which entails less computational load. Moreover, this approach allows for a more flexible search through the parameter spaces, since all parameters are optimized together, without imposing any constraints.

On the other hand, approach B is first constrained with fixed parameters considered for the cognition module. This systematic way of exploration for the optimizer can lead to more parameters being uniquely identified, at the expense of diminishing the flexibility of the search by the optimizer.

2.3 Implementation of MMM

To facilitate the use of MMM, this section summarizes how the model is configured and executed for real-world applications. The implementation involves two complementary procedures: (1) an offline setup, which outlines the procedural steps to prepare and initialize MMM for a specific context, and (2) an online execution, which demonstrates how MMM computes perception, cognition, and decision-making variables per discrete time step during its real-time execution. Together, these two procedures illustrate how the mathematical formulation of MMM presented in Section 2.1 is used in practice, relying on the parameter identification process described in Section 2.2.

The offline setup, presented in Algorithm 5, outlines the procedural steps to configure and instantiate the MMM for a specific context. It begins with defining the variables relevant to the intended application. Then, the mathematical expressions that describe the rational reasoning process (see (2)), the equation describing the rational action selection, and the ranges $\mathcal{X}_{j,m}$ in (6) are specified. Although most components of MMM are already provided in a general parameterized formulation that can be directly applied in practice, as introduced in Section 2.1, the three components mentioned above depend strongly on the context and are therefore tailored at this stage. Finally, the offline setup also includes the identification of the model parameters (detailed in Section 2.2).

Algorithm 5 Offline setup of MMM.

1: For each variable group (e.g., perceived data, beliefs, intentions), declare the variables relevant in the context
2: Specify the functions $f_{ij}^{\text{RR}}(\cdot)$ of the rational reasoning process, introduced in (2)
3: For each cognitive variable, define the set of distinct ranges $\mathcal{X}_{j,m}$ of the influencing variables, in (6);
4: Define equation of rational action selection;
5: Identify parameters of the model, following the algorithms explained in Section 2.2.

The online execution demonstrates how MMM processes one discrete time step during real-time interactions, explicitly linking the sequence of equations across perception (via

(1)—(2)), cognition (via (3)—(6)), and decision-making (via (7). MMM follows a discrete-time state space representation, in which beliefs, goals, emotions, and auxiliary variables evolve as state variables through difference equations. At each step, real-life data is measured, the perception module produces rationally perceived knowledge, the cognition module updates the mental states, and the decision-making module produces the actions of the human, based on those mental states. The procedure is summarized in Algorithm 6 and in Fig. 5.

Algorithm 6 Online execution of MMM.

1: Measure real-life data $u_i(k^{\mathrm{p}})$
2: Execute perceptual access: Compute (1) for $i = 1, \ldots, |\mathcal{R}|$
3: Execute rational reasoning: Compute (2) for $j = 1, \cdots, |\mathcal{K}|$
4: Calculate current realizations of the weights of the cognitive module
$z_{\ell i}(k)$: Compute (4b) for all i and all ℓ
$w_{ij}(k)$: Compute (6) for all i and for all j except $i \neq j$
5: Calculate next realization of each variable $x_i(k+1)$: Compute (3) for all i
6: Execute rational intention selection: Compute (7) for each $i \in \mathcal{I}$
7: Execute rational action selection

In accordance with Remark 4, perception is considered instantaneous with respect to cognition and decision-making. Thus, per update of MMM, perception is executed once (i.e., (1)-(2) are computed), and its output is fed into the cognitive module. Subsequently, the cognitive and decision-making modules are executed. The frequency of this loop corresponds to the internal update rate of MMM, which governs the evolution of the state variables. This internal update rate may differ from the interaction or control frequencies, as these external processes may run at slower rates and therefore sample the output of MMM less frequently than its internal update rate.

2.4 Closed-loop model-based control of SRs

Once the model is mathematically and dynamically formulated, after identifying its parameters for a particular human, the model is embedded within a closed-loop control system for SRs to be used in interactions with the human. Figure 6 shows the architecture of the Model-Based Controller (MBC) system and illustrates how the dynamic ToM model, MMM, is integrated within the control loop.

The MBC system generates a control input $\boldsymbol{u}_k$, which includes all the controlled interactive actions of the robot, such that given criteria for the HSRI are met. In general, satisfying such criteria is associated with optimizing given objectives (e.g., maximizing the interaction time, the user engagement, or satisfaction). To do so, an optimizer may be used in the control loop (see Fig. 6) to suggest candidate control inputs. Note that such a controller, in general, works upon a prediction window, across which candidate control inputs are generated (for the current and all future time steps within the prediction window). In case the size of the prediction window is unity, the MBC system looks only one step ahead (i.e., considers only the impact of the current control input on the HSRI).

In general, every HSRI is subject to various constraints that impact the actions of the SR. For instance, if the SR should offer any joint activities to a person, it should consider the physical and cognitive limitations of the person in making the suggestion. Therefore, the optimization problem of the MBC system is in general a constrained one (see Fig. 6).

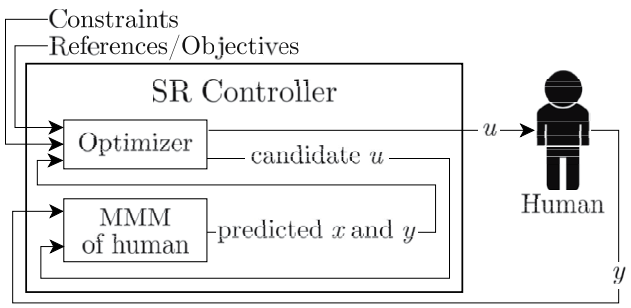


Fig. 6 Block diagram of the model-based controller of the SR using the ToM model

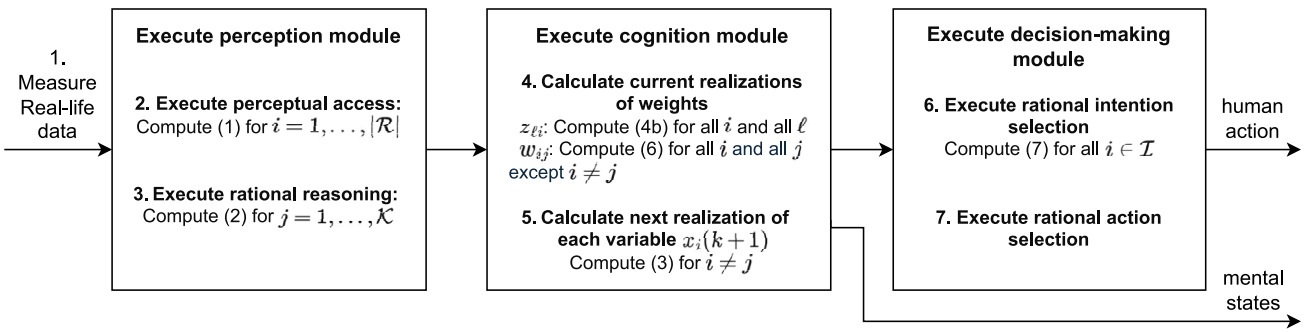


Fig. 5 Overview of the online execution of the MMM. At each discrete time step k, real-life data is measured, the perception module produces rationally perceived knowledge (via (1)—(2)), the cognition module updates the mental states (via (3)—(6)), and the decision-making module generates human actions (via (7))

Subsequently, the candidate control inputs are fed into the dynamic ToM model, which predicts the mental states (shown by x in the figure) of the human as a result of these stimuli (i.e., suggested interactive actions of the SR), as well as the corresponding actions by the human (shown by y in the figure). In case these fulfill the desired objectives and satisfy the imposed constraints, the candidate control inputs are selected for the SR. Otherwise, the optimizer should adjust and propose new candidate control inputs, and the process is repeated.

By following such an approach, it is possible to ensure that the HSRIs comply with pre-defined constraints (e.g., keeping the user engagement above a certain level) and that the interactive actions of the SR result in maximizing the given criteria (e.g., sustaining the HSRIs as long as possible), by systematically incorporating the current (and future) mental states and actions of the user within the control loop.

2.5 Assumptions underlying MMM and the proposed framework

MMM and its integration within an MBC rely on a set of underlying assumptions that delineate the current scope of applicability of the framework and define operational boundaries. Therefore, clarifying these assumptions is essential for setting realistic expectations regarding the performance of MMM and the framework, understanding the trade-offs involved in practical deployment, and identifying adjustments required for broader applicability across diverse users and contexts.

Assumption 1: Boundedness of cognitive variables All cognitive variables of MMM are bounded to the interval $[-1, 1]$ to ensure stability and to provide a shared numerical scale across users. Nevertheless, personalization is preserved because the mapping between the intensity of a mental state and its numerical realization is user-specific. For example, a moderate level of engagement may be represented as 0.4 for one user and 0.6 for another. Similarly, the value 0.4 may not convey the same level of engagement across users, as it reflects their individually identified cognitive dynamics. However, this boundedness may limit the ability of MMM to capture atypical or extreme behaviors. For example, if users exhibit extreme emotional responses that deviate substantially from those observed in the interactions used for model identification, this may lead to realizations of the mental states that fall outside the interval $[-1, 1]$. This limitation reflects the fact that MMM is designed to capture representative cognitive patterns rather than unlikely outliers. If atypical or extreme responses do occur and play a central role in an application, the model can be re-identified to include such dynamics.

Assumption 2: Simplified update rules for state evolution The dynamics of the state variables are represented mathematically using piecewise-linear approximations, rather than fully non-linear formulations (see (6)). This design choice keeps the model computationally tractable and suitable for real-time execution and control. However, it may limit the capacity of MMM to capture highly complex or abrupt cognitive shifts. The impact of this assumption may be mitigated by increasing the number of piecewise regions — supported by the proposed mathematical framework — to better capture the nonlinearity of the dynamics of the state variables. Such refinement, however, entails a trade-off between improved accuracy and higher computational demand of the model.

Assumption 3: Modular structure and decoupling for identification MMM is structured based on distinct modules for perception, cognition, and decision-making processes, as is common in cognitive architectures. To address data scarcity in typical HSRI scenarios, the proposed identification algorithms (explained in Section 2.2) temporarily decouple perception and cognition modules in the first step of the multi-staged identification process (see Algorithm 2). Some parameters of the perception module are pre-identified under a zero-bias assumption (i.e., rationally perceived knowledge and beliefs are considered the same) to provide warm-start values for the subsequent identification process. A final joint optimization stage refines all parameters in the fully coupled configuration, effectively restoring perception-cognition dependencies and improving accuracy. However, in extreme cases – particularly for users with strong emotional biases — the initial warm-start may skew the search space towards local optima. Nevertheless, the identification process inherently mitigates this effect by performing multiple independent gradient-descent runs from different initialization points and by bounding parameter ranges.

Assumption 4: Neurotypical user modeling MMM is grounded in cognitive theories that reflect neurotypical users. When applied to users with different cognitive profiles (e.g., neurodivergent or cognitively impaired individuals), adjustments to some components of the model (such as parameter tuning or minor structural refinements) may be required to better reflect individual cognitive dynamics. Thus, future studies involving neurodivergent or cognitively

impaired users may be needed to refine the model's applicability to these user groups.

Overall, the four assumptions outlined above define the boundaries of MMM in its current representation, while providing a foundation for future refinements and for potential extensions of the framework.

3 Case study

To assess our framework, we carried out an experiment involving HSRIs with 10 volunteer human participants. In these HSRIs, each participant solved multiple chess puzzles while interacting with a Nao robot. Each participant interacted with Nao in 3 sessions of 45 to 90 minutes.

