International Journal of Social Robotics (2024) 16:1297–1314
https://doi.org/10.1007/s12369-023-00988-0

A Novel Mirror Neuron Inspired Decision-Making Architecture for Human–Robot Interaction

Mehdi Sobhani[1] · Jim Smith[2] · Anthony Pipe[1] · Angelika Peer[3]

Accepted: 15 February 2023 / Published online: 27 March 2023
This is a U.S. Government work and not under copyright protection in the US; foreign copyright protection may apply 2023

Abstract

Inspired by the role of mirror neurons and the importance of predictions in joint action, a novel decision-making structure is proposed, designed and tested for both individual and dyadic action. The structure comprises models representing individual decision policies, policy integration layer(s), and a negotiation layer. The latter is introduced to prevent and resolve conflicts among individuals through internal simulation rather than via explicit agent-agent communication. As the main modelling tool, Dynamic Neural Fields (DNFs) were chosen. Data was captured from human-human experiments with a decision-making task performed by either one or two participants. The task involves choosing and picking blocks one by one from seven wooden blocks to create an alpha/numeric character on a 7-segment. The task is designed to be as generic as possible. Recorded hand and blocks movements were used for developing DNF-based models by optimising parameters using a genetic algorithm. Results show that decision policies can be modelled and integrated with acceptable accuracy for individual performances. In the dyadic experiment, using only individual models without the negotiation layer, the model failed to resolve conflicts. However, with the implementation of a negotiation layer, this problem could be overcome. The proposed decision-making structure based on DNFs is developed and tested for a simple pick-and-place task. However, the main primitive underlying action of this task, pick-and-place, is indeed part of many more complex tasks people perform in their day-to-day life. Paired with the possibility to gradually evolve the architecture by adding new policies on demand, the architecture provides a general framework for modelling decision-making in joint action tasks.

Keywords Decision-making · Decision policy · Joint action · Human–robot interaction

1 Introduction

Efficient collaboration between two human agents requires them to have multiple capabilities such as perspective-taking [1], understanding affordances [2], forming of expectations of the next action [3], and timing ability [4,5] that all together form a social cognitive process. The cognitive process starts at the perception level when the agents assess the situation. By constantly monitoring the environment and partner(s) the agents can reach an understanding of the scene and predict the next action/event, which will allow the agents to choose the required action, also known as "decision-making", with the help of problem-solving and reasoning skills [6]. The agents may also communicate or interact non-verbally, if needed, to decide on a shared plan. The process and results may be stored for future use and allow remembering, reflection, and learning [7].

For smooth and seamless Human–Robot Collaboration (HRC) a robot also needs to be endowed with similar cog-

✉ Mehdi Sobhani
Mehdi.Sobhani@brl.ac.uk

Jim Smith
James.Smith@uwe.ac.uk

Anthony Pipe
Tony.Pipe@brl.ac.uk

Angelika Peer
Angelika.Peer@unibz.it

[1] Bristol Robotics Laboratory, University of the West of England, Coldharbour Lane, Bristol BS16 1QY, UK

[2] Department of Computer Science and Creative Technologies, University of the West of England, Coldharbour Lane, Bristol BS16 1QY, UK

[3] Faculty of Science and Technology, Free University of Bozen-Bolzano, piazza Università, 39100 Bolzano, Italy

nitive abilities, but today we are still far from achieving this goal as cognitive architectures for joint action tasks have rarely been investigated. Aiming for making a step towards endowing robots with human-like decision-making and negotiation abilities, in this paper we develop decision-making models based on the observed behaviour of human participants collaborating in a joint action task, considering that human-human interaction can provide a good model for human–robot interaction [8]. While we also model policies (a decision policy determines the strategy of the agent in making a decision) and integrate them, the main contribution of this paper is the proposition of a negotiation layer compliant with the rest of the overall presented architecture to resolve/prevent conflicts.

Our proposed architecture has the potential to increase efficiency as besides serial actions (taking turns) also parallel actions (working at the same time towards the same goal) are covered. Further, the architecture can easily evolve over time as new policies can be added on demand. The modular nature of the architecture makes it e.g. possible to add policy models like preferences and affordances to make the architecture applicable to a larger set of decision-making scenarios and not only rationalist ones as the decision-making could be based on a series of perceptual or experiential processes. This can be achieved by modelling decision policies separately and by integrating them only if required with the help of the Policy Integration layer as explained in Sect. 3.5.3.

We consider and compare several modelling approaches resulting in choosing the Dynamic Neural Field (DNF) as the main modelling tool. To develop our decision-making models and structure, a task was designed that is generically representative of many collaborative joint object manipulation tasks. The task involves choosing from a set of seven wooden blocks to create a character on a 7-segment. The pick-and place nature of the task is chosen as pick-and-place is part of day-to-day life when interacting with objects in different contexts from cooking together in the kitchen to working on an assembly line in manufacturing environments. Human hand and chest movements as well as the location of objects are being tracked and used as inputs to the model, see Sects. 3.1–3.4 for details. Depending on the decision policy different DNF structures are adopted as further explained in Sect. 3.5. Finally, the performance of the models is measured on training and test datasets (67/33 percent split) and compared to trained Artificial Neural Networks (ANN) on the same data with results presented in Sect. 4. These results are discussed in detail in Sect. 5 and on their basis conclusions and future work are formulated in Sect. 6.

2 Proposed Architecture and Related Work

Our proposed architecture for human–robot collaboration, as depicted in Fig. 1, is inspired by neuroscientific findings. Findings suggest that an agent runs internal simulations whenever s/he attempts to perform an action or whenever an action is observed while being performed by someone else [9]. Since the 1980 s the Simulation theory (ST), first presented by Gordon [10], along with other approaches like Theory theory (TT) and Rationality theory were competing to explain different aspects of human cognition. TT argues that people form a theory about their partner's mental states, Rationality theory uses rationality principles to achieve this, whereas ST suggests that people internally simulate their partner's mental state to reach a“pretend”state [11]. Simulation theory has gained additional support in explaining cognitive processes of human interaction after the discovery of mirror neurons [12,13]. ST has also inspired roboticists to develop cognitive architectures for safer [14] and more ethical [15] robots.

Particularly in the pre-motor cortex, two types of mirror neurons and canonical neurons have been found activated during action execution, imitation, or when only observing other agent's action. The mirror neurons were found to be activated during an action execution or observation with a specific goal, while the canonical neurons were found to be activated with the presentation of objects that afford goal-oriented actions [16,17]. Inspired by the role of the mirror neurons in joint action [18] and the fact that prediction is an essential part of this process, we propose a novel decision-making architecture. Considering the importance of prediction in the joint action process in which one's own action system is used to understand and interact with others [13,19] to enable an agent to form expectations about the next action of a collaborating partner, our proposed architecture foresees mental models of the decision-making processes of both the agent and of the interaction partner. Each agent has its own decision-making system that allows combining a series of independent individual policies by means of an integration layer. The two decision-making systems of the agents run in parallel through an internal simulation when collaborating on a joint action task and their outcome enters a negotiation layer. This layer is introduced to prevent conflicts in action execution by negotiating own independently taken decisions with anticipated partner's decisions. The latter is obtained by internal simulation of the mental model of the partner. So, each agent is assumed to simulate its own and its partner's decision-making process and to integrate the two independent decisions deriving from these processes into one final outcome.