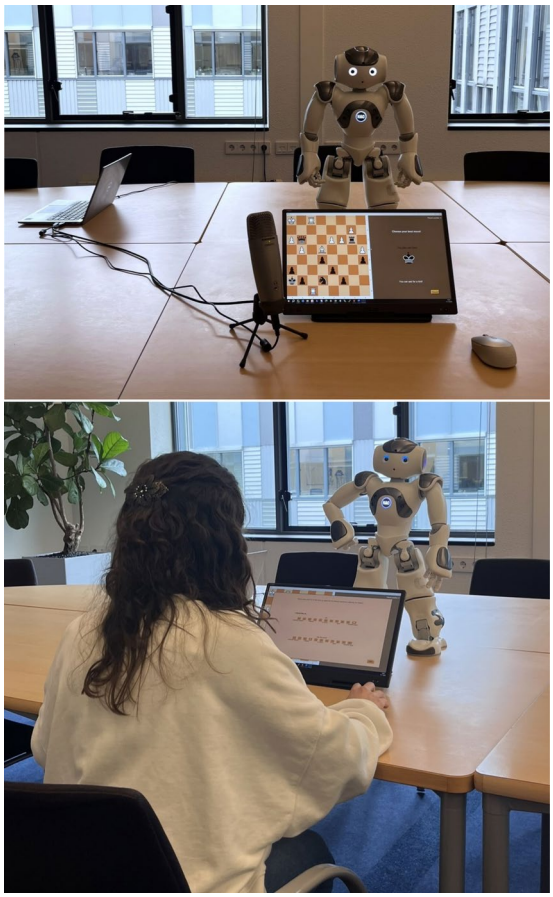


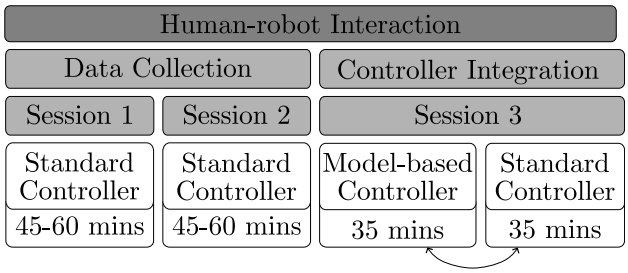
Fig. 7 Experimental setup: The participant sat at a table in the room. The Nao robot was placed on top of that table, facing the participant. A display was placed on the table between Nao and the participant. The display showed the chess puzzles and the questions asked to the participant during the session. A mouse and a microphone were placed on the sides of the display. The microphone and the display were both connected to a laptop that managed the HSRIs and that was connected via Wi-Fi to Nao to steer its behavior via Python

3.1 Experimental setup

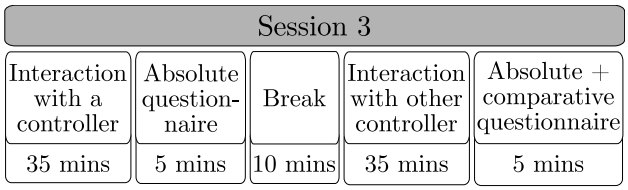
Setup The experimental setup included a Nao robot (a programmable humanoid robot), a display, a microphone, and a mouse, as shown in Fig. 7. In each experiment session, the participant was seated at a table, with the Nao robot placed in front, facing them. A display, which was placed between the user and the robot, showed the chess puzzles, instructions, and questions to the participant during the session. A desktop microphone was placed on the side of the screen to facilitate the verbal communication of the participants with Nao, allowing more robust communication than when the microphones of the robot were used. Finally, a mouse was placed on the other side of the display to enable the participants to play the puzzles. A laptop was connected to both the Nao robot (via WiFi), the display, and the microphone (via a wired connection). This setup enabled the autonomous control of Nao through Python and facilitated data exchange during the sessions. On the third session with each participant, a camera was placed on top of the display pointing to the participant to record their facial expressions and status during the session. This data was used afterward to analyze the engagement of the participants. Finally, the chess puzzles were taken from the *lichess.org* database [50].

Design of interaction sessions The experiment with each participant consisted of 3 sessions, where the details are given in Fig. 8. The first two sessions were designed to collect data about the changes in the relevant mental states of the participant in order to identify the MMM for that participant (personalization). In the third session, the identified model was embedded in the controller of the robot, so that the robot interacted with the participant following the method described in Section 2.4 and based on the personalized MMM (see Fig. 8a).

In the first two sessions, the participants were asked to solve at least 18 chess puzzles or to continue solving puzzles for at least 45 minutes. After playing 30 puzzles or solving puzzles for 60 minutes, the session was ended nonetheless. The participants were in all cases allowed to terminate or quit the session any time earlier, if they wished to. The last session consisted of 2 interactions of 35 minutes each. After each interaction, the participant was given a questionnaire to complete. Moreover, a ten-minute break was given between the two interactions (see Fig. 8b). In one of the two interactions in this session, the participant interacted with a version of Nao that was steered via the MBC embedding the MMM identified for that participant. In the other interaction, Nao was controlled by a model-free, rule-based controller that resembled the current simplified steering systems used for SRs. This controller (specified as the standard controller in



(a) Description of the experiment layout, including the three sessions, their purposes, and durations.



(b) Detailed layout of the third session of the experiment, where the HSRI using our proposed model-based steering framework, including MMM, was assessed.

Fig. 8 Experiment layout including three separate sessions

Fig. 8) was also used to steer the behavior of Nao in the first two sessions designed for data collection. Half of the participants interacted with the standard controller in the first interaction of the third session, while the other half interacted with it in the second interaction of the third session. For each participant, the sessions were at least one week apart from each other, and the entirety of the three sessions was conducted within one month.

Execution of sessions Before the start of the experiment with each participant, the participant was informed about how to interact with Nao and how to play the puzzles via a briefing document. Those participants who had little or no experience with chess could request an extra document that described the basic moves and rules of chess. Then, the experiment started with a brief introduction given by Nao which summarized the most essential points described in the briefing document. Subsequently, the first puzzle showed up on the screen. The debriefing documents can be found in [51].

Each puzzle consisted of a sequence of 2 to 6 most advantageous moves that the participant had to perform correctly. To perform a move, the participant had to click on the desired piece first, and then on the square that the piece was supposed to move to. If the move was correct, the piece was placed in the new square and the next move was made by the program. Otherwise, the move was refused and the participant had to try again until the right move was done. On the right-hand side of the screen, the color that the participant had to play with and the number of puzzles that the participant had completed so far were displayed while the participant was solving a puzzle.

In the first and second sessions, twice per puzzle, the participant was asked to answer questions about their mental states. More specifically, the participant was asked to quantify their beliefs, goals, and emotions relevant for the situation, in a discrete range from 0 to 10, additionally categorized by qualitative terms from "completely disagree" to "completely agree". The questions showed up on the screen, and the participant had to reply by choosing a number using the mouse. These questions are presented in Appendix A.

While playing, participants could verbally ask Nao for a hint, to skip the current puzzle, or to quit the session. Nao would then confirm the request verbally with the participant. When participants asked Nao for the first hint about a puzzle, it provided a general tip about the objective of the puzzle (e.g., the objective was to check-mate the opponent). If the participant asked Nao for a second hint about the same puzzle, it would reveal which piece had to be moved. In response to requesting a third hint, if it concerned the same move, Nao would reveal the optimal move of the piece. If the hint concerned a new move, Nao would again reveal the piece to be moved. When a participant asked to skip the current puzzle, that puzzle disappeared and the next puzzle showed up on the screen. At the end of each puzzle, the participant could request more difficult or easier puzzles, by replying to a question about this that appeared on the screen. All this information was included in the briefing document that was given to the participants at the beginning of the sessions. This document can be found in [51].

Participants of the experiments Ten volunteers aged between 25 and 35 years old participated in the experiment. The participant pool consisted of 30% women and 70% men.

The sample size was determined by the practical constraints of the experimental protocol and the objectives of the study. To obtain multiple and representative measurements for identification and validation of the model under realistic HSRI conditions, each participant completed three sessions of 45 to 90 minutes, spaced at least one week apart. Given the duration and intensity of this multi-session design, recruiting a larger participant pool was not feasible within the scope of this study. Comparable sample sizes are common in early-stage HSRI research, where experiments are time-intensive and primarily aimed at demonstrating feasibility and validating the methodological approach, rather than achieving broad statistical generalization.

Role of Nao Besides giving general indications about the chess puzzles to the participants and socially interacting with the participants, Nao also decided the difficulty level of the puzzles and whether to give rewards to the participants, in order to keep them engaged.

Regarding the difficulty level of the puzzles, six levels of difficulty were defined based on the ratings of the puzzles given in the *lichess.org* database [50] (where the rating of a puzzle is a measure of its difficulty). For each difficulty level, a range of the rating was considered and the puzzles with a rating in that range were included. The ratings that corresponded to each difficulty level considered in this case study can be found in [51].

To give rewards to the participants, Nao performed one of the following entertaining movements or gestures lasting around 10 to 30s: pretending to play a guitar (posing as if the robot was holding a guitar, but there was no real guitar in the room), dancing, performing tai-chi, pretending to be an elephant, pretending to take a photo of the participant (with an imaginary camera). These movements are presented in Appendix B.

During the first two sessions, the actions of Nao were carefully scheduled as to instigate different beliefs, goals, and emotions for the participants, and to provide a diverse dataset to be subsequently used to identify the MMM per participant. Thus, to stimulate the mental states as widely as possible, and to capture the dynamic effects of increasing or decreasing the difficulty level of the puzzles when the participant was in different mental states, we distributed the difficulty levels of the puzzles to include both increasing and decreasing sequences. The difficulty level was kept identical for three puzzles in a row to ensure that the data gathered truly captured the effects of solving puzzles of that difficulty for the participant. Then this difficulty level would increase or decrease with a step size of 2. More specifically, in the first session, the sequence of the levels of difficulties was $\{0, 2, 4, 4, 2, 0\}$, meaning that the difficulty level started at 0 (for three consecutive puzzles), increased to 2 (also for three consecutive puzzles), and then to 4. After playing two groups of three puzzles of difficulty 4, the difficulty level decreased to 2, and then to 0. In the second session, the sequence of the levels of difficulties was $\{5, 3, 1, 1, 3, 5\}$. In these sessions, if the number of puzzles that the participants solved surpassed 18 puzzles, then the difficulty level of puzzle number $18 + n^{\text{puzzle}}$, with n^{puzzle} an integer, was the same as the difficulty level of puzzle n^{puzzle} (taking into account that the maximum number of puzzles offered per session was 30).

As for giving rewards to the participants, in the first two sessions, Nao randomly performed one celebratory movement or gesture for each set of three puzzles with the same difficulty level. Note that the main objective of this design choice was to capture the dynamic effects of giving a generally encouraging/entertaining reward on the mental states of the participants, rather than the specific design of such rewards.

Measurement methods Three complementary measurement methods were used in this case study: (1) self-reported feedback during interactions, (2) post-interaction questionnaires, and (3) automated video analysis. Self-reports were used to obtain the ground truth of user mental states, whereas all three methods contributed to assessing user engagement and overall interaction quality.

The first method — self-reported feedback during interactions — consisted of participants providing self-reported ratings of their mental states (i.e., beliefs, goals, emotions, and intentions) at regular intervals upon request during the interaction sessions. Responses were recorded on a discrete $0 - 10$ scale with qualitative anchors ranging from completely disagree to completely agree (see Appendix A for the complete list of questions).

In the first two sessions, self-reports were collected twice per puzzle and after hint requests, providing the data required for identifying and validating the MMM parameters. In the third session, feedback was requested if more than 150s had passed since the last feedback request or every three puzzles. In this session, the responses were used both to reinitialize the user's mental states estimated by MMM, which are required by the controller, and to assess the quality of the ongoing interaction between the participant and Nao. To minimize unreliability and inconsistency, participants were briefed on the importance of honest responses, and the questionnaire combined numerical scales with qualitative anchors. The frequency of the feedback requested from participants was carefully selected to balance the need for sufficient data for model identification against the risk of distracting participants or causing fatigue, which could otherwise compromise response quality.

The second method consisted of post-interaction questionnaires. In the final session, after each of the two interactions, participants completed a standardized 7-point Likert-scale questionnaire assessing their engagement level and their perceived robot awareness, adaptability, and personalization (available in the experimental dataset [51]).

These questionnaires provided a complementary subjective evaluation of engagement and interaction quality.

During preliminary testing, a scale drift was observed between the two questionnaires, which were administered approximately 45 minutes apart, suggesting that the time gap between administrations may have affected the consistency of responses. Therefore, a comparative questionnaire

was added at the end of the third session to allow the participants to directly compare the two interaction conditions, after first completing the independent absolute questionnaires for each condition. This approach enhanced the reliability and interpretability of the questionnaire results.

Finally, the third method — an automated, computer-vision-based engagement detection tool, presented in [52] — was additionally used in the final session as a complementary, non-intrusive measurement of engagement that did not rely on user self-reports. This tool estimated engagement levels of participants on a continuous scale from 0 (no engagement) to 1 (full engagement). However, since the engagement detection tool had been trained on robot-centric interactions with a single focus point, whereas participants in our setting alternated attention between the robot and the puzzle screen, its application involved a multi-focus interaction context differing from the original training conditions. Accordingly, the results obtained from this method are reported together with their statistical significance and are interpreted with caution in Section 4.

The integration of multiple complementary measurement methods in our experiments enabled cross-validation of the model and controller, ensuring a reliable evaluation of the proposed framework.

3.2 Model-based controller for Nao

The MBC steering system of Nao used in the third session of this case study had to decide, at the end of each puzzle, about the level of difficulty of the next puzzle and about whether to give a reward to the participant. This decision had to optimize the mental states of the participant according to a cost function that depended on the model identified for each participant. Next we describe the details for designing the MBC steering system of Nao.

Table 3 Concepts that represent the real-life data and perceived data of the perception module for the case study

Real life data	Perceived data (corresponding to the real-life data in the same row of this table)	Concept	Type
$u_1(k^{\mathrm{P}})$	$y_1^{\mathrm{PA}}(k^{\mathrm{P}})$	Puzzle difficulty level	Integer
$u_2(k^{\mathrm{P}})$	$y_2^{\mathrm{PA}}(k^{\mathrm{P}})$	Number of hints	Integer
$u_3(k^{\mathrm{P}})$	$y_3^{\mathrm{PA}}(k^{\mathrm{P}})$	Number of wrong attempts	Integer
$u_4(k^{\mathrm{P}})$	$y_4^{\mathrm{PA}}(k^{\mathrm{P}})$	Time to solve a puzzle	Continuous
$u_5(k^{\mathrm{P}})$	$y_5^{\mathrm{PA}}(k^{\mathrm{P}})$	Skipped puzzle?	Boolean
$u_6(k^{\mathrm{P}})$	$y_6^{\mathrm{PA}}(k)$	Reward given?	Boolean

3.2.1 MMM for participants

One of the crucial elements of an MBC steering system is a model of the process (i.e., the perception, cognition, and decision-making of the participants) that should be impacted in desired ways by the designed steering system. This section describes the mathematical modeling of the perception, cognition, and decision-making modules for the participants of this case-study, based on the MMM detailed in Section 2.1. A graphical representation of each module is given in Appendix C.

Perception module The perception module, as explained before (see Fig. 1), is in general composed of perceptual access and rational reasoning sub-processes. Table 3 shows the pieces of real-life data and perceived data that are relevant for and used in the case study. The list of the pieces of rationally perceived knowledge is given in Table 4. In this case study, due to the nature and setup of the experiments, it is sensible to assume that the participants perfectly observe all relevant pieces of real-life data. Thus, based on (1), for all perception time steps k^{P}, for the perceptual access we have $y_i^{\mathrm{PA}}(k^{\mathrm{P}}) = u_i(k^{\mathrm{P}})$, where y_i^{PA} is a piece of perceived data (i.e., an output from the perceptual access sub-process) and $u_i(k^{\mathrm{P}})$ is its corresponding input real-life data. As given in Table 3, for this case study, $i = 1, \ldots, 6$. To model the rational reasoning sub-process, which is formulated via (2) in MMM, the relationships between all pieces of the perceived data $y_i^{\mathrm{PA}}(k^{\mathrm{P}})$ for $i = 1, \ldots, 6$ (i.e., the inputs to the sub-process) and the corresponding piece of rationally perceived knowledge $y_j^{\mathrm{RR}}(k^{\mathrm{P}})$ (i.e., the outputs of the sub-process) should be defined by providing relevant expressions for function $f_{ij}^{\mathrm{RR}}(\cdot)$ in (2). Fig. 14 illustrates the pieces of perceived data that influence each rationally perceived knowledge, where $y_1^{\mathrm{RR}}(k^{\mathrm{P}})$ in Table 4 is influenced by $y_1^{\mathrm{PA}}(k^{\mathrm{P}}), \ldots, y_5^{\mathrm{PA}}(k^{\mathrm{P}})$ given in Table 3, and $y_1^{\mathrm{RR}}(k^{\mathrm{P}})$ is influenced by $y_6^{\mathrm{PA}}(k^{\mathrm{P}})$. In order to model function $f_{ij}^{\mathrm{RR}}(\cdot)$ to mathematically represent the processes that are illustrated in Fig. 14, three types of mathematical expressions were considered, i.e., affine (8a), exponential (8b), and boolean (8c). The choice of each expression depended on the nature of the input variable $y_i^{\mathrm{PA}}(k^{\mathrm{P}})$ as is explained further when the particular use cases of these functions are discussed. The expressions for these functions are:

$$f_{ij}^{\mathrm{RR,F}} \quad \begin{aligned} &\left(y_i^{\mathrm{PA}}(k^{\mathrm{P}}); \theta_{ij,0}, \theta_{ij,1}\right) = \\ &= \theta_{ij,0} y_i^{\mathrm{PA}}(k^{\mathrm{P}}) + \theta_{ij,1} \end{aligned} \tag{8a}$$

$$f_{ij}^{\mathrm{RR,E}} \quad \begin{aligned} &\left(y_i^{\mathrm{PA}}(k^{\mathrm{P}}); \theta_{ij,0}, \theta_{ij,1}\right) = \\ &= \theta_{ij,1} \exp\left(\theta_{ij,0} y_i^{\mathrm{PA}}(k^{\mathrm{P}})\right) + 1 \end{aligned} \tag{8b}$$

Table 4 Concepts that represent the rationally perceived knowledge, perceived knowledge, beliefs, goals, emotions, and biases related to the cognitive module for the case study. Note that the number of the relevant beliefs, goals, emotions, and biases involved in the experiment are, respectively, 2, 4, 2, and 1

Rationally perceived knowledge	Perceived knowledge (corresponding to the rationally perceived knowledge in the same row of this table)	Beliefs (corresponding to the perceived knowledge in the same row of this table)	Concept
$y_1^{\mathrm{RR}}(k)$	$x_1^{\mathrm{PK}}(k)$	$x_1^{\mathrm{b}}(k)$	The puzzle is difficult
$y_2^{\mathrm{RR}}(k)$	$x_2^{\mathrm{PK}}(k)$	$x_2^{\mathrm{b}}(k)$	Nao offered a reward
Goals			Concept
$x_1^{\mathrm{g}}(k)$			Quit the session
$x_2^{\mathrm{g}}(k)$			Skip the puzzle
$x_3^{\mathrm{g}}(k)$			Get help
$x_4^{\mathrm{g}}(k)$			Change difficulty
Emotions			Concept
$x_1^{\mathrm{e}}(k)$			Boredom
$x_2^{\mathrm{e}}(k)$			Frustration
Bias			Concept
$x^{\mathrm{bias}}(k)$			The puzzle is difficult

$$f_{ij}^{\mathrm{RR,B}}\left(y_i^{\mathrm{PA}}\left(k^{\mathrm{p}}\right);\theta_{ij,0},\theta_{ij,1}\right)=\begin{cases}\theta_{ij,0}, & y_i^{\mathrm{PA}}\left(k^{\mathrm{p}}\right)=0\\ \theta_{ij,1}, & y_i^{\mathrm{PA}}\left(k^{\mathrm{p}}\right)=1\end{cases} \tag{8c}$$

We have selected the affine relationship (8a) to estimate the impact of the inputs that are integer, with an upper bound. These inputs include *puzzle difficulty level*, $y_1^{\mathrm{PA}}(k^{\mathrm{p}})$, and *number of hints*, $y_2^{\mathrm{PA}}(k^{\mathrm{p}})$, both being integers with an upper bound of, respectively, 5 and $1+2n^{\mathrm{moves}}$ where n^{moves} is an integer that represents the number of moves in the puzzle. For the inputs of the rational reasoning sub-process that are integers but have no upper bound, i.e., for *number of wrong attempts*, $y_3^{\mathrm{PA}}(k^{\mathrm{p}})$, and *time to solve a puzzle*, $y_4^{\mathrm{PA}}(k^{\mathrm{p}})$, the impact on the output of the sub-process was modeled using an exponential function, as given in (8b). Finally, for those inputs to the sub-process that are of a Boolean nature, i.e., for *skipped puzzle?*, $y_4^{\mathrm{PA}}(k^{\mathrm{p}})$, and *reward given?*, $y_5^{\mathrm{PA}}(k^{\mathrm{p}})$, the impact was modeled using (8c). The first output $y_1^{\mathrm{RR}}(k^{\mathrm{p}})$ of the rational reasoning sub-process, i.e., *the puzzle is difficult*, is generated by all first 5 pieces of inputs from the perceived data indicated in Table 3, i.e., by $y_1^{\mathrm{PA}}(k^{\mathrm{p}}),\ldots,y_5^{\mathrm{PA}}(k^{\mathrm{p}})$, as is shown in Fig. 14. The second output $y_2^{\mathrm{RR}}(k^{\mathrm{p}})$, i.e., *Nao offered a reward*, is generated solely by $y_6^{\mathrm{PA}}(k^{\mathrm{p}})$, i.e., by the perceive data *reward given?* (see Table 3). Using (2) and (8a) we have:

$$\begin{aligned}y_1^{\mathrm{RR}}(k^{\mathrm{p}}) &= \theta_{11,0}y_1^{\mathrm{PA}}(k^{\mathrm{p}})+\theta_{11,1}+\\ &\theta_{21,0}y_2^{\mathrm{PA}}(k^{\mathrm{p}})+\theta_{21,1}+\\ &\theta_{31,1}\exp\left(\theta_{31,0}y_3^{\mathrm{PA}}(k^{\mathrm{p}})\right)+1)+\\ &\theta_{41,0}\exp\left(\theta_{41,1}y_4^{\mathrm{PA}}(k^{\mathrm{p}})\right)+1)+\\ &f_{51}^{\mathrm{RR,B}}\left(y_5^{\mathrm{PA}}\left(k^{\mathrm{p}}\right);\theta_{51,0},\theta_{51,1}\right)\end{aligned} \tag{9a}$$

$$y_2^{\mathrm{RR}}(k^{\mathrm{p}}) = f_{61}^{\mathrm{RR,B}}\left(y_6^{\mathrm{PA}}\left(k^{\mathrm{p}}\right);\theta_{61,0},\theta_{61,1}\right) \tag{9b}$$

Remark 8 Note that in (9a) the term $\theta_{11,1}+\theta_{21,1}+f_{51}^{\mathrm{RR,B}}\left(y_5^{\mathrm{PA}}\left(k^{\mathrm{p}}\right);\theta_{51,0},\theta_{51,1}\right)$ may be identified as one single parameter.

Cognition module The cognition module (see Fig. 2) is composed of three main state variables, i.e., beliefs, goals, and emotions. In the setup of this case study, the number of beliefs, goals, and emotions involved is, respectively, 2, 4, and 2, where the corresponding concepts have been given in Table 4. The realization of these variables are real values which can vary between -1 and 1. Figures 15 and 16 show the structure of the cognitive module, including variables and linkages. Figure 15 corresponds to the complete model used in the first and second sessions. Following the strategy described in Section 2.2, a simplified version of the previous model was used to integrate the model-based control session. Figure 16 represents this simplified model.

To generate the perceived knowledge, (3) is used. The first term of this equation is null (i.e., $f_i(x_i(k))=0$), and $z_{\ell i}(l)=1$. As shown in Fig. 2, the second term of this equation reflects the influence of the biases.

In order to update the beliefs, goals, and emotions, (3)–(6) are used, with weight w_{ii} set to 0 for beliefs and to 0.9 for both goals and emotions. The weights $w_{ji}(k)$ are given by:

$$w_{ji}(k)=\begin{cases}w_{ji}^{-}, & x_j(k)\le 0\\ w_{ji}^{+}, & x_j(k)>0\end{cases} \tag{10}$$

where the parameter values w_{ji}^{+} and w_{ji}^{-} should be identified for each participant.

Remark 9 To assess the validity of the hypothesis that the goals and emotions are updated with a lower frequency than the beliefs, as stated in Section 2.1, in addition to considering the values indicated above for w_{ii}, an additional identification procedure for the MMM has been conducted for all participants of the case study using $w_{ii} = 0$ for both the beliefs and the goals. The estimation errors have been shown and compared in Section 4, and the validity of the hypothesis has been discussed there.

Remark 10 Since the biases are auxiliary variables used to describe the effect of emotions on new beliefs, we represent them without dynamics.

Decision-making module The decision-making module (see Fig. 16) is composed of two main sub-processes, rational intention selection and rational action selection. Table 5 includes the concepts that define the intentions and actions relevant for the decision-making module in this case study.

Regarding the rational intention selection, the intensity of each intention y_i^{RIS} is given by (7). In the context of the case study, each intention is only influenced by one goal and is not influenced by any belief, yielding a particular case of (7). Table 5 lists the goal that influences each intention, where the 4 goals, described earlier in this section, result in 5 possible intentions. This information is also graphically presented in Fig. 17.

As for the action selection, we assume that each action is performed if the intention corresponding to the same concept is positive (and not performed otherwise). Hence, the rational action selection is mathematically represented as:

$$y_i^{\mathrm{RAS}}(k) = \begin{cases} 0, & y_i^{\mathrm{RIS}}(k) \leq 0 \\ 1, & y_i^{\mathrm{RIS}}(k) > 0 \end{cases} \tag{11}$$

Table 5 Concepts that represent the intentions and actions of the decision-making for the case study.

Intention	Action	Concept	Cognitive variable that influences the intention
y_1^{RIS}	y_1^{RAS}	Quit the game	$x_1^{\mathrm{g}}(k)$: Quit the game
y_2^{RIS}	y_2^{RAS}	Skip the puzzle	$x_2^{\mathrm{g}}(k)$: Skip the puzzle
y_3^{RIS}	y_3^{RAS}	Ask for help	$x_3^{\mathrm{g}}(k)$: Get help;
y_4^{RIS}	y_4^{RAS}	Ask for an easier puzzle	$x_4^{\mathrm{g}}(k)$: Change difficulty
y_5^{RIS}	y_5^{RAS}	Ask for a more difficult puzzle	$x_4^{\mathrm{g}}(k)$: Change difficulty

Measurement frequency The MBC should determine whether or not the participant receives a reward from Nao, as well as the difficulty level of the next puzzle, every time a puzzle is finished. Since the measurements of the cognitive variables are gathered by asking the participants about their mental states (see Appendix A), there needs to be a trade-off between collecting a sufficient number of measurements and not disturbing or distracting the participants, as it may affect the reliability of the measurements. Therefore, to balance the number of questions asked, they were posed twice per puzzle and after the participant requested a hint.

Every time a measurement of the mental states was received through the feedback of the participant, the inputs given in Table 3 were also collected. These inputs were directly accessible in the code. Then, the estimates for the variables of the MMM were updated to incorporate the new measurements. However, the MMM, as explained in Section 2.3, may be updated at a higher frequency than the frequency of sampling the measurements. In this case study, the variables of the MMM were updated two times in between every two consecutive measurements.

3.2.2 Model identification

Preliminary HSRI were carried out with two volunteer participants, where in addition to the concepts presented in Section 3.2.1 (see Table 4), the preliminary model included two extra concepts for the belief, i.e., *Nao hinders solving the puzzle* and *participant made progress*, one for the goal, i.e., *get a reward*, and one for the emotion, i.e., *happiness*. Moreover, in addition to determining the difficulty level of the next puzzle and whether or not to offer a reward to the participant, Nao had a choice to hinder the participant by purposely giving incorrect clues or to help the participant. Hindering was hypothesized to induce frustration in the participants, generating more diverse data and improving the identification process. Note that these two volunteer participants differed from the 10 participants of the experiments, and their data has not been included in the results. Their participation aimed to assess the designed sessions by repeating the experiments multiple times, collecting feedback, and assessing the estimations. By doing so, the design of the sessions was enhanced as much as possible before involving the participants.