The final decision on the next action is reached after both its own and the predicted partner's decisions are integrated in the negotiation layer. The negotiation layer works as an implicit communication, as after the actions of both agents are updated in real-time, these updated actions again trigger a new outcome of the internal simulations of both agents. Thus, if for example, both agents come to the same decision,

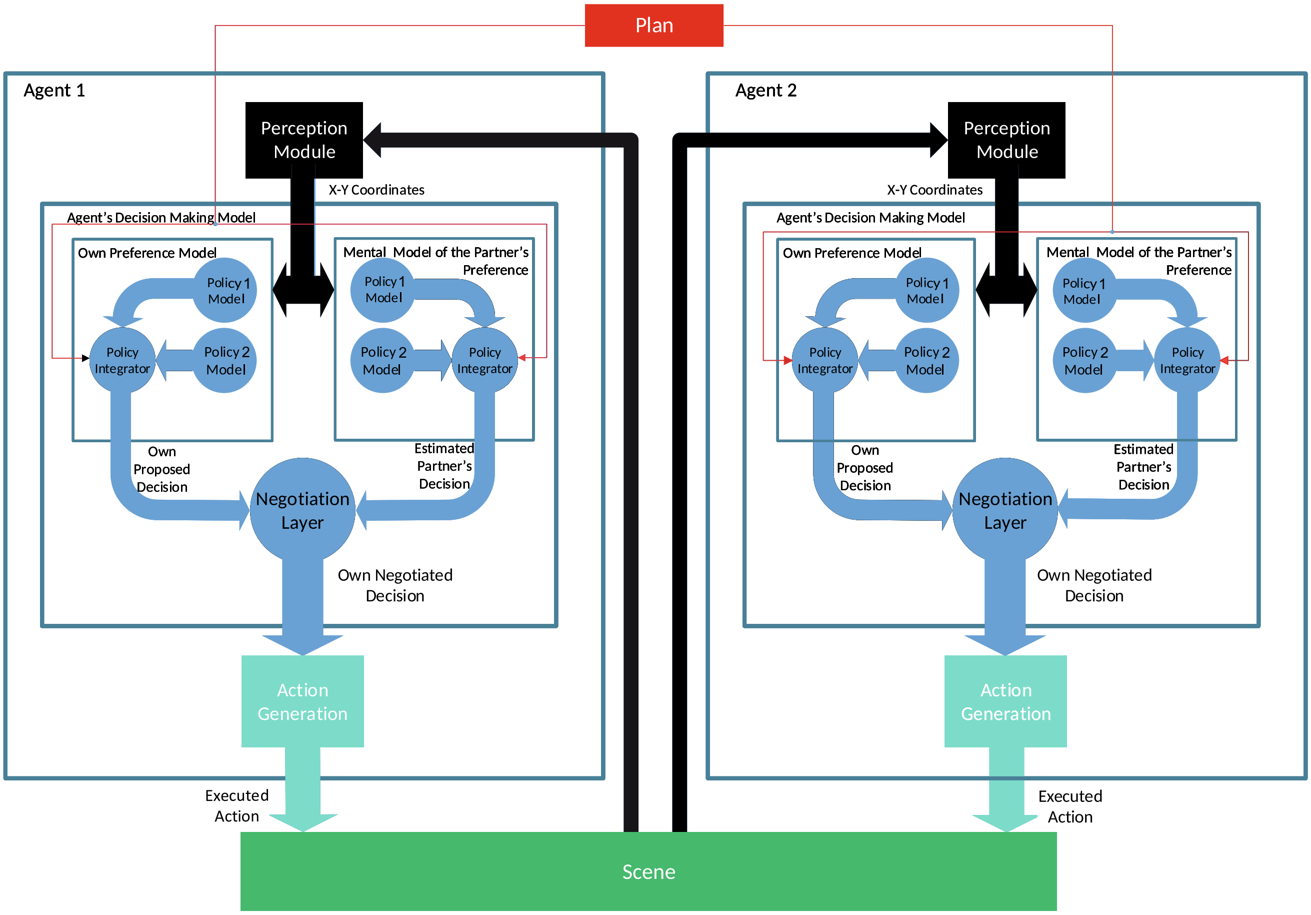


Fig. 1 Proposed decision-making architecture for joint action: an abstract depiction of the decision-making process of two agents. Each agent's decision-making process takes into account (i) its own preference model (including different policies and policy integrator); (ii) an internal simulation of its partner's decision model (inspired by mirror neurons); and a negotiation layer that combines (i) and (ii)

the one implementing the decision faster will have its action allowed to be executed, while the other will be prohibited to continue until the next foreseen action of the shared plan starts.

The perception module in the architecture represents any proprioceptive sensors that provide information on body movements as well as sensors that provide information on object movements. In developing our models, we utilised a Vicon motion capture system and Microsoft Kinect to track hand and object movements. However, depending on the complexity of the recognition system, other stereo vision or RGB-D cameras could be used. (The experimental setup and used tracking sensors are described in Sect. 3.1).

Action generation is considered to be a module that derives required action commands for action execution and is not considered the main topic of this research. It is assumed to be covered by standard motion planning algorithms available in the literature when, e.g., implementing the structure on a robot. It will receive the final decision and generates a series of commands to be send to the low-level control system of the actuators that then execute the individual actions.

The plan module is to implement a shared plan for joint action and differentiates between parts of the plan that can be executed in series or also in parallel. It also activates the related policy models or layers required for task implementation.

As Vinciarelli et al. [20] point out "mutual influences" in the interaction process have not been well investigated. Similar to our architecture, some of those reported in literature are based on the concept of internal simulation of the partner. For instance, Wolpert et al. [9], originally proposing only a structure for action production or action observation of individuals based on the MOSAIC model, argue that this concept of forward-inverse models used for modelling single motor control actions can be also applied to social interaction. However, the proposed architecture has never been implemented for a real interaction scenario involving joint action.

Bicho et al. [21], on the other hand, proposed a decision-making system for joint action based on Dynamic Neural

Fields (DNF), but decision policies were hard-coded rather than taken from human experimental data. They divided the workspace into two sides to be able to predict an action to be performed by a co-actor. Their assumption is that objects in the area closer to each actor (human or robot) will only be picked by the nearest actor. In our human-human collaboration experiments, however, people did not necessarily act in this way. Further, their decision-making system was only tested in joint action scenarios that involve serial actions with collaborators taking turns and performing complementary actions, hence significantly reducing potential conflicts, while our proposed architecture is developed for both serial and parallel actions. In this way, there will be no need for an actor to wait for the collaborator's action to be finished and if there is no physical constraint or limitations imposed by the shared plan, the actor can perform an independent action in parallel to the co-actor.

Further, there exists a large body of research that considers different kinds of cognitive models and aspects of human–robot interaction. However, only a few works consider joint action tasks. For instance, Sarthou et. al [22] proposed an architecture for perspective-taking where an agent (human/robot) instructs the other. By nature of the task, there is, however, no parallel action, and also the model does not predict the other agent's decision or action but provides only its perspective for the partner.

Another example is the mirror neuron model developed for learning grasping skills by Metta et. al [23]. It is important to note that in contrast to their work, we are not proposing a model of mirror neurons, but rather an architecture for human–robot interaction inspired by the role of mirror neurons. In addition, our work focuses on joint action with two agents rather than a robot that learns grasping from a human demonstrator.

Recently, Beraldo et. al [24] further proposed a decision-making structure that breaks down the process into decision policies. This work also lacks the joint action aspect of the interaction and has been developed for the teleoperation of a mobile robot using a Brain-Machine interface (BMI) based on Electroencephalography (EEG). In their architecture, there is no module for conflict resolution similar to our Negotiation Layer as reaching the same decision is desirable and is not considered a conflict. The robot follows the control command unless an obstacle needs to be avoided. In that case, the robot adjusts the path accordingly through "policy fusion" which may be considered related to our policy integrator.

There are also a series of cognitive architectures in the literature, like ACT-R [25], Soar [26] or R-CAST, which are based on Recognition Primed Decision (RPD) models [27]. What makes the proposed architecture different is that, while these architectures are developed for individuals, we present a model to be applied to joint action scenarios. More specifically we introduce a model that helps resolve conflicts in a joint action by means of a negotiation layer. Moreover, decision-making policy models as well as the integration and negotiation layer are developed based on a dynamical system (DNF), while aforementioned architectures are either based on declarative memory retrieval using instance-based models or rule-based (ACT-R) or probabilistic modeling approaches like decision trees (Soar). One of the most recent applications of the ACT-R architecture in modelling decision-making is presented by Zhang et al. [28] for Human-Computer Interaction (HCI). While they developed a dynamic model for "complex" interactions in HCI, their model only produces a prediction of the individual's decision, but the missing embodied nature of robots and the collaborative nature of the task with a shared plan make such a model not necessarily applicable to HRC and joint-action scenarios.