It was found out that, with the data collected in two sessions lasting 45 to 60 minutes per participant, it was not possible to identify all parameters unequivocally, deploying the identification and assessment methods presented in Section 2.2. Performing a third session with each participant (i.e., extending the training dataset) slightly improved the number of parameters that were unequivocally identifiable, but did

not completely address the issue. It was concluded that the control input concerning Nao yielding false hints and its corresponding belief could create confusion, especially for participants with no or limited experience with chess playing. This introduces undesirable noise into the data, hindering the identification process. By analyzing the data collected in the preliminary sessions, the belief *participant made progress*, the goal *get a reward*, and the emotion *happiness* were found to be redundant. Furthermore, the other concepts for these variables were more relevant to be tracked and optimized by Nao. Consequently, the model was simplified according to the approach that has been presented in Section 2.2.

After the preliminary phase, the model described in Section 3.2.1, which we refer to as *complete model* and is shown in Fig. 15, was obtained. This *complete model* was identified for both preliminary participants in order to assess its accuracy in tracking their mental states. Then, two extra simplifications were designed to reduce the model that was embedded in the MBC used in the last session. The first simplification was the removal of the goals *get help* and *change difficulty* (see Fig. 16 for a representation of the *simplified model*), as these goals did not directly reflect the quality of the interactions (and, consequently, did not need to be optimized), nor did they influence the other mental states. For the ten volunteer users, the choice to embed the complete or the simplified model in the MBC was done per user, by selecting the model that achieved a smaller validation error and that corresponded to a larger number of unequivocally identifiable parameters. The second simplification was the replacement of the scheduled weights in (10) by just one scalar parameter (i.e., $w_{ji}(k) = w_{ji}^{-} = w_{ji}^{+}$). This second simplification was considered only for participants whose simplified model could not be fully unequivocally identified.

Note that the weight parameters w_{ii} in (10) were set to 0.9 whenever $x_i(k)$ represented a goal or an emotion. For the participants for whom setting $w_{ii} = 0.9$ did not result in a satisfactory optimization cost, w_{ii} was gradually reduced with a step of 0.1 until the optimization cost was acceptable. Finally, to facilitate and improve the model identification, the inputs of the real-life data were normalized per participant between 0 and 1.

3.2.3 Optimizer

The integration of the MMM into an MBC is based on Fig. 6. The control input is composed of two variables, where the first one, i.e., the difficulty level of the next puzzle, is an integer variable with 6 possible realizations and the second one, i.e., *reward given?*, is a Boolean with 2 possible realizations (1 when reward is given, and 0 otherwise). Therefore, 12 possible combinations and 12 possible control inputs exist.

A cost function was formulated based on the belief, goals, and emotions that are predicted at time step k for the upcoming time step $k+1$ by the MMM, when the current state vector $\boldsymbol{x}(k) = [x_1^{\mathrm{b}}(k), x_2^{\mathrm{b}}(k), x_1^{\mathrm{g}}(k), x_2^{\mathrm{g}}(k), x_3^{\mathrm{g}}(k), x_4^{\mathrm{g}}(k), x_1^{\mathrm{e}}(k), x_2^{\mathrm{e}}(k)]^{\top}$ is measured:

$$J(\boldsymbol{x}(k)) = |x_1^{\mathrm{b}}(k+1)| + w^{\mathrm{g}} \sum_{i=1}^{2} x_i^{\mathrm{g}}(k+1) + w^{\mathrm{e}} \sum_{i=1}^{2} x_i^{\mathrm{e}}(k+1) \tag{12}$$

The estimation of the upcoming state variables based on the current measured states has been detailed in Section 3.2.1.

The formulation of (12) balances a trade-off in minimizing the cognitive variables whose rise will negatively impact the participants, including the absolute value of the first element of the belief, i.e., *the puzzle is difficult* (see Table 4), the first two elements of the goal (i.e., *quit the session* and *skip the puzzle*), and both elements of the emotion (i.e., *boredom* and *frustration*). The lower the value of the mentioned goals and emotions is, the better for the HSRI. As for the belief, since the minimum and maximum values of this variable correspond to the belief that the puzzle is too easy or too difficult, respectively, it is ideal to keep this belief as neutral as possible (i.e., as close to 0 as possible). The parameters w^{g} and w^{e} weigh these concepts relatively. In this case study, we gave an equal level of importance to all these concepts and tuned them as $w^{\mathrm{g}} = w^{\mathrm{e}} = 1$. The control input that yields in the lowest value of the cost function given by (12) is selected by the MBC for time step k.

4 Results and discussions

Next, we present the results and discussion on identifying the MMM for the participants and using the identified models in MBCs to steer behavior of Nao in the final interaction session.

4.1 Results for model identification

In order to assess the accuracy of the model in estimating the mental states, we divided the data points of the first two sessions into training data (67%) and test data (33%). Thus, the training data was used to identify the model and the test data to assess it. For this purpose, we used the complete model composed of the variables shown in Tables 3, 4, and 5, and displayed in Fig. 15. Note that in general, the preferences and personality traits, which remain constant in the short term (e.g., during the three interaction sessions), regulate the variations in the mental states (see [28] for more details). In this paper, due to the nature of the HSRIs, the overall interest of the participant in playing chess (which is a general preference) may regulate the speed for reaching

Table 6 Average MSE obtained when estimating the mental states (emotions, goals, beliefs) using the training and the test data sets in the model identification procedure. The error is normalized over all the mental states (beliefs, goals, and emotions) and time steps. Given that the mental states are bounded in $[-1, 1]$, the maximum MSE is 4. The values presented are an average over all the participants

Data set	Different frequencies[a]	Identification of perception and cognition		
		Conventional	Approach[b] A	Approach[b] B
Training	No	0.063	0.064	0.065
	Yes	0.053	0.053	0.053
Test	No	0.075	0.075	0.082
	Yes	0.067	0.067	0.067

[a] Goals and emotions are once updated with a lower frequency than beliefs and once with the same frequency, as explained in Remark 9, Section 3.2.1

[b] Approaches A and B are explained in Section 2.2

the goal "quit the game". Moreover, how focused and confident a participant is in general (as a trait), as illustrated in Fig. 15, impacts the evolution of the emotions "boredom" and "frustration", respectively. Thus, these fixed parameters have been included in MMM and are identified for the participants.

The average Mean Squared Error (MSE) in the estimation of the mental states is presented in Table 6 for when the goals and emotions are updated with a frequency lower than the frequency of updating the beliefs, and when the goals and emotions are updated with the same frequency as the beliefs (see Remark 9, Section 3.2.1). These average MSE values are presented for three cases concerning the identification procedure: (1) **Conventional**: When the coupled configuration (see Fig. 4) is used to identify the perception and cognition modules simultaneously, using a multi-start optimization but no warm-start. (2) **Approach A**: When the perception and cognition modules are identified following Algorithms 2 and 3 of Section 2.2. (3) **Approach B**: When the perception and cognition modules are identified following Algorithms 2 and 4 of Section 2.2.

The average MSE values achieved for both the training and test data sets are very satisfactory, all presenting a value under 0.1, where the maximum possible value for the MSE is 4. These values indicate the excellent capacity of MMM to estimate the invisible cognitive procedures of various participants in a personalized way for the given HSRIs. Furthermore, they demonstrate that the identification was carried out successfully. Furthermore, the hypothesis that the goals and emotions should be updated with a lower frequency than the beliefs is supported by the results, where applying this hypothesis results in a decrease in the average MSE of around 16% and 11% for the training and test data sets, respectively.

Finally, the identification of the perception and cognition following approaches A and B improves the percentage of parameters that are unequivocally identified when compared with the conventional approach, as shown in Table 7. Therefore, these two approaches contribute to making the model more interpretable, while not impacting the accuracy of the estimations made by the identified models, as shown in Table 6.

Table 7 Percentage of parameters unequivocally identified during the optimization process with the different identification approaches. The values presented are an average over all the participants

Different frequencies	Identification of perception and cognition		
	Conventional	Approach A	Approach B
No	59.44%	76.94%	80.56%
Yes	73.61%	96.39%	97.78%

Next, we present the results obtained when using the MBC for behavioral control of Nao.

4.2 Results for model-based controller

To showcase the performance of the proposed controller, which embeds MMM for one-step predictive decision-making (see Section 3.2 for details), we display in Figs. 9 and 10 the evolution of the belief, goals, and emotions of two participants over time, as well as the corresponding decisions (inputs) made by the controller. For this purpose, we used the simplified model (see Fig. 16), since this model provided better results for each participant, as described in Section 3.2.2.

To evaluate the HSRIs when our control framework is used, both objective and subjective metrics were deployed. As explained in Section 3.1, in the third session, the participants interacted with two versions of Nao — once steered by a controller based on our proposed framework and once steered by a rule-based controller that selected the inputs based on a uniform distribution. During each interaction in this session, we tracked the following metrics: the values for the self-reported goals and emotions of the participants, the results from self-reported questionnaires completed after the interaction (available in the experimental dataset [51]), and the engagement level of participants estimated from a recording of the interaction.

The answers given by the participants regarding their emotions of "boredom" and "frustration", as well as their goals to "quit the game" and "skip the puzzle", were collected with the following frequency: After the first move by the participant in the first puzzle, the questions appeared on the display and the participants were asked to provide their response. In the subsequent puzzles, after the first move of the participant in the puzzle, if more than 150s passed or if the participant played 3 puzzles since the last set of questions, then the questions were posed again. The values provided by the participants were used both to reset the values

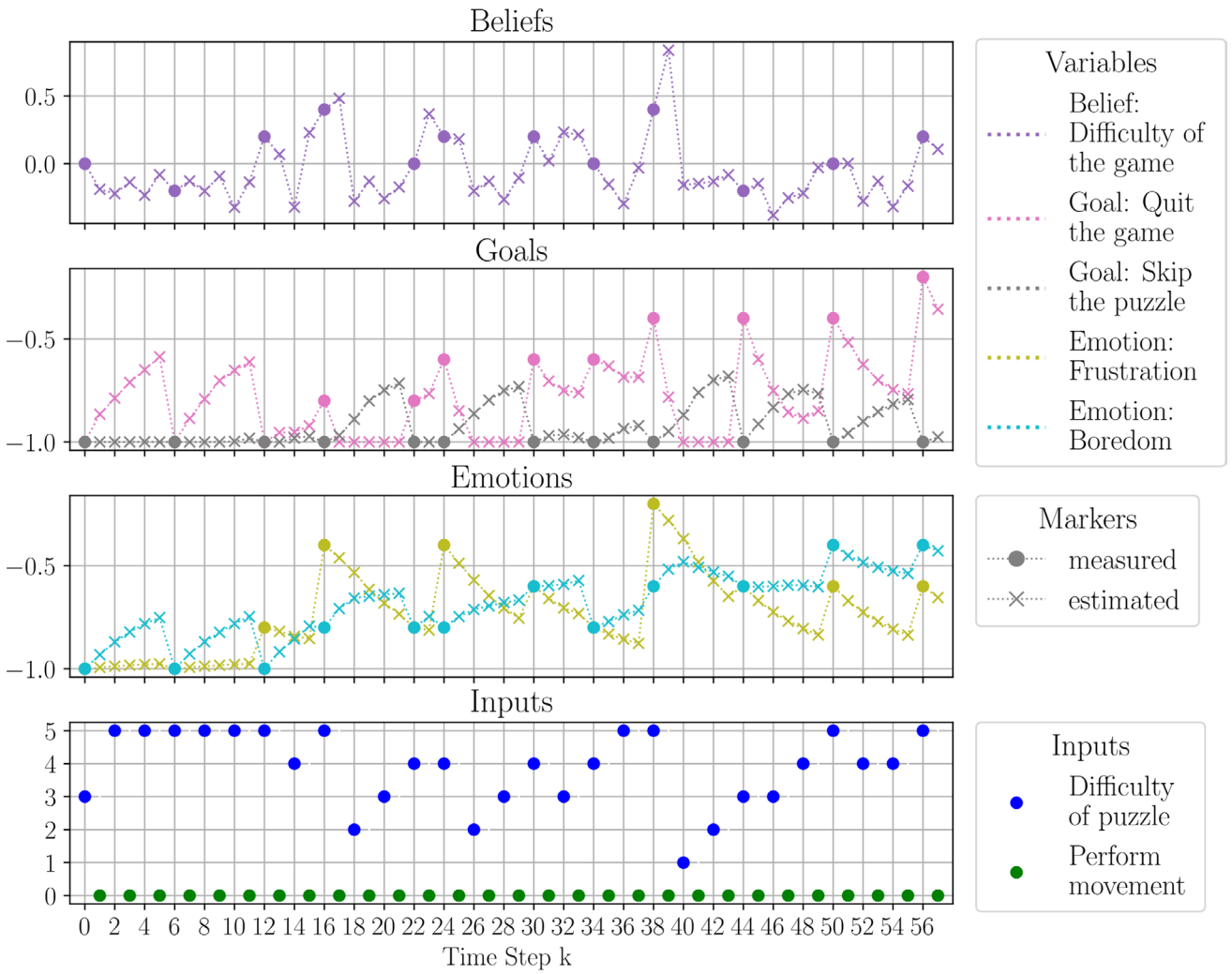


Fig. 9 Values corresponding to the mental states and inputs (i.e., decision variables of the controller) for one participant over time. The vertical lines correspond to the start of a new puzzle. After the introduction of a puzzle, two time steps were considered, such that the first time step was in the middle of playing the puzzle and the second time step coincided with the start of the next puzzle. Around time step $k = 16$, the participant suddenly became frustrated, as is deduced from the plot for emotions. Thus, in the next time step, the controller lowered the puzzle difficulty level, which, over time (i.e., by reaching time step $k = 30$), led to a decrease in the frustration. The same happens again at time step $k = 38$. Moreover, when the participant presented the same level of frustration with a higher level of boredom (see time steps $k = 44$ and $k = 40$), the difficulty level of the puzzle for the next time step was chosen to be higher by the controller. The controller effectively prevents the engagement of the participant from increasing excessively during interactions by appropriately responding to their mental states. As for the reward, the controller has inferred that the participant preferred not to receive rewarding movements from Nao

of the mental states used by MMM within the MBC and to assess the quality of the interactions. The average emotions and goals reported by each participant over the interactions with the MBC and with the rule-based controller are shown in Fig. 11.

For the majority of the participants, interacting with our MBC resulted in lower boredom (thus higher engagement) and a lower desire to quit the game or skip the puzzle. In fact, 8 out of 10 participants reported lower levels of boredom throughout the interaction with the MBC, and only 1 participant reported higher levels of boredom throughout this interaction. Regarding the frustration level, there does not seem to be a clear tendency when interacting with the MBC. Some participants showed higher frustration levels, but less boredom with the MBC approach (see participants P3 and P8 in Fig. 11). This suggests that, for some participants, a trade-off between frustration and engagement should be considered. An appropriate increase in the frustration level may be necessary to sustain engagement (e.g., for highly competitive people) and, thus, a longer-term interaction.

Additionally, Table 8 compares the average values of the goals and emotions obtained with each controller. On average, when interacting with the Nao controlled

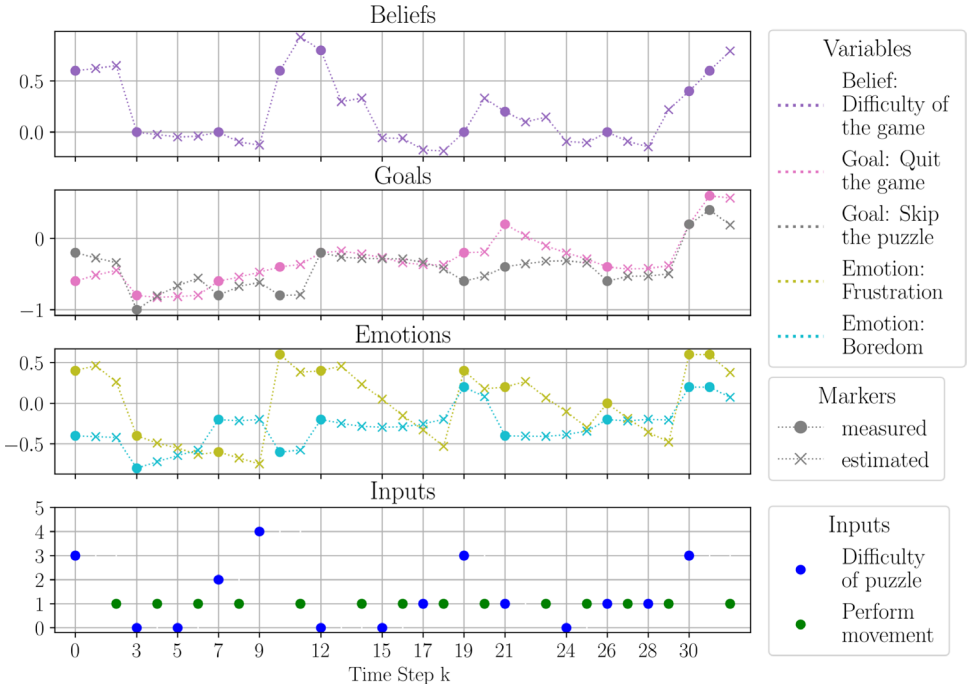


Fig. 10 Values corresponding to the mental states and inputs (i.e., decision variables of the controller) for one participant over time. The vertical lines correspond to the start of a new puzzle. Right after the start of the interaction, the participant felt frustrated, as illustrated in the plot for emotions. Hence, the controller immediately decreased the difficulty level of the next puzzle, which resulted in a decrease in the frustration of the participant. At time step $k = 7$, the participant started becoming bored. Thus, for the next time step $k = 9$, the controller has increased the level of difficulty of the puzzle. Again, the participant became frustrated, and the controller decreased the difficulty level of the puzzle accordingly. The controller shows to be able to sustain the engagement of the participant stably throughout the interactions by properly reacting to their mental states. As for the reward, the controller has inferred that the participant prefers to always receive rewarding movements from Nao

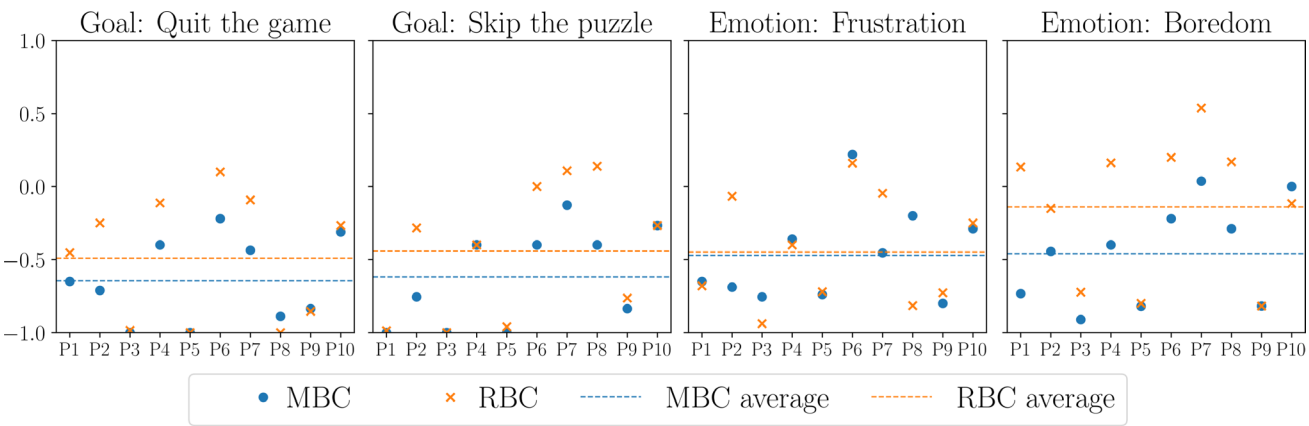


Fig. 11 Average self-reported values of the emotions and goals of each participant over the interactions with the MBC and the rule-based controller. Each graph corresponds to one variable. For a participant, the average value of that variable over the interaction with each controller is represented by a marker, whereas the dotted line represents the average value over all participants for that variable and controller (RBC stands for rule-based controller)

by the MBC approach, the desire of the participants to quit the game and to skip the puzzles were, respectively, 7.70% ($p = 0.032$) and 8.86% ($p = 0.029$) less than when the rule-based controller was deployed. Furthermore, participants felt 15.98% ($p = 0.009$) more engaged when interacting with Nao steered by the MBC. There were in general no indications that using the MBC approach was able to reduce the frustration of the participants ($p = 0.827$).

After interacting with Nao using each controller, the participant was given a questionnaire to assess the HSRI (see Fig. 8b for the structure of the session). The questionnaire, which is included in [51], consisted of five questions

Table 8 Comparison between the average values of the goals and emotions for all participants when interacting with our MBC and with the rule-based controller. The improvement of the average values per variable when the participants interacted with the MBC, as well as the p-values, are also specified

Variable	Average		Diff.	
	MBC	RBC	(%)	p-value
$x_1^{\mathrm{g}}(k)$: Quit the game	-0.645	-0.491	7.70	0.032
$x_2^{\mathrm{g}}(k)$: Skip the puzzle	-0.619	-0.441	8.86	0.029
$x_2^{\mathrm{e}}(k)$: Frustration	-0.472	-0.448	1.17	0.827
$x_1^{\mathrm{e}}(k)$: Boredom	-0.460	-0.140	15.98	0.009

regarding the engagement, adaptability, personalization, and awareness of the robot regarding the mental states of the participant. Two questions concerned the engagement, asking participants how bored and how engaged they felt in the course of the interactions. The participants answered these questions using a 7-point Likert scale. During the debrief, preliminary participants (2 participants whose input was used to improve the setup of the experiments, but whose data was not included in the results) reported perceiving a (significant) difference in some of these categories between the two interactions deploying the MBC and the rule-based controller. However, their absolute answers to the questionnaires did not reflect this difference. When questioned about this, the preliminary participants reported that they were unsure whether they considered the same scale to answer both questionnaires due to the time gap between the sessions. They reported that, when answering the second questionnaire, they did not remember in detail the answers given in the first interaction, and that, during the second interaction, they experienced a change of perception of the assessed aspects (i.e., engagement, adaptability, etc.), which could have led them to use a different scale. To prevent this undesirable effect, after answering the questionnaire per session (once after each interaction), the participants were given an additional questionnaire where they could provide their responses in a relative sense, comparing their experiences about the two sessions. This comparative questionnaire is available in the experimental dataset [51]. Table 9 shows the results from the questionnaires.

Although the average answers regarding the mental states favored the proposed MBC, the answers given by the participants in the absolute questionnaires regarding the engagement, awareness shown by the robot about their mental states, personalization, and boredom are not conclusive ($p > 0.05$). Only the answers regarding the personalization of the robot demonstrate that participants experienced our proposed approach as tailored to themselves ($p = 0.04$). Regarding the other four aspects, the comparative answers must be analyzed: The results indicate that 7 participants reported feeling more engaged and less bored when interacting with the MBC, while 3 participants reported the opposite. Furthermore, 8 participants perceived that the robot controlled by the MBC approach was more aware of their mental states, 1 participant had no opinion, and the other participant perceived the rule-based controller to display more awareness. Finally, 6 participants perceived our MBC approach to be more adapted to them, whereas 3 participants stated having no opinion.

Finally, a video stream was used to capture the engagement of the users, using the tool presented in [52]. This tool quantified the level of engagement between 0 and 1, where a value close to 1 indicates that the participant is engaged. Although this tool correctly measured the engagement from the perspective of the robot in [52], the dataset where it was trained included only robot-centric interactions, i.e., the robot was the focus point. In our case, the main objective was for the robot to keep the participants engaged in solving the puzzles. Thus, we placed the camera right above the screen where the puzzles were displayed to record the participants, rather than on the robot. Hence, the tool indicated a high level of engagement when the participant gazed at the screen. Nevertheless, it indicated a low score regardless of whether the participant looked away from the interaction

Table 9 Results obtained from the two questionnaires given to participants in the third session. The absolute answers refer to the two questionnaires posed at the end of each interaction of the third session, while the relative answers correspond to the final questionnaire given at the end of the third session (see Fig. 8b). Note that the absolute answers were given in the range [1, 7]. Thus, the closer the value of the positive indicators (i.e., engagement, awareness, adaptability, and personalization) are to 7, the better. The value of the boredom should be as low as possible. In the comparative questionnaire, participants could choose between one of the two controllers or neither for each question. For each criteria, the number of participants who have voted for each of controller (or neither) is counted. Therefore, the numbers per row add up to the total number of participants, i.e., 10

Question	Absolute answers			Comparative answers		
	MBC	RBC	p-value	MBC	RBC	No opinion
Q1: Engagement	4.7	4.4	0.496	7	3	0
Q2: Awareness	4.0	2.9	0.057	8	1	1
Q3: Adaptability	3.6	2.5	0.040	5	1	4
Q4: Personalization	3.4	3.0	0.479	6	1	3
Q5: Boredom	3.5	4.9	0.158	2	7	1

or at the robot. The average engagement of each participant, according to this tool, is presented in Table 10. The results show non-significant differences between the two interactions of each participant ($p = 0.106$) and are not coherent with the self-reported answers to the questions asked during the interactions or the questionnaire. This discrepancy is likely due to the engagement measurement tool not being suitable for our application since it did not account for a second focus point. As a result, this metric was not representative of the real engagement of the user in the current HSRI.

Overall, these results demonstrate the effectiveness of the proposed model-based control framework in adapting robot behavior to the mental states of users. However, the number of participants in this case study was limited, which constrains the generalizability of the findings. As explained in Section 3, this is a common characteristic of early-stage HSRI experiments that involve multi-session protocols and personalized modeling. In such cases, the primary goal is to establish feasibility and to validate the underlying approach rather than to achieve broad statistical generalization. Broader validation of the proposed framework requires future studies with larger participant pools and more diverse scenarios, as discussed in Section 5.

4.3 Computational efficiency

To assess the computational feasibility of the proposed framework, we analyzed the run times and memory requirements associated with the parameter identification and the real-time deployment of the MBC.

4.3.1 Computational efficiency of model identification

The run time of model identification is determined mainly by three factors: the number of gradient descent runs, the number of iterations per run, and the dimensionality of the identification problem (i.e., the number of model parameters and data points used for their identification).

Each gradient descent run corresponds to an independent process with different initialization values. Consequently, the runs can be executed in parallel across available processor cores. Thus, the total identification time increases linearly with the number of runs and decreases inversely with the number of parallel workers.