2.1 Selection of Mathematical Modelling Framework

Considering the dynamic nature of human decision-making processes, the desired mathematical framework for modelling the decision-making module has to be able to implement this dynamic and predictive nature of the processes. Although there are many dynamic probabilistic modelling approaches available in the literature, finding accurate probability information on human decision-making would require a large database. This makes the modelling based on such approaches difficult, if not impossible as it would require recordings of a large series of real human-human collaboration experiments. Thus, deterministic methods are preferred as then modelling requires a relatively smaller dataset. As argued by Kahneman and Tversky [29] people do not necessarily make rational decisions. So, to have a well-generalizing model, methods based on rationality assumptions are not suitable for this work. At the same time, the system should be able to cope with uncertainty and multiple alternatives while avoiding assumptions hence avoiding normative approaches. So, the features of the required modelling are categorised as:

- *Desirable*: Predictive, Multi Alternative, Dynamic, Coping with uncertainty
- *Undesirable*: Probabilistic, Normative, Static, based on Rationality assumption

To finally choose a proper mathematical modelling framework, some of the well-known techniques for implementing decision-making were reviewed as presented in the following paragraphs.

2.1.1 Decision Trees

Decision trees are one of the most popular decision support tools, a tree-like graph that starts from the decision that needs to be made and branches to the chance nodes and further sub-decisions and consequences of the decision in different situations. The value of uncertain outcomes O is calculated by multiplying the probability by the gained value of O.

Decision Trees (DT) have been used in many different fields, for example: i) corporate decision making; ii) Artificial Intelligence (AI) and machine learning for applications like decision support, regression, data mining; iii) path planning for mobile robots [30,31]. There has been an effort to make DTs as dynamic as possible [31]. However, they have been generally found to be not applicable when decisions have to be made for a continuously changing and dynamic environment. This is due to the fact that the general structure of the tree and the main consequences, including their probabilities of occurring, have to be known at the outset. This is not always possible for applications like HRI when human behaviour needs to be considered when establishing the DT structure.

2.1.2 Expected Utility and Prospect Theory

In classical economics, Expected Utility (EU) Theory is used in a descriptive way trying to explain *why* people make a specific decision. In philosophy, on the other hand, it is used as a normative theorem explaining how people *should* make decisions. The essence of the theory is that people are considered rational so they will make decisions to maximise the utility of the outcome of their action [32]. The action of the decision maker will be state-dependent and, since the states are uncertain, the expected value is calculated as a probabilistic weighted sum of the utility of outcomes of action in different states.

This theory has been popular in different disciplines to explain human decision-making. However, as will be explained in the following, EU theory has difficulties predicting human behaviour. In terms of its use in robotics, there is research on action planning using utility maximisation like [33] with reported improved performance of planning. Like DT this approach relies on knowing the probabilities of consequences and needs information on the task at the outset.

As Kahneman and Tversky [29] well pointed out, EU theory, as a descriptive or predictive theorem, is likely to fail when it comes to real-life decision making. Instead, they suggested Prospect Theory which tries to explain why people are not always rational and do not always make optimal decisions. The main idea of this theory is that people are neither always risk-averse nor always risk-seeking. They mostly seek risk when there is a high loss and mostly avoid risk when there is a high gain. This makes the value function, describing the value of an outcome, nonlinear in contrast to the linear one in EU theory. Having a steeper value function for losses means they have a higher effect than gains. Hence, in Prospect Theory the final utility gets lower as people give less value to higher gains by avoiding and not taking risks in such a situation. Conversely, when there is a high loss people tend to take higher risks and they give a higher value to the outcome compared to what seems to be the rational value. Like Utility Theory, Prospect Theory has been used in many disciplines to explain human behaviour and decision-making processes. Particularly in robotics, for example, it has been used to model human behaviour for assistive robots [34]. Prospect Theory is also relying on knowing probabilities of events and consequences which limits its use in highly dynamic environments. In addition, Expected Utility and Prospect Theory have been mainly used for two alternative tasks but increasing the number of alternatives may render the problem highly complicated.

2.1.3 Markov Decision Processes (MDP)

MDPs are a mathematical discrete stochastic model of decision-making. An MDP includes several states, in each of which the decision maker can choose from a pool of available actions. The probability of moving from one state to another is a function of the current state, so, the next state depends on the current one and the chosen action by the decision maker. The decision maker will receive a reward each time the process moves from one state to another [35]. The whole process relies on having complete knowledge of finite states and actions. MDPs have been used in several applications like economics, automated control, manufacturing, and robotics. A more generalised variant of MDPs are Partially Observable Markov Decision Processes (POMDP) in which the process does not have complete information on the current state (the current state is uncertain) and not all the states are completely known or“observable”. Similar to MDPs the transition to the next state is a function of the current state and current action. The main goal of both MDPs and POMDPs is to maximise the cumulative reward by optimising the policies for choosing actions.

Among all the probabilistic approaches, POMDPs have been used most in robotics as they can be applied in uncertain and dynamic environments. Examples can be found in control, planning, and navigation [36–39]. POMDPs have been applied to a vast range of fields like machine vision, business, corporate policy, and marketing. However, they can only deal with problems with certain characteristics such as having a finite state set and following the Markov Property (meaning that future states only depend on the current state and not past states). Also, it can be highly computationally expensive to assess all the rewards, transition probabilities, and observation probabilities [40] and thus, solutions are often approximated.

2.1.4 Decision Field Theory

Decision Field Theory was introduced by Busemeyer and Townsend [41], as a dynamical stochastic mathematical model of decision making, initially focusing on problems of approach-avoidance behaviour [42]. In contrast to normative theories, it tries to explain people's behaviour and decisions without a rationality assumption. The main feature of this theory is that it dynamically models the evolution of the decision during deliberation time rather than considering fixed states of preference. The theory is based on two main psychological principles namely, approach-avoidance in motivation theories and information-processing theories of choice response time [41].

Decision Field Theory (DFT) is considered both dynamic and continuous in time in contrast to previous theories which are discrete in time. The real-time DFT is defined by introducing a time variable h which is the time needed to process each sample of valence difference. This is equal to the time needed to process a pair of predicted consequences before switching attention to another pair. By having h approaching zero the preference state will be developed in a roughly real-time way. Although the initially presented formulation of DFT is for a two-alternative task, a multivariate DFT has been also presented in a connectionist interpretation way [43].

DFT has been applied to different cognitive processes like visual sensory detection [44] and conceptual classifications [45]. However, since DFT is in its nature a Markov process [41], Markov assumptions are assumed to hold. In contrast, human decisions may depend on past experiences. The main advantage of DFT over other decision-making models, however, is that DFT tries to explain the *process* of decision making rather than merely the end result, as it models the evolution of the decision during the deliberation time.

2.1.5 Dynamic Field Theory

Dynamic Field Theory was introduced by Schöner [46] based on the mathematical formulation of dynamic neural fields by Amari [47], as a framework for modelling cognitive processes like detection, selection or working memory. It combines the dynamics of attractors and repellers to form a dynamic behaviour, formulated as follows:

$$\begin{aligned}\tau \dot{u}(x,t) &= -u(x,t) + h + S(x,t) \\ &\quad + \int w(x-x')\sigma[u(x',t)]dx', \qquad (1)\\ \sigma(x) &= \frac{1}{1+e^{-\beta x}}, \qquad (2)\end{aligned}$$

where τ is the time scale, u the activation function over the feature space x at time t, $h < 0$ a constant resting level, S an external input or stimulus to the field, and the integral part is to drive lateral interaction in the population with $w(x-x')$ as interaction kernel and $\sigma[u(x',t)]$ a sigmoidal nonlinear threshold function with a scaling parameter β. Depending on the type of interaction kernel, the nature of the interaction can change from global inhibition to local inhibition or global excitation. This property is being used to model different cognitive processes. A global inhibition, for example, is used for the selection process to achieve a stable choice with minimised effect of environmental noise on the process so that unless the target is shifting to another alternative, the choice won't change due to small environmental perturbance, or a local inhibition is required for a detection process in which the neural field needs to be inhibited in the immediate vicinity of the point of interest so that it stands out of the neighbouring points. These are achieved through the interaction kernel as depicted in Fig. 2. As can be seen from the equation, the interaction is computed through convolving the interaction kernel with a sigmoidal threshold of the activation function.

The lateral interaction kernel is the key player in changing the behaviour of the dynamic model. One common formulation of the kernel is the following exponential equation:

$$w(x) = c_e x^{\frac{-x^2}{2\sigma_e^2}} - c_i x^{\frac{-x^2}{2\sigma_i^2}}, \qquad (3)$$

where subscript e stands for the excitatory and i for the inhibitory part of the kernel. By changing values of c_e and c_i the excitatory or inhibitory effects of the kernel can be varied. Similar to a normal bell curve σ_e and σ_i are to adjust the bell shape. In Fig. 2 three cases of interaction kernels are depicted, the green curve is for modelling working memory, the blue one is to model a detection mechanism and the red one is to model a selection process in the neural field. Neural Field Theory has been used in the field of cognitive science to model sensorimotor decisions [48], visual cognition [49], modelling object localisation in the visual cortex [50], modelling visual perception [51] and action understanding [52]. In terms of robotics applications, Dynamic Field Theory has been applied to areas like navigation [53], aspects of human–robot interaction and collaborations like action understanding through imitation [54], object recognition [55], verbal and non-verbal communication [56], decision making and joint action for human–robot collaboration [21,57].

2.1.6 Summary of Comparison of Mathematical Frameworks

Characteristics of mathematical modelling frameworks introduced so far are summarized in Table 1. In this table, when

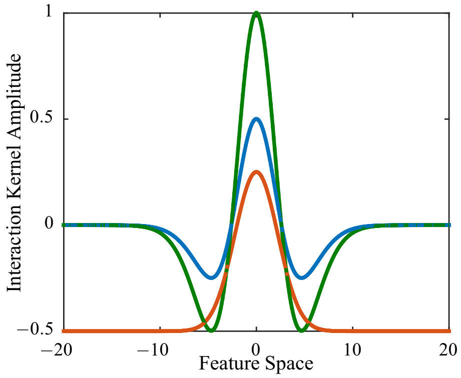


Fig. 2 Examples of interaction kernels: the green curve is for modelling working memory, the blue one is to model a detection mechanism and the red one is to model a selection process. (Color figure online)

Table 1 Comparison of decision-making modelling methods

	Decision Tree	EU	Prospect Theory	POMDP	DFT	DNF
Probabilistic	✓	✓	✓	✓	✓	X
Normative	✓	✓	X	X	X	X
Predictive	X	X	✓	✓	✓	✓
Multi Alternative	X	X	X	✓	✓	✓
Static	✓	✓	✓	X	X	X
Dynamic	X	X	X	✓	✓	✓
Coping with uncertainty	X	X	X	✓	✓	✓
Rationality Assumption	✓	✓	X	✓	X	X

Red: undesirable, green: desirable, ✓: feature available, X: feature unavailable

a method has a feature, it is shown by a ✓, and if it doesn't with an X; colours red and green mean undesirable and desirable, respectively. Apart from Dynamic Field Theory, the reviewed approaches are mainly probabilistic methods requiring information on the probability of actions and consequences. While Expected Utility theory-based methods require static and completely known problems in terms of alternatives and consequences, Markov Decision Processes, particularly POMDPs, Decision Field Theory, and Dynamic Field Theory are applicable to uncertain and dynamic problems. Most of these approaches have been initially developed for two-alternative forced choice tasks, however, some like POMDP, DFT, and Dynamic Field Theory (DNF) can be extended to multivariate alternatives. As can be seen from Table 1, Dynamic Neural Field (DNF) has all the desired requirements for developing a dynamic decision-making model, even when only small datasets are available.

Further, although it is possible to create decision-making models based on other frameworks with an extra layer of data processing; the input data needs to be properly coded so that the model can produce the required outcome like a prediction of the next action/decision, most mathematical models may fail to capture the embodied dimension of the mirror neurons. In this context, however, DNFs bring a clear advantage since they have the potential to map the physical world on a dynamic neural field (see also section 3.5). All these reasons make DNFs the best choice for our application.

3 Method

To model decision policies and develop the initially-introduced decision-making module, we designed a series of experiments to have either human individuals or dyads to work on an instructed task so that we could observe their behaviour and record data required to modelling their decision processes. Ethics approval for these experiments was obtained from the ethics committee of the University of the West of England (reference number: UREC16-17.03.10).

3.1 Experimental Setup

Participants were asked to work together in a table-top pick-and-place task. Participants were monitored and a set of data consisting of tracked 3D hand and chest position were recorded using a Vicon motion capture system. To have a clear baseline for evaluation, participants were instructed to perform the task in a particular way, following a specific pre-defined policy as introduced further below. The blocks were equipped with Augmented Reality (AR) markers and their motion was captured using a Kinect sensor. The experimental setup is depicted in Fig. 3.

3.2 Task

The chosen task was designed considering certain requirements. The task was supposed to capture an aspect of day-to-day life and be able to be completed by either individuals or pairs. It should be of an abstract level and be extendable or generalisable later on to more complex tasks. Also, as the focus of the work is on the process of decision-making, the task should be as simple as possible to not require any other cognitive processes like problem-solving adding cognitive load, which might affect participants' decision-making.

Taking these considerations into account, the task was chosen as follows: Participant(s) were asked to use provided coloured blocks to form some alpha/numeric characters on a 7-segment template, according to a provided instruction. In terms of experiments with an individual participant, each

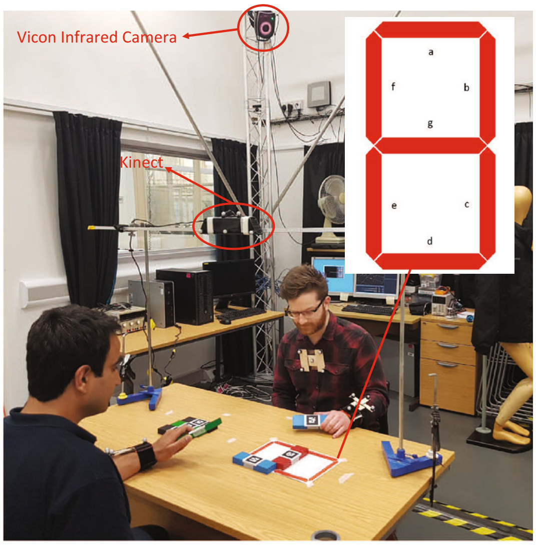


Fig. 3 Experimental setup. Two subjects collaborating in the dyadic condition. The 7-segment shown on the top right is placed on the table horizontally so that segment "d" comes to lie at the center of the table. The subjects have markers attached to their hands and chest that are captured by the Vicon tracking system. The AR tags on the blocks are tracked by the Kinect camera

person was asked to pick blocks one by one and place them on the marked 7-segment shape. In the case of joint action, each participant was asked to pick blocks one by one, being told to "work together", but leaving them the freedom to either take turns or pick blocks at the same time.

3.3 Instructions and Procedure

Before starting the experiment, participant(s) were made familiar with the setup and handed out an information sheet that described their task. They were asked to sign a consent form and were informed that they can withdraw their participation at any point. Then, the participants' right/left wrist and chest were marked with motion capture markers. In each trial participant(s) were asked to perform and complete characters "H","3","E","9","6","2","5","8" on the 7-segment shape using provided coloured blocks. For instance, a number 9 could be formed by covering segments "a","b", "c","d","f" and "g". These characters were chosen to counter-balance any effect of the blocks' final position on the outcome (e.g. H and 8 are symmetric while others are mirrored with respect to different axes). To counter-balance any effect of blocks' initial positions, blocks were randomly placed in the middle of the table by the experimenter after each character was formed, with initial orientation changing between vertical or horizontal placement (Fig. 4). Blocks' in-the-line position was also randomised. Participants were asked to sit on either side of a table with minimum possible movement to perform the calibration of all tracking systems assisted by the experiment conductor. Participants were provided with a set of blocks marked with AR markers (Augmented Reality markers) and were instructed to grab each block in a way not to cover the markers. Participants were also asked to pick and place blocks one by one and to follow the predefined policy. Participants' movements were recorded when performing the task. In terms of the collaborative phase, participants were asked to always place the first block on the central segment. This is to have both serial (meaning participants take turns in picking up blocks) and parallel actions as it was observed in pilot experiments that some participants performed the task only in a parallel manner (picking up blocks at the same time).