The number of iterations per run also contributes linearly to the total run time, as each iteration involves recomputing the gradients and updating the model parameters. Empirical measurements were consistent with this linear relationship across multiple runs, participants, and model configurations. The number of iterations per run, however, varies across runs because a pruning strategy has been implemented to terminate runs with poor or stagnated performance after a predefined number of iterations, improving overall efficiency.

Finally, the identification time also scales approximately linearly with the problem complexity, which is determined by the number of model parameters and the number of data points, as both factors proportionally affect the duration of each iteration. Empirical measurements across participants and model configurations confirmed this relationship, with a normalized run time per iteration of 0.204 ms on average and 0.250 ms at maximum, using an Intel Xeon Gold 6448Y (2.1 GHz) CPU.

As a reference, the maximum observed identification time corresponded to the participant with the largest training dataset (including 87 data points) and the most complex model (including 36 parameters). On a machine equipped with two Intel Xeon Gold 6448Y (2.1 GHz) processors, using 48 parallel workers, this configuration required 964s for 250 independent runs with up to 120 iterations per run. The multi-stage identification procedures described in Section 2.2 showed comparable run times: 983 s for Approach A (i.e., pre-identification of the perception module, followed by a joint optimization), and 921 s for Approach B (i.e., pre-identification of the perception and cognition modules, followed by a joint optimization).

Since the model identification is intended for long-term HSRIs and can be performed offline between sessions, these run times remain well within practical limits.

In terms of memory requirements, the baseline memory consumption per parallel process, after environment initialization, was at most 65.10 MB. This was measured with the most complex model (including 36 parameters) and the largest dataset (including 87 data points). Although memory usage increases slightly with the number of model parameters and the size of the training dataset, the variation is insignificant in practice. For example, reducing the number of parameters from 36 to 24 or the number of data points from 87 to 35 yielded marginal decreases to 65.06 MB and 64.87 MB, respectively. During active identification, each gradient-descent process required an additional 1.04 MB at peak usage, for the largest model and training dataset. Consequently, when multiple workers are executed in parallel, the total memory overhead

Table 10 Results obtained for the participants using the engagement measurement tool provided in [52]

Controller	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Average
MBC	0.948	0.962	0.903	0.859	0.945	0.841	0.968	0.884	0.931	0.923	0.916
RBC	0.948	0.942	0.930	0.880	0.951	0.864	0.970	0.919	0.924	0.932	0.926

scales approximately linearly with the number of active processors.

Overall, these requirements are negligible relative to modern computational capacities, confirming that memory is not a limiting factor for the proposed identification procedure.

4.3.2 Computational efficiency of real-time deployment

For real-time deployment, we evaluated the run time and memory requirements of the MBC for different model configurations, corresponding to varying numbers of parameters. Across all tested configurations, the controller computed a new action within 5.4 ms on a laptop equipped with an Intel i7-1185G7 (3.0 GHz) CPU. Additionally, the controller required less than 8.14 MB of RAM across all configurations, with only negligible variations due to model complexity. The runtime environment, including the Python 3 interpreter and supporting libraries, occupied 66.53 MB.

Because the control loop operates in discrete time steps of a few seconds, consistent with the natural timescale of HSRIs, these run times are negligible in practice. Hence, the controller runs reliably in real time, without introducing perceptible delays. Combined with the low memory footprint, these results confirm the computational efficiency and suitability of the proposed framework for real-time interactive social robotic applications.

4.4 Limitations and mitigation strategies

While MMM and its integration within an MBC offer a mathematically grounded, transparent, and adaptive framework for controlling SRs based on the mental states of their users, certain limitations must be discussed, along with corresponding mitigation strategies. This section clarifies the main factors that may affect the effectiveness of the model and highlights the mechanisms through which their impact can be mitigated within the proposed design.

A primary source of potential underperformance of the presented framework arises from model mismatch between MMM and the actual cognitive dynamics of the user. Such a mismatch may occur when the data available for parameter identification is scarce or lacks sufficient diversity, limiting the ability to achieve accurate model identification and personalization. It may also occur when users display highly unpredictable or atypical behaviors, for which MMM may fail to capture the corresponding cognitive dynamics accurately. Additionally, if users fail to provide accurate and reliable self-reports during the initial sessions when data is collected for identifying MMM, the resulting parameter estimates may be imprecise, thereby reducing the accuracy of the identified model. These conditions reduce the accuracy of the estimations made by the model and may consequently lead to sub-optimal behavioral adaptation by the robot during deployment. Nevertheless, model mismatch is a challenge inherent to all model-based approaches and is not unique to MMM.

In addition to model mismatch, underperformance may also occur due to state estimation drift. Since mental states are not directly visible or measurable, the controller relies on MMM to propagate state estimates between intermittent self-reports. Such self-reports should not be requested too frequently to avoid disruption in natural HSRIs. Therefore, the estimates may gradually drift from the actual mental states of the user over time, and potentially degrade the performance and effectiveness of the control system, which relies on estimates provided by MMM.

While the challenges discussed in this section are inherent to any model-based approach, they can be effectively mitigated through several complementary strategies. First, regular re-identification of model parameters over time helps maintain alignment between MMM and the cognitive dynamics of the user. Second, if necessary, the structure of the model may be extended to capture user-specific or previously unmodeled behavioral patterns, which enhances its representational accuracy. Third, integrating a state estimator within the control loop is expected to improve the accuracy of mental state estimates between self-report updates.

These mitigation strategies are particularly feasible in long-term HSRIs, where extended interaction times provide sufficient data for iterative refinement and progressive improvement of the model.

5 Conclusions and topics for future research

This paper introduces a novel systems-and-control theoretic framework for designing and sustaining Human-Social-Robot Interactions (HSRIs), ensuring desired subjective outcomes (e.g., long-term human engagement) and enriched human experiences (e.g., feeling understood by the robot), while enabling systematic adaptability driven by the robot in the interactions.

Leveraging the recently introduced mathematical model of human perception, cognition, and decision-making, Mathematical Model of Mind (MMM) [28], grounded in the principles of Theory of Mind (ToM), we have adapted MMM for Social Robots (SRs) to exhibit ToM-like behavior. This model has been embedded into a controller, enabling SRs to track and adapt their interactive behavior

based on the evolving mental states of their human counterpart. This critical capability addresses a key limitation in current SRs, empowering them to sustain meaningful social interactions with humans [3, 19] and to engage them for extended periods. This is expected to result in significant societal impacts, particularly for vulnerable populations [4, 13, 14, 16, 17, 22].

We carried out a case study with 10 volunteer participants who solved chess puzzles on a screen while interacting and being guided by a Nao robot. The case-study consisted of 3 sessions lasting 45 to 90 minutes. During these sessions, the robot dynamically adjusted the difficulty level of the puzzles and rewarded participants with entertaining movements. In the first two sessions, data on mental states of the participants was collected while they played the chess puzzles and interacted with Nao. This data was then used to personalize the parameters of MMM for each participant. In the final session, participants interacted with Nao under two conditions: once, when the robot was steered by a model-based controller embedding their personalized MMM, and once when Nao was guided by a conventional rule-based controller that did not systematically consider their mental states.

MMM achieved an average mean squared error of 0.067 in tracking the beliefs, goals, and emotions of the participants within a normalized range of $[-1, 1]$. Compared to the rule-based controller, the MMM-based controller increased participant engagement by 16% ($p = 0.009$) and decreased the goal of quitting the game by 8% ($p = 0.032$), as measured objectively in the final sessions. Responses to a post-interaction questionnaire further confirmed that most participants perceived Nao, when controlled by the MMM-based controller, to be more engaging, more aware of and adaptive to their mental states, and more personalized to their needs.

Although the MMM-based controller successfully adjusted the difficulty level of the puzzles based on the mental states of the participants in the final session, it did not do the same for rewarding them. One possible reason is that during the first two sessions, while the novelty effect (i.e., a temporary increase in the engagement and interest of individuals when exposed to new experiences, e.g., interacting with a Nao robot) was still present, participants consistently reacted positively or negatively to the entertaining movements of Nao. This may have led MMM to infer that these movements always had either a positive or a negative influence on the participants, regardless of their actual mental states. Future work should explore the impact of the novelty effect on collected data, quantifying its duration and magnitude to determine when user responses stabilize. This will enable the development of identification strategies less affected by novelty effects in the training data, such as re-identifying the model once the novelty effect has subsided.

Moreover, the present case study involved a relatively small and homogeneous group of participants, as it was designed mainly to demonstrate the feasibility and effectiveness of the proposed framework under controlled conditions. Future studies should expand the participant pool through larger-scale, multi-phase, and longer-duration experiments to strengthen the statistical validity and generalizability of the findings. Additionally, the current design of MMM and the experiments focused on neurotypical users. Future work should include neurodivergent or cognitively impaired participants to examine whether the formulation of MMM generalizes across different cognitive profiles, and, if needed, to refine the model for those user groups.

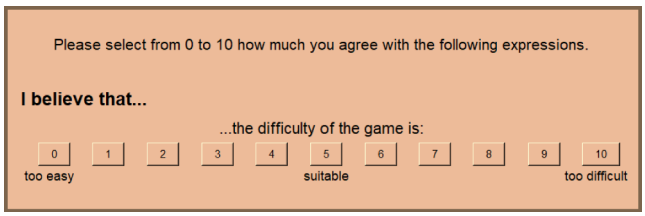
Given the central role of engagement in HSRIs, future developments should also improve the objective engagement measurement method used in this study. In the present case study, this measurement method relied on a single camera view and thus lacked precision in multi-focus settings. Future implementations should include one camera per focus point and merge the results from all cameras to gather an accurate estimate of the engagement of the user. Additionally, a personality test could be administered to the participants prior to the interaction to establish which mental states are relevant to optimize (e.g., whether the frustration might boost or hinder the engagement of that participant).

Finally, while MMM demonstrated excellent performance in short-term predictions of mental states, its long-term estimations declined for some participants. Future work will focus on integrating dedicated state estimators into the closed-loop control system to mitigate this estimation drift. This will reduce reliance on direct self-reports from users and enable greater robustness and improved real-time adaptation during extended HSRIs.

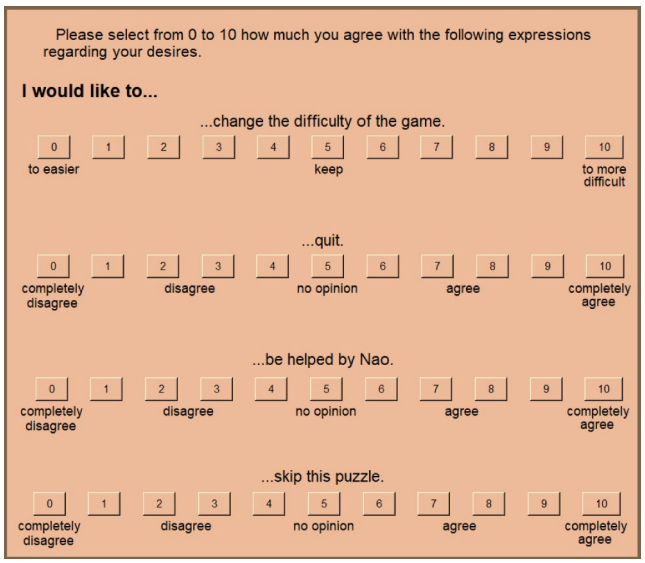
Overall, this paper is the first to explore systems-and-control-theoretic methods for HSRIs, providing guarantees on performance and safety. Our findings demonstrate how cognitively grounded, model-based control and control-theoretic principles can enable SRs to exhibit adaptive, cognition-aware, and personalized behavior, leading to more meaningful HSRIs and sustaining long-term engagement. In doing so, our work charts a new trajectory toward fundamentally new forms of SRs control that could enable sustained and personalized long-term support across fields such as healthcare, education, and assisted living, all of which face pressing challenges related to increasing demand for long-term, individualized support.

Appendix A: Questions posed to the participants during the interactions with Nao

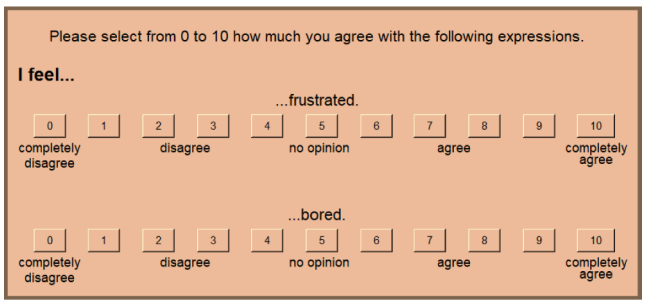
This appendix includes the questions asked to the participants during their interactions with the Nao robot. The questions were posed to the participants to gather information about their beliefs (Fig. 12a), goals (Fig. 12b), and emotions (Fig. 12c). Furthermore, at the end of each puzzle, participants were presented with the questions shown in Fig. 12d, allowing them to request an easier or more difficult puzzle. However, while participants could make this request, Nao did not fulfill it. Instead, the request was used to gather information about the participant's actions.