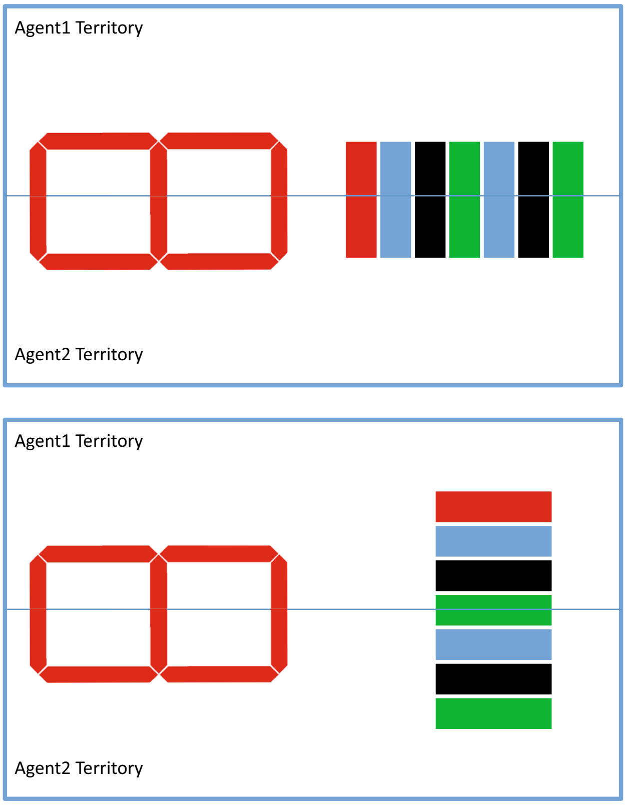


Fig. 4 Blocks' initial position; the initial position was rotated 90 degrees for every other character in the task meaning if for the current character blocks were initially placed horizontally (top image) for the next they were aligned vertically (lower image)

In two separate experiments for individuals and dyads the following four conditions have been tested:

1. *Distance policy*: Each person has been asked to only pick the closest block.
2. *Colour policy*: Each person has been asked to pick blocks according to an order of colour irrespective of their physical position, i.e., no matter where the block is located the order of colours should be applied first. In terms of the

dyadic experiment, each participant is given the same order of colours.

3. *Colour and Distance*: Participants have been asked to follow a colour order and at the same time pick the closest block when there is more than one block of the same colour. The data captured in this condition was used to develop the policy integration layer.
4. *Uninstructed*: In this condition, participants are free to pick blocks without being instructed to follow any order.

3.4 Experimental Design and Participants

A between-subject design was chosen to avoid carry-over effects from one condition to the other. In the first phase of the experiment, 60 individuals took part, 15 people per condition, 44 of which were male and 16 were female. The average age of participants was 28.78 (SD 5.7) ranging from 20 to 48 years old with an average height of 175.4 cm (SD 8.63) ranging from 157 to 191 cm. In terms of handedness, 49 participants were right-handed and 11 were left-handed. All participants reported normal or corrected to normal vision (18 wearing glasses). In the second phase, 96 people in 48 pairs took part, 12 pairs per condition, having an equal number of male and female participants. The average age of participants was 30.92 (SD 10.87) ranging from 18 to 67 years old with an average height of 171.27 cm (SD 10.26) ranging from 148 to 193 cm. For handedness, 85 people were right-handed, 11 were left-handed and 2 reported being dual-handed but used their right hand in the experiment. Participants formed 16 male only, 16 female only and 16 male/female dyads, 37 pairs of both participants being right-handed and 11 pairs of mixed right and left-handed ones. All reported normal or corrected to normal eyesight (44 wearing glasses).

3.5 Neural Field Structure

3.5.1 Structure for Distance policy

For modelling *Distance policy*, the table-top setup of the experiment is mapped into a 2D DNF:

$$\tau\dot{u}(x, y, t) = -u(x, y, t) + h + S(x, y, t) + \iint w(x - x', y - y')\sigma[u(x', y', t)]dx'dy'$$

with

$$w(x, y) = C_{gi} + C_e e^{\frac{-x^2-y^2}{2\sigma_e^2}} - C_i e^{\frac{-x^2-y^2}{2\sigma_i^2}}, \tag{4}$$

where τ is the time scale, u the activation function over the feature space x (mapped to the length of the table) and y (mapped to the width of the table) at time t, $h < 0$ a constant resting level, S an external input or stimulus to the field (with one stimulus for each block and participants' hands). The integral part is to drive lateral interaction in the population with w as interaction kernel and σ a sigmoidal nonlinear threshold function with a scaling parameter β similar to (Eq. 2). In the interaction kernel, $w(x, y)$, subscript e stands for the excitatory and i for the inhibitory part of the kernel. By changing values of c_e and c_i the excitatory or inhibitory effects of the kernel can be varied. Similar to a normal bell curve σ_e and σ_i are to adjust the bell shape and C_{gi} decides the amplitude of the global inhibition.

The projected position of the centre point of each block and the wrist position of the participants' wrists is mapped on the x-y plane. The x and y axes are then used as features so that each x-y coordinate of the blocks and wrist is considered the position of an input stimulus to the neural field. Each stimulus is modelled with a 2D Gaussian and the interaction of these stimuli (through lateral interaction shown as integral part of (Eq. 4) changes the field activation level in different locations as the input stimuli change due to the agents' motions. The parameters to be learned for this setting are mainly interaction kernel parameters (c_e, σ_e, c_i, σ_i, β, c_{gi}). Having properly trained the model, interaction kernel parameters can change the neural field behaviour such that the response to stimuli will result in an activation of the field at the point of interest, respectively the location of the chosen block.

3.5.2 Structure of Colour policy

The *Colour policy* model, is a 1D DNF coupled with a memory trace (Eq. 5) having only colours as stimuli:

$$\tau_p\dot{P}(x, t)) = \lambda_{build}(-P(x, t) + f(u(x, t)))f(u(x, t)) - \lambda_{decay}P(x, t)(1 - f(u(x, t))). \tag{5}$$

where τ_p is the time scale, $P(x, t)$ is the strength of memory at point x of the DNF with $u(x, t)$ as its activation function and f is a sigmoid function. λ_{build} and λ_{decay} determine the rate of build-up and decay of the memory trace [58]. The training for this structure is like memorising the colour order by demonstrating the order and showing blocks one by one. The memory then forms pre-shapes for the colour order. This model structure is similar to the work by Sandamirskaya and Schöner [59] but implemented in a way that the neural field stays activated to wait in the order until all blocks of the same colour are removed by the participant(s) before moving to the next colour in the order. This is done to simulate tasks with an equal priority of actions in the plan. The parameters of the Colour Policy DNF are chosen to be the same as the ones reported in [59].

3.5.3 Structure of Policy Integrator

This layer of architecture plays an important role in future expansion. Having different policies modelled separately, and integrated through this layer, makes the architecture adaptable to different tasks. For the task at hand, to have a correct prediction on the chosen block, the colour policy model is coupled with the distance policy model. This provides a measure to decide when there exist multiple blocks of the same colour. This means that the colour policy creates a short list of the blocks to be picked and the distance policy model predicts which one will be picked up. This is done by having a DNF similar to the *Distance policy* with only shortlisted blocks as stimuli being implemented and the final block is chosen from the shortlist according to the distance policy. This process occurs naturally in the DNF of the policy integrator as the amplitude of the input stimuli from the output of the colour policy and distance policy models will intensify the neural field activation for the chosen block.

3.5.4 Structure of Negotiation Layer

A simulation of the predicted partners' actions runs simultaneously with the 'own' model in the negotiation layer. This simulation is to adjust the own decision to the predicted partner's decision accordingly to prevent any conflicts like picking up the same object. This will also adjust the decision based on the plan, so, if the model predicts that the partner would perform the next step, like when a partner reaching quicker to an object, the agent should either move on to the next action or wait for the appropriate moment to perform the next action. This is done by inhibiting own decisions when the model predicts that the partner will perform the same action, or excite the decision when it predicts that the partner is waiting or performing another action. To achieve this, the interaction kernel ($W(x, y)$ in Eq. 4) of two DNFs of the own agent model and the partner model, is adjusted based on the human-human interaction experiment. This means the desired outcome is achieved by learning when each DNF should be inhibited (activation function being locally or globally deactivated) or excited (activation function either locally or globally being further activated)

3.6 Training Method

The recorded data from individual/dyad participants completing the task was used for training policy models as well as integration and negotiation layers. The training was fairly time-consuming as it took one month to optimise all parameters for the individual phase and as layers added up it became even more computationally expensive. The last training of the negotiation layer for the colour and distance condition took three months.

In the following, we briefly explain the procedure adopted for training, which was applied to all policies (colour, distance, colour and distance) as well as the integration and negotiation layer. The recorded data was split into training and test datasets by randomly choosing the data from two third of the participants for training and the rest for test. With 15 participants for one condition, for example, data from 10 participants was used for training and the rest for testing. All the structures of Sect. 3.5 were implemented in MATLAB using the DNF toolbox COSINIVA [60]. Parameters of the DNF models were optimised such that it resulted in the desired activation of the field at the correct position and time in the feature space. Basic work on how to train DNFs focuses on gradient-based methods for a local search or evolutionary algorithms for a global search as reported in [61]. We adopted a Genetic Algorithm (GA) [62] for a global search. The recorded data was used as input to the network. Information about the picked block and the predicted decision by the model was used to calculate the error over the whole captured data. To make the results comparable, we used the same error equation (E_t) for all the policies:

$$E_t = \begin{cases} 1 & \text{if} \quad \text{Block}_{p,t} \neq \text{Block}_{n,t} \\ 0 & \text{if} \quad \text{Block}_{p,t} == \text{Block}_{n,t} \end{cases} \tag{6}$$

$$E = \frac{\Sigma E_t}{N}, \tag{7}$$

where $Block_{p,t}$ is the predicted to-be-next-picked block evaluated at time t by the model and $Block_{n,t}$ is the next picked block evaluated at time t. The overall error E is then calculated by the average of errors over time. In total, 9 parameters for each field are tuned by the GA: τ, h, c_e, σ_e, c_i, σ_i, β, c_{gi} and σ_w (width of the Gaussian stimuli for the participants' hands). The population size was 200, with stochastic Uniform selection, Scattered crossover and Adaptive Feasible mutation. The training was considered to be finished when the stall generation limit of 50 was reached. The training was performed using pure global coordinates for wrist and blocks. This computation was done in parallel on an HP Z640 14-core machine with 64 GB memory (RAM).

It is worth noting that the training of the negotiation layer was done based on observed human behaviour during joint action. Participants were observed, completing the task, namely taking turns (serial actions) or at the same time (parallel actions) and in few cases a mixture of both serial and parallel actions. When the DNF was optimised for serial actions, the amplitude of global inhibition was larger, while for the parallel actions, it was smaller and the amplitude of local inhibition was larger compared to the serial actions. For a definition of global and local inhibition please refer to Sect. 2.1.5. Consequently, a training set was formed by mixing data from 3 trials with participants having mainly serial actions and 3 trials with participants having mainly parallel actions.

4 Results and Model Validation

Validation of the trained model was performed using a binary performance measure similar to (7):

$$P_t = \begin{cases} 0 & \text{if} \quad \text{Block}_{p,t} \neq \text{Block}_{n,t} \\ 1 & \text{if} \quad \text{Block}_{p,t} == \text{Block}_{n,t} \end{cases} \tag{8}$$

$$P = \frac{\Sigma P_t}{N}, \tag{9}$$

where $Block_{p,t}$ is the predicted to-be-next-picked block evaluated at time t by the model and $Block_{n,t}$ is the next picked block evaluated at time t. The value of P should be close to zero for a trained model well-fitted to the data, assuming that subjects perfectly followed the instructed policy. This measure was used for all the models to have a meaningful comparison.

4.1 Validation Performance

The trained models were tested using separate test sets created randomly from the recorded data of one-third of the participants. The achieved model performance and optimised DNF parameters are reported in Tables 2, 3 and 4. As can be seen from Table 3, DNF parameters of the negotiation layer when being trained for the Colour policy are the same as for the Distance and Colour condition with serial actions. This indicates that in the Colour only condition, the majority of the participants were taking turns in picking the blocks and serial actions are the main form of interaction. In addition to the DNF activation function and interaction kernel parameters, σ_w representing the width of the Gaussian stimuli for the wrist of participants is presented in this table. While its value has been in the same order for all conditions, for the Distance policy of dyadic experiments the width was found to be much smaller. This is due to requiring higher precision as participants might pick two blocks next to each other. This is why this parameter was also optimised, while the width of Gaussian stimuli for the blocks remained at a constant value of 0.5.

To compute performance measures, the developed models were applied to the recorded data. For individual models, only data from individual experiments was used for training, while for testing data obtained in dyadic experiments was used. In addition, the last row in Table 4 shows the performance of the system without a negotiation layer. These numbers were computed by adopting trained individual models for each agent and applying them to all data of the dyadic experiments. The system, in this case, has a relatively low accuracy for colour and colour and distance conditions and shows slightly better performance for the distance condition as each participant picks up the closest block hence reducing the potential conflicts (unless participants of a dyad are of opposite handedness).

As an example, snapshots of the activation function of the 2D DNF mapped on the table-top along with “block layover” are also depicted in Figs. 5, 6, and 7 to demonstrate how these models work. The small sphere represents a participant’s hand and the ellipse is their upper torso position. The lines for the upper and lower arms are drawn approximately as there has been no tracking information for the elbows or shoulders. Figure 5 and 6, show activation of the neural field and the peak on the approached block, meaning that it is predicted to be picked up by an individual participant. Figure 7 is for the same experiment, above showing the DNF activation for participant 1 and below showing the DNF activation for participant 2, respectively. As can be seen, when participant 2 is approaching the blue block, the DNF of participant 1 is inhibited (no activation peak) and the DNF for participant 2 has an activation peak over the blue block, meaning it will be picked up by participant 2.

4.2 Comparison to Artificial Neural Networks (ANN)

An all-in-one approach aiming for learning the decision without breaking down the process into policies using a single DFT model failed at the early stages of our research. Thus, we decided to model policies separately as also indicated in the overall architecture, resulting in the proposed gray-box model. But to also test the DNF and the developed architecture against another black-box technique which has been used in many fields of machine learning including decision-making we further decided to compare it to Artificial Neural Networks (ANN). For this purpose, a multi-layer perceptron (MLP) was implemented with h hidden layers of d nodes with *ReLU* activation, *l2-norm* batch normalisation with a regularisation penalty $l2$, dropout at rate d_r, batch size b, using the “Adam” optimiser with early stopping when the accuracy on a 10% validation set had not improved for 20 epochs. To tune these hyper-parameters, we used a random search in the space of all combinations of hyper-parameters as defined by our grid of possible values controlling topology, batch size, normalisation, and dropout. Scikit-learn’s Randomised Search method with 5-fold cross validation over 500 iterations was used to tune the meta-parameters, thus in total 100 runs of 500 epochs were used to tune the MLP hyper-parameters, resulting in $h = 4$, $d = 64$, $l2 = 0.01$, $d_r = 0.2$ and $b = 512$.

Unlike the DNF, the MLP model does not have state, so is making an independent prediction at each sampling time-step. However, in practice, the users’ intentions only change periodically, and far slower than the observation

Table 2 Performance of the policy models for the individual experiment

Individuals experiment		Condition			
		Distance	STD	Colour and distance	STD
Individual	Training set	89.52%	3.72	86.7%	2.94
	Test set	84.69%	4.28	86.26%	3.08

The standard deviation is calculated by computing the accuracy of the model for each participant and then calculating the standard deviation of these values (the accuracy scale is from 0 to 100%)

Table 3 DNF parameters of developed models for individual and dyadic experiments

Individuals experiment	DNF parameters								
	h	τ	β	c_e	σ_e	c_i	σ_i	c_{gi}	σ_w
Distance	−2.99	19.44	9.52	25.49	1.01	11.02	3.59	−0.54	34.77
Integrator	−2.93	18.94	10.78	25.01	0.88	12.52	3.59	−1.43	36.77
Colour	For parameters see work by Sandamirskaya and Schöner [59]								
Dyads experiment	DNF parameters of the negotiation layer in each condition								
Distance	−3.10	17.06	2.28	3.62	16.59	9.89	21.91	−14.94	0.79
Colour	−8.23	17.36	1.55	12.40	16.38	9.12	20.36	−13.34	38.30
Distance and colour (all actions)	−7.19	13.04	1.29	20.03	10.82	5.35	23.60	−0.22	34.20
Distance and colour (only serial actions)	−8.23	17.36	1.55	12.40	16.38	9.12	20.36	−13.34	38.30