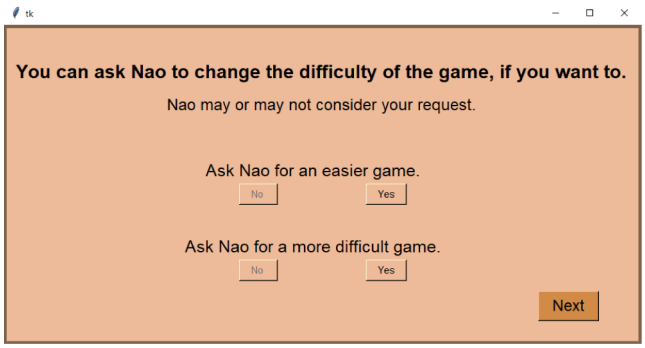
(a) Beliefs



(b) Goals



(c) Emotions

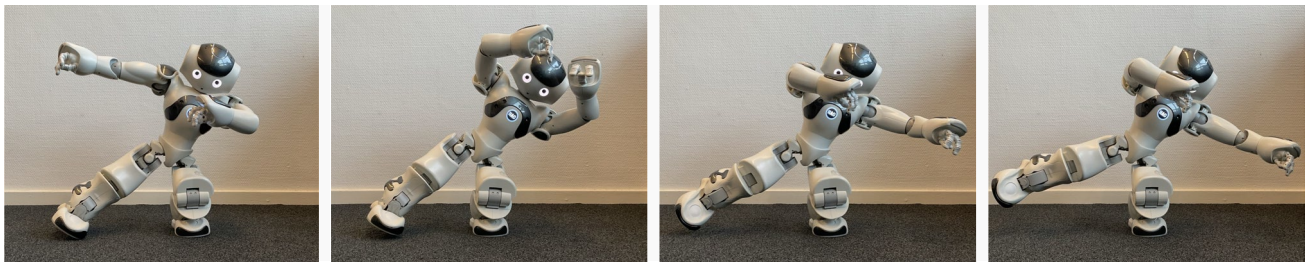


(d) Actions

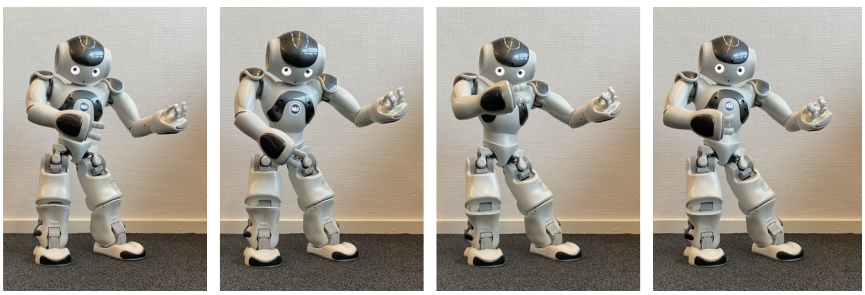
Fig. 12 Periodic questions asked to the participants during the interaction with the Nao robot. In the first and second sessions, they enabled the collection of the data used to identify the MMM per participant. In the last session, they were used to assess the interaction and to update the state variables of the MMM that was integrated in the controller of Nao

Appendix B: Reward movements performed by Nao in the case study

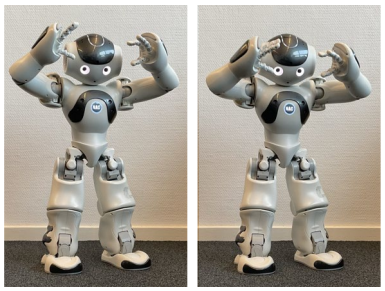
This appendix contains a visual representation of the five entertaining movements performed by the Nao during experimental sessions of the case study. These movements were performed by the Nao robot to reward the participants and keep them engagement in the interaction. These five movements included performing tai chi (Fig. 13a), playing the guitar (Fig. 13b), pretending to take a photo (Fig. 13c), dancing (Fig. 13d), and pretending to be an elephant (Fig. 13e).



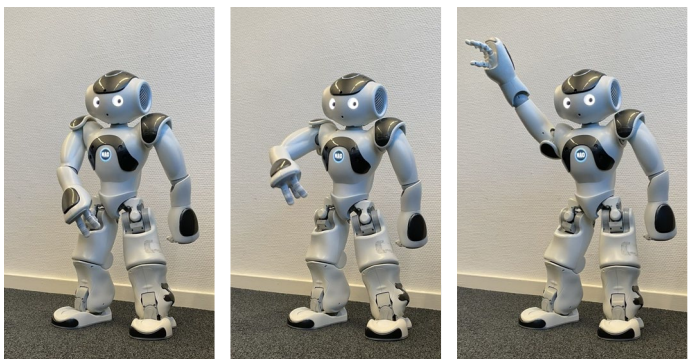
(a) Performing Tai Chi



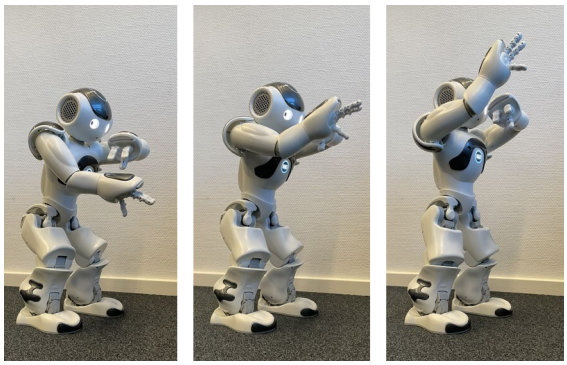
(b) Playing the guitar



(c) Taking a photo



(d) Dancing



(e) Mimicking an elephant

Fig. 13 The five reward movements performed by the Nao robot in the experimental sessions with the participants

Appendix C: Structure of the modules of the MMM used in the case study

This appendix includes graphical representations of part of the perception module (Fig. 14), the cognition module (Figs. 15 and 16), and part of the decision-making module (see Fig. 16). These figures showcase the variables that are part of each module, as well as which variables influence each other (these influences are represented by arrows).

Furthermore, two structures are displayed for the cognitive module, Figs. 15 and 16. While Fig. 15 represents the model used in the first two sessions with the participants, Fig. 16 shows the model used in the last session for the integration within the model-based controller. For clarity, the influencers of the variables in the cognition module are also given in Table 11.

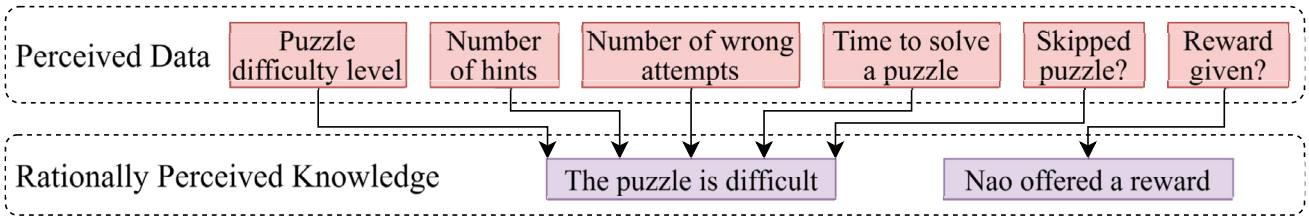


Fig. 14 Rational reasoning sub-process of the perception module of the ToM model (MMM) used in the case study

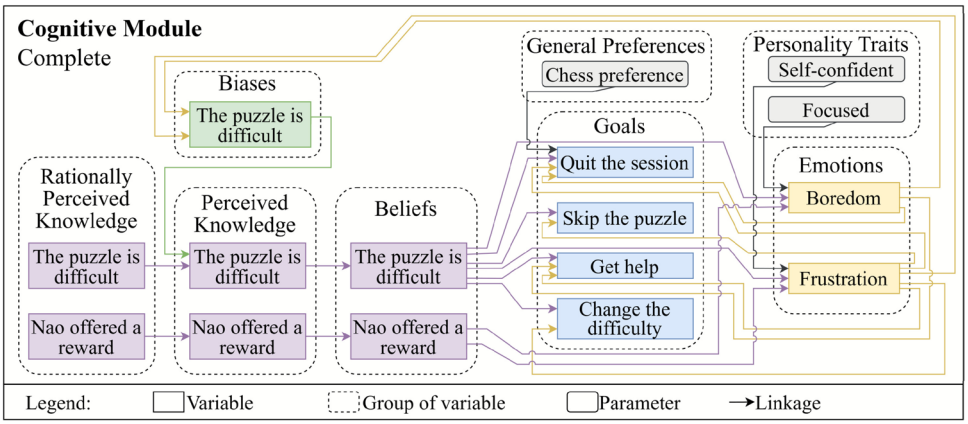


Fig. 15 Variables and linkages of the cognitive module of the ToM model (MMM) used in the case study

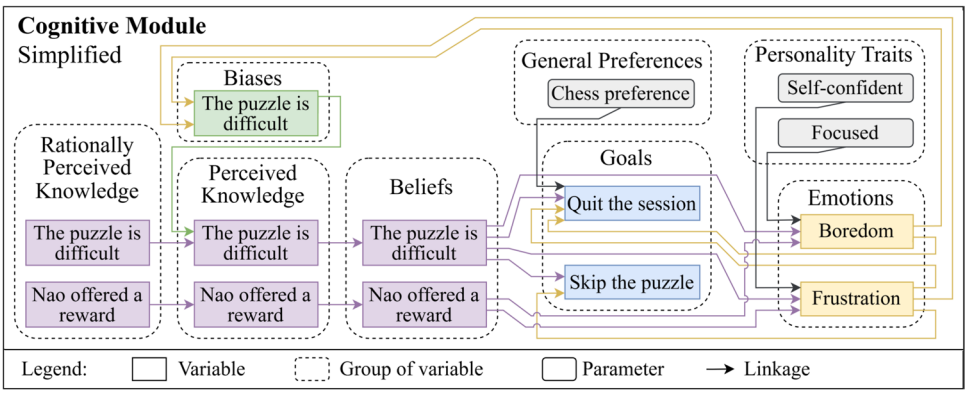


Fig. 16 Variables and linkages of the simplified cognitive module of the ToM model (MMM) used in the case study

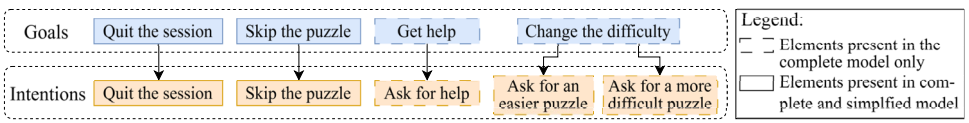


Fig. 17 Variables and linkages of the rational intention selection of the decision-making module of the ToM model (MMM) used in the case study

Table 11 The influencer variables of each one of the variables of the cognition module

Variable and concept	Influencers (and concepts)
$x_1^{\mathrm{b}}(k)$ (The puzzle is difficult)	$x_1^{\mathrm{PK}}(k)$ (The puzzle is difficult)
$x_2^{\mathrm{b}}(k)$ (Nao offered a reward)	$x_2^{\mathrm{PK}}(k)$ (Nao offered a reward)
$x_1^{\mathrm{g}}(k)$ (Quit the session)	$x_1^{\mathrm{b}}(k)$ (The puzzle is difficult), $x_1^{\mathrm{e}}(k)$ (Boredom), and $x_2^{\mathrm{e}}(k)$ (Frustration)
$x_2^{\mathrm{g}}(k)$ (Skip the puzzle)	$x_1^{\mathrm{b}}(k)$ (The puzzle is difficult) and $x_2^{\mathrm{e}}(k)$ (Frustration)
$x_3^{\mathrm{g}}(k)$ (Get help)	$x_1^{\mathrm{b}}(k)$ (The puzzle is difficult), $x_1^{\mathrm{e}}(k)$ (Boredom), and $x_2^{\mathrm{e}}(k)$ (Frustration)
$x_4^{\mathrm{g}}(k)$ (Change difficulty)	$x_1^{\mathrm{b}}(k)$ (The puzzle is difficult) and $x_2^{\mathrm{e}}(k)$ (Frustration)
$x_1^{\mathrm{e}}(k)$ (Boredom)	$x_1^{\mathrm{b}}(k)$ (The puzzle is difficult) and $x_2^{\mathrm{b}}(k)$ (Nao offered a reward)
$x_2^{\mathrm{e}}(k)$ (Frustration)	$x_1^{\mathrm{b}}(k)$ (The puzzle is difficult) and $x_2^{\mathrm{b}}(k)$ (Nao offered a reward)
$x^{\mathrm{bias}}(k)$ (The puzzle is difficult)	$x_1^{\mathrm{e}}(k)$ (Boredom) and $x_2^{\mathrm{e}}(k)$ (Frustration)
$x_1^{\mathrm{PK}}(k)$ (The puzzle is difficult)	$y_1^{\mathrm{RR}}(k)$ (The puzzle is difficult) and $x^{\mathrm{bias}}(k)$ (The puzzle is difficult)
$x_2^{\mathrm{PK}}(k)$ (Nao offered a reward)	$y_2^{\mathrm{RR}}(k)$ (Nao offered a reward)

Acknowledgements This research has been supported by the TU Delft AI Labs & Talent programme.

Data Availability The data generated and analyzed in this study is available in an anonymized format in https://doi.org/10.4121/ccadc914-9502-46d6-9ba5-fef581f2933f. This includes the anonymized raw data collected during experiments and the processed data derived from it. However, video recordings of participants are not publicly available due to restrictions outlined in the informed consent forms.

Declarations

Ethical Approval All experiments involving human participants described in this article were approved by the Human Research Ethics Committee of TU Delft under *Approval No. 3780*. The procedures of data collection were GDPR compliant and an informed consent was obtained from all participants.