Parameters of the distance policy and the integrator (colour and distance) are trained from the individual experiment presented in the first two rows. Parameters of the negotiation layer for each condition were trained based on the data from the dyadic experiment presented in the third to sixth rows. The two last rows present two sets of parameters for the negotiation layer of the distance and colour condition for all actions (fifth row) and the last row for trials in which participants were taking turns performing serial actions

Table 4 Performance of proposed system with and without the negotiation layer in different conditions of the dyadic experiment

Dyads experiment		Condition					
		Distance	STD	Colour	STD	Distance and colour	STD
With negotiation layer	Training set	88.46%	5.83	84.64%	2.89	85.31%	2.90
	Test set	80.57%	2.37	83.58%	1.88	81.39%	3.00
Without negotiation layer		72.01%	11.72	57.67%	9.75	65.5%	8.1

No training was done for “Without Negotiation Layer” and the performance result is based on all recorded data. The standard deviation is calculated by computing the accuracy of the model based on individual participants and then calculating the standard deviation of these values (the accuracy scale is from 0 to 100%)

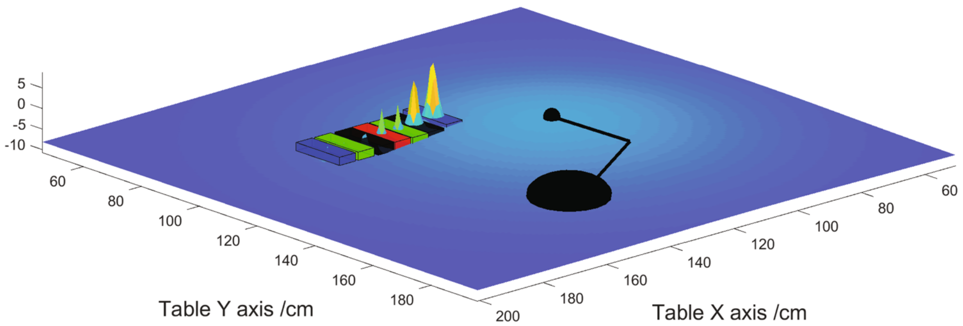


Fig. 5 Snapshot of 2D DNF activation mapped on the tabletop and overlaid blocks when the participant is approaching the first blue block on the right in an individual trial. (Color figure online)

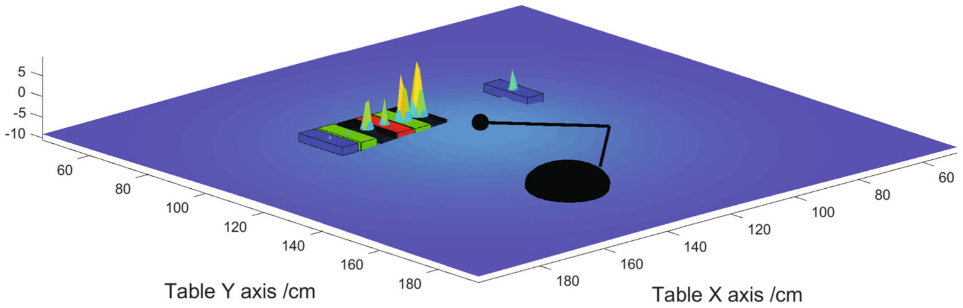
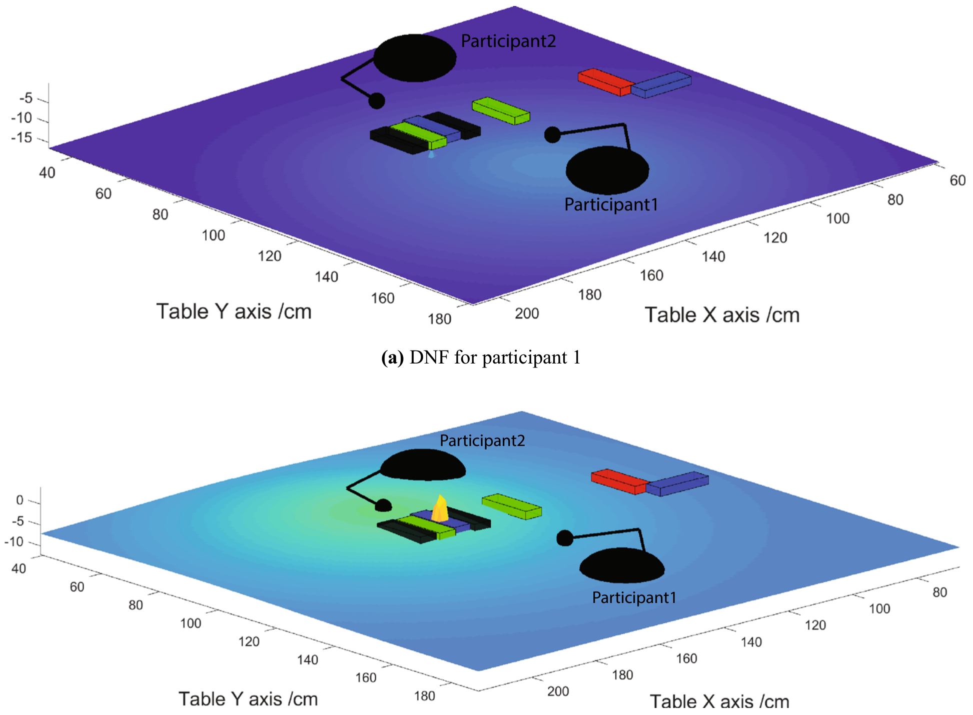


Fig. 6 Snapshot of 2D DNF activation mapped on the tabletop and overlaid blocks when the participant is approaching the black block after placing the blue block in an individual trial. (Color figure online)



(a) DNF for participant 1

(b) DNF for participant 2

Fig. 7 Snapshot of 2D DNF activation for participant 1 (above) and participant 2 (below) mapped on the tabletop and overlaid blocks when participant 1 and participant 2 are approaching a blue block at the same time. The neural field of participant 2 is activated with a peak over the blue block predicting this block will be picked up since participant 2 was moving faster than participant 1, while the activation for participant 1 was inhibited. (Color figure online)

sampling rate, so access to "memory" could be beneficial. Therefore, we also implemented an LSTM recurrent neural network (RNN), presenting it with a sliding window of samples $(x_{t-w}, x_{t-w+1}, \ldots, x_t)$ from which to predict x_{t+1}. Tuning the meta-parameters, in particular for the batch and window size w proved excessively computationally expensive, so after initial experimentation a topology of two layers of 50 LSTM nodes followed by a single dense layer of 64 nodes, and $d_r = 0.25$ was used, with "data" values of $w = 8$, $b = 512$. The $ReLU$ activation functions was chosen as in our experience they work well on a range of non-image problems. We also did some preliminary experimentation with the use of alternative activation functions such as sigmoid and tank, but the results were not promising.

The results from the two ANN's are shown in Table 5 and further compared to the DNF performance in Table 5. Please note that the accuracy is similar in most conditions for training data but the ANN model is not comparable for the Colour condition as the DNF model memorises the order and has 100% accuracy in individual experiments (Table 6).

As can be seen, despite having "memory", the recurrent network did not always achieve the same level of training accuracy as the more basic MLP, and the gap between accuracy on the training and test sets was typically far greater—a classic sign of "overfitting". The MLP also displays some signs of overfitting and never reaches the test accuracy observed with the DNF. Possible reasons for this are as follows:

- The form of the model: the MLP does not have state, and so cannot take advantage of the differences between the rate of decision-making and sampling. This should have been ameliorated by the use of LSTM nodes in the early layers of the recurrent model—and was for all but the individual-distance combination.