Conflicts of Interest The authors declare that they have no competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Bradwell HL, Noury GEA, Edwards KJ, Winnington R, Thill S, Jones RB (2021) Design recommendations for socially assistive robots for health and social care based on a large scale analysis of stakeholder positions: Social robot design recommendations. Health Policy Technol 10:1–7. https://doi.org/10.1016/J.HLPT.2021.100544
2. Macalupu V, Miller E, Martin L, Caldwell G (2025) Human-robot interactions and experiences of staff and service robots in aged care. Scientif Reports 15:1–15. https://doi.org/10.1038/S41598-025-86255-w
3. Matarić MJ, Scassellati B (2016) Socially Assistive Robotics. In: Springer Handbook of Robotics, 1st edn. Springer, Heidelberg, Germany, pp 1973–1993. https://doi.org/10.1007/978-3-319-32552-1_73
4. Lee MH, Siewiorek DP, Smailagic A, Bernardino A, Badia SB (2023) Design, development, and evaluation of an interactive personalized social robot to monitor and coach post-stroke rehabilitation exercises. User Model User-Adapted Interact 33:545–569. https://doi.org/10.1007/S11257-022-09348-5/TABLES/5
5. Bettosi C, Baillie L, Ross MK, Broz F (2024) A systematic approach to modeling structured behavior in social robots. In: Proceedings of the 2024 International Symposium on Technological Advances in Human-Robot Interaction, New York, NY, USA, pp 29–37. https://doi.org/10.1145/3648536.3648540
6. Céspedes N, Raigoso D, Múnera M, Cifuentes CA (2021) Long-term social human-robot interaction for neurorehabilitation: Robots as a tool to support gait therapy in the pandemic. Front Neurorobot 15:1–12. https://doi.org/10.3389/FNBOT.2021.612034
7. Kubota A, Peterson EIC, Rajendren V, Kress-Gazit H, Riek LD (2020) Jessie: Synthesizing social robot behaviors for personalized neurorehabilitation and beyond. In: Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction, New York, NY, USA, pp 121–130. https://doi.org/10.1145/3319502.3374836
8. Benedictis RD, Umbrico A, Fracasso F, Cortellessa G, Orlandini A, Cesta A (2023) A dichotomic approach to adaptive interaction for socially assistive robots. User Model User-Adapted Interact 33:293–331. https://doi.org/10.1007/s11257-022-09347-6
9. Sequeira P, Alves-Oliveira P, Ribeiro T, Di Tullio E, Petisca S, Melo FS, Castellano G, Paiva A (2016) Discovering social interaction strategies for robots from restricted-perception wizard-of-oz studies. In: 2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Christchurch, New Zealand, pp 197–204. https://doi.org/10.1109/HRI.2016.7451752
10. Schicchi D, Pilato G (2018) A social humanoid robot as a playfellow for vocabulary enhancement. In: 2018 Second IEEE International Conference on Robotic Computing (IRC), Laguna Hills, CA, USA, pp 205–208. https://doi.org/10.1109/IRC.2018.00044
11. Páez J, González E, Impedovo M (2020) Hrs-edu: Architecture to control social robots in education. In: Methodologies and Intelligent Systems for Technology Enhanced Learning, 10th International Conference, L'Aquila, Italy, pp 117–126. https://doi.org/10.1007/978-3-030-52538-5_13
12. Cao HL, Beir AD, Esteban PG, Simut R, Perre GVD, Lefeber D, Vanderborght B (2017) An end-user interface to generate homeostatic behavior for nao robot in robot-assisted social therapies. Adv Computat Intell 10306:609–619. https://doi.org/10.1007/978-3-319-59147-6_52
13. Scassellati B, Boccanfuso L, Huang CM, Mademtzi M, Qin M, Salomons N, Ventola P, Shic F (2018) Improving social skills in children with ASD using a long-term, in-home social robot. Sci Robot 3:1–9. https://doi.org/10.1126/scirobotics.aat7544

14. Clabaugh C, Mahajan K, Jain S, Pakkar R, Becerra D, Shi Z, Deng E, Lee R, Ragusa G, Matarić M (2019) Long-term personalization of an in-home socially assistive robot for children with autism spectrum disorders. Front Robot AI 6:1–18. https://doi.org/10.3389/frobt.2019.00110
15. Ascensão T, Jamshidnejad A (2022) Autonomous socially assistive drones performing personalized dance movement therapy: An adaptive fuzzy-logic-based control approach for interaction with humans. IEEE Access 10:15746–15770. https://doi.org/10.1109/ACCESS.2022.3143992
16. Luperto M, Romeo M, Lunardini F, Monroy J, Hernandez Garcia D, Abbate C, Cangelosi A, Ferrante S, González-Jiménez J, Basilico N, Borghese A (2024) Exploring the viability of socially assistive robots for at-home cognitive monitoring: Potential and limitations. Intern J Soc Robot 17:823–841. https://doi.org/10.1007/s12369-024-01158-6
17. Khosla R, Chu M-T, Khaksar SMS, Nguyen K, Nishida T (2019) Engagement and experience of older people with socially assistive robots in home care. Assist Technol 33:1–15. https://doi.org/10.1080/10400435.2019.1588805
18. Cooper S, Di Fava A, Vivas C, Marchionni L, Ferro F (2020) Ari: the social assistive robot and companion. In: 2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), Naples, Italy, pp 745–751. https://doi.org/10.1109/RO-MAN47096.2020.9223470
19. Leite I, Martinho C, Paiva A (2013) Social Robots for Long-Term Interaction: A Survey. Intern J Soc Robot 5:291–308. https://doi.org/10.1007/s12369-013-0178-y
20. Matheus K, Ramnauth R, Scassellati B, Salomons N (2025) Long-term interactions with social robots: Trends, insights, and recommendations. ACM Trans Human-Robot Interact 14:1–42. https://doi.org/10.1145/3729539
21. Maroto-Gómez M, Alonso-Martín F, Malfaz M, Castro-González, Castillo JC, Salichs M (2023) A systematic literature review of decision-making and control systems for autonomous and social robots. Intern J Soc Robot 15:745–789. https://doi.org/10.1007/S12369-023-00977-3/FIGURES/20
22. Tapus A, Mataric MJ (2008) Socially assistive robots: The link between personality, empathy, physiological signals, and task performance. In: AAAI Spring Symposium: Emotion, Personality, and Social Behavior, Stanford, CA, USA, pp 133–140
23. Dell'Anna D, Jamshidnejad A (2022) Evolving fuzzy logic systems for creative personalized socially assistive robots. Eng Appl Artif Intell 114:105064. https://doi.org/10.1016/j.engappai.2022.105064
24. Bagheri E, Roesler O, Cao HL, Vanderborght B (2021) A reinforcement learning based cognitive empathy framework for social robots. Intern J Soc Robot 13:1079–1093. https://doi.org/10.1007/S12369-020-00683-4/TABLES/7
25. Tapus A, Mataric M, Scassellati B (2007) The grand challenges in socially assistive robotics. IEEE Robot Autom Mag 14:35–42. https://doi.org/10.1109/MRA.2007.339605
26. Premack D, Woodruff G (1978) Does the chimpanzee have a theory of mind? Behav Brain Sci 1:515–526. https://doi.org/10.1017/S0140525X00076512
27. Patacchiola M, Cangelosi A (2020) A developmental cognitive architecture for trust and theory of mind in humanoid robots. IEEE Trans Cybern 52:1–13. https://doi.org/10.1109/TCYB.2020.3002892
28. Patrício MLM, Jamshidnejad A (2023) Dynamic mathematical models of theory of mind for socially assistive robots. IEEE Access. 11:103956–103975. https://doi.org/10.1109/ACCESS.2023.3316603
29. Gao Y, Barendregt W, Obaid M, Castellano G (2018) When robot personalisation does not help: Insights from a robot-supported learning study. In: RO-MAN 2018 - 27th IEEE International Symposium on Robot Anata Human Interactive Communication, Nanjing, China, pp 705–712. https://doi.org/10.1109/ROMAN.2018.8525832
30. Hussain N, Erzin E, Sezgin TM, Yemez Y (2022) Training socially engaging robots: Modeling backchannel behaviors with batch reinforcement learning. IEEE Trans Affect Comput 13:1840–1853. https://doi.org/10.1109/TAFFC.2022.3190233
31. Filippini C, Spadolini E, Cardone D, Bianchi D, Preziuso M, Sciarretta C, Cimmuto V, Lisciani D, Merla A (2021) Facilitating the child-robot interaction by endowing the robot with the capability of understanding the child engagement: The case of mio amico robot. Intern J Soc Robot 13:677–689. https://doi.org/10.1007/S12369-020-00661-W/FIGURES/8
32. Yu C, Pei H (2021) Face recognition framework based on effective computing and adversarial neural network and its implementation in machine vision for social robots. Comput Electric Eng 92:107128. https://doi.org/10.1016/j.compeleceng.2021.107128
33. Merino-Fidalgo S, Sánchez-Girón C, Zalama E, Gómez-García-Bermejo J, Duque-Domingo J (2025) Behavior tree generation and adaptation for a social robot control with llms. Robot Autonom Syst 194:105165. https://doi.org/10.1016/j.robot.2025.105165
34. Mascaro EV, Lee D (2025) Robot behavior generation for social human-robot interaction. Intern J Soc Robot 2025(11):1–20. https://doi.org/10.1007/S12369-025-01333-3
35. Schrum M, Park CH, Howard A (2019) Humanoid therapy robot for encouraging exercise in dementia patients. In: 14th ACM/IEEE International Conference on Human-Robot Interaction, Daegu, Korea, pp 564–565. https://doi.org/10.1109/HRI.2019.8673155
36. Rossi S, Larafa M, Ruocco M (2020) Emotional and behavioural distraction by a social robot for children anxiety reduction during vaccination. Intern J Soc Robot 12:765–777. https://doi.org/10.1007/S12369-019-00616-W/FIGURES/6
37. Tani J, Mao X, Zeng Y, Zhao Y, Zhang T, Zhao D, Zhao F, Lu E (2020) A brain-inspired model of theory of mind. Front Neurorobot 14:1–17. https://doi.org/10.3389/fnbot.2020.00060
38. Maroto-Gómez M, Malfaz M, Castro-González A, Salichs MA (2023) A motivational model based on artificial biological functions for the intelligent decision-making of social robots. Memetic Comput 15:237–257. https://doi.org/10.1007/s12293-023-00390-3
39. Dell'Anna D, Jamshidnejad A (2024) Sonar: An adaptive control architecture for social norm aware robots. Intern J Soc Robot 16:1969–2000. https://doi.org/10.1007/s12369-024-01172-8
40. Cao HL, Perre GVD, Kennedy J, Senft E, Esteban PG, Beir AD, Simut R, Belpaeme T, Lefeber D, Vanderborght B (2019) A personalized and platform-independent behavior control system for social robots in therapy: Development and applications. IEEE Trans Cogn Developm Syst 11:334–346. https://doi.org/10.1109/TCDS.2018.2795343
41. Scassellati B (2002) Theory of mind for a humanoid robot. Autonom Robot 12:13–24. https://doi.org/10.1023/A:1013298507114
42. Dissing L, Bolander T (2020) Implementing theory of mind on a robot using dynamic epistemic logic. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20), Yokohama, Japan, pp 1615–1621. https://doi.org/10.24963/ijcai.2020/224
43. Vossen P, Baez S, Bajcetić L, Kraaijeveld B (2018) Leolani: a reference machine with a theory of mind for social communication. In: International Conference on Text, Speech, and Dialogue, Brno, Czech Republic, pp 15–25. https://doi.org/10.1007/978-3-030-00794-2_2
44. Lee JJ, Sha F, Breazeal C (2019) A Bayesian theory of mind approach to nonverbal communication. In: ACM/IEEE

International Conference on Human-Robot Interaction, Daegu, Korea, pp 487–496. https://doi.org/10.1109/HRI.2019.8673023
45. Benedetto A, Morrone MC, Tomassini A (2020) The common rhythm of action and perception. J Cogn Neurosci 32:187–200. https://doi.org/10.1162/jocn_a_01436
46. Buzsáki G, Draguhn A (2004) Neuronal oscillations in cortical networks. Science 304:1926–1929. https://doi.org/10.1126/science.1099745
47. Mahmoud MS, Singh MG (2012) Discrete Systems: Analysis, Control and Optimization, 1st edn. Springer, Heidelberg, Germany
48. Bishop CM, Bishop H (2024) Gradient descent. In: Deep Learning: Foundations and Concepts, 1st edn. Springer, Switzerland, pp 209–232. https://doi.org/10.1007/978-3-031-45468-4_7
49. Kramer O (2017) Genetic Algorithms Essentials, 1st edn. Springer, Germany
50. Lichess.org (2024) Lichess.org open database. Original dataset of chess puzzles. https://database.lichess.org/#puzzles
51. Patrício MLM, Jamshidnejad A. Data from experiments of "Leveraging Systems and Control Theory for Social Robotics: A Model-Based Behavioral Control Approach to Human-Robot Interaction". https://doi.org/10.4121/ccadc914-9502-46d6-9ba5-fef581f2933f
52. Duchetto FD, Baxter P, Hanheide M (2020) Are you still with me? Continuous engagement assessment from a robot's point of view. Front Robot AI. 7:116. https://doi.org/10.3389/FROBT.2020.00116/BIBTEX