Table 5 The performance of the developed ANN models. The standard deviation values were calculated over 10 runs (accuracy scale is 0 to 100%)

ANN results			Condition					
			Distance	STD	Colour	STD	Distance and colour	STD
MLP	Individual	Training set	79.08%	3.27	86.62%	3.44	92.23%	1.04
		Test set	43.92%	2.32	77.99%	1.93	85.18%	1.38
	Dyad	Training set	75.25%	3.19	89.54%	0.76	90.30%	2.05
		Test set	45.08	2.14	67.24%	2.31	73.69%	2.64
RNN	Individual	Training set	61.82%	1.18	78.67%	1.23	79.71%	0.71
		Test set	24.72%	1.35	64.32%	0.81	63.68%	1.02
	Dyad	Training set	59.21%	5.26	81.84%	0.34	85.89%	0.82
		Test set	18.76%	2.75	67.81%	1.39	46.61%	0.88

Table 6 Comparing performance of DNF, MLP and RNN in different conditions of the individual and dyadic experiments

Condition		Accuracy (%)		
		DNF	MLP	RNN
Distance				
Individual	Training set	**89.52**	79.08	61.82
	Test set	**84.69**	43.92	24.72
Dyad	Training set	**88.46**	75.25	59.21
	Test set	**80.57**	45.08	18.76
Colour				
Individual	Training set	**100**	86.62	78.67
	Test set	**100**	77.99	64.32
Dyad	Training set	84.64	**89.54**	81.84
	Test set	**83.58**	67.24	67.81
Colour and distance				
Individual	Training set	86.7	**92.23**	79.71
	Test set	**86.26**	85.18	63.68
Dyad	Training set	85.31	**90.30**	85.89
	Test set	**81.39**	73.69	46.61

Bold type indicates highest value in each row, but we do not assert statistical significance (accuracy scale is 0 to 100%)

- Insufficient computational budget for meta-parameter tuning and training: direct comparisons are difficult as the MLP and RNN were built on a 48-core processor exploiting two fast GPU cards. However each was given a week's runtime, which approximately equates to the month on a slower machine for the DNF.
- The greater complexity (number of weights to learn) is greater than the DNF and consequently requires more training data to avoid overfitting. This should have been ameliorated by the use of early stopping. Nevertheless, both MLP and RNN have several categorical meta-parameters to tune, followed by thousands of continuous-valued weights for which values must be optimised.
- The algorithm used to optimise the model parameters: both used stochastic algorithms, but the DNF parameter space was searched using the *global* search of an evolutionary algorithm, whereas because of the far greater search space size, the MLP and RNN networks used a variant of *local* search (gradient descent).

In summary, the evidence suggests that the MLP is outperformed by the DNF due to its lack of memory. For some tasks, the RNN approached or bettered, the accuracies of the DNF in its performance on the training data. However, to an even greater extent than the MLP, it suffered from "overfitting", so that predictive accuracy on unseen test data was poor. Given greater computational budget for tuning the meta-parameters (such as network size, "early-stopping" rule) could possibly have been improved. However, the greater simplicity of the DNF makes global optimisation feasible, and reduces the likelihood of overfitting, making it faster and simpler to tune to a good level of accuracy for different tasks.

5 Discussion

The proposed decision-making structure based on DNFs is developed and tested for a simple task. However, the main primitive underlying action of this task, pick-and-place, is indeed part of many more complex tasks people perform in their day-to-day life. Hence, we strongly believe that it is possible to apply this structure also to more complicated tasks, which needs to be still proven in our future work. There might be a need to integrate more policy models though and to eventually also update them over time. This modularity, however, is considered a clear advantage of our architecture as it allows evolving it over time by adding or removing required policies for different tasks with different degrees of complexity. On the other hand, as pick-and-place is a natural part of many collaborative tasks, we consider it reasonable to re-use the already trained *Distance policy* without any need for further training and to extend the idea of the *Colour policy* to any preferred order of actions. Hence, there would be only a need to train further involved policies as well as the integration layer. We further assume the negotiation layer would not need to be re-trained as long as the main nature of the task remains a pick-and-place process, each agent is performing an action either in series or in parallel to the other agent to achieve the shared goal and there is no shared action like carrying a large object together as well as the negotiation layer has been trained with a sufficiently rich dataset to capture a large series of eventual cultural or personal preferences. Otherwise, offline re-training or better online updating of the negotiation layer may be required. All these assumptions, however, need to be still proven in our future work.

An all-in-one approach aiming for learning the decision without breaking the process down into policies with a single model based on a DNF failed at the early stages of our research, hence, we decided to model policies separately as also indicated in the overall architecture and resulting in the proposed gray-box model. A clear benefit of modelling each policy separately and integrating them in a second step rather than a learning-all-in-one approach is that properties of individual policies can be considered. For instance, in our task, the *Colour policy* is learnt by memorising the colour order and the *Distance policy* model is trained based on the recorded data optimising DNF parameters using a GA. Using the integration layer makes it possible to have several policy models trained with various approaches and integrated in the overall architecture. Furthermore, having the policy models developed separately makes it possible to easily add more policies to the architecture depending on future tasks. Some possible extensions are explained in the next section as future work. In addition, the chosen task and instructions make it possible to generalise the developed models for a complex task without requiring a complete retraining of the models. As mentioned before, a *Distance policy* is an integrated part of most pick-and-place tasks. The colours order also has been chosen in a way so that *Colour policy* could be generalised for other tasks with orders of actions, either with serial order like picking the red block first then the blue one, or having a parallel order for two actions with the same priority, like picking either of the blue blocks.

Despite close performance of the RNN models to DNF models for the training data in some conditions, for the test data, DNF outperformed both RNN and MLP. It can be due to the nature of DNF models and having a travelling wave for the movement of the blocks and participants, while MLP does not have state, and so cannot take advantage of the differences between the rate of decision-making and sampling. This should have been ameliorated by the use of LSTM nodes in the early layers of the recurrent model. However, insufficient computational budget for meta-parameter tuning and training of RNN models considering their greater complexity (number of weights to learn) than the DNF model consequently requires more training data to avoid overfitting.

It is noteworthy that the GA itself is governed by several parameters and operator choices, most notably whether it was allowed to re-evaluate duplicates (we did not restrict this). Moreover, the GA is evolving the weights for the DNF, whereas the random search is tuning the MLP hyperparameters, but the MLP weights are being tuned by a sophisticated meta-heuristic (Adam). Therefore we would argue that although we have attempted to achieve parity of computational effort within the use of standard toolkits (to replicate research), it is probably never possible to guarantee exact equality. As for LSTM, for computational reasons, we were not able to undergo the same systematic use of grid-search in the space of hyper-parameters. However, our preliminary studies revealed the LSTM performance was not overly affected by variations.

As the model was developed using a supervised learning approach, the outcome was compared to ANN as another supervised learning method. Nonetheless, an attempt was also made to apply POMDP on the datasets. Wang [63] found, although initial implementations using POMDP based on an artificially created dataset seemed promising when applied to the real dataset the POMDP-based model completely failed. This is likely due to the noise in the data. The recorded data consists of tracking coordinate frames of participants' motion and there are many short temporary losses of tracking. This, however, does not affect the DNF-based model as its dynamic nature damps a sudden change in the input stimuli. In addition, when lacking the probability of events and consequences, Reinforcement-Learning(RL)-based approaches typically require many iterations so would be less practical for human–machine collaboration.

In our experiments, the majority of people tended to minimise their energy consumption by picking the closest object. This has been also observed when participants were asked to perform the task without any instruction, making *Distance policy* the main naturally chosen policy in a pick and place task. In this case, using the system only with *Distance policy* models resulted in 87.78% prediction accuracy.

Inspired by mirror neurons and implemented by using DNFs, the negotiation layer may also facilitate safe human–robot collaboration. The collaboration will be safer as the Negotiation Layer reduces the chances of conflicts and, thereby, unwanted or unintentional contact since the human's action is directly affecting the robot's decisions. As Table 4 presents, our results showed that in dyadic scenarios, incorporating the negotiation layer improved the model performance, for example, the highest improvement was for the *Colour Policy* experiment by 26%.

Using DNFs along with a global optimisation approach like a GA to optimise its parameters has one disadvantage, i.e., it is computationally expensive. However, one could expect this to be mitigated in the future by emerging faster high-performance computers. For example, in this work during the training phase, a machine with 14 cores was used to run the optimisation in parallel making the training significantly faster than using a normal computer. Gradient descent approaches (available in Matlab) were also tested for optimising the parameters, however, they did not converge and ended with a relatively high error. This suggests that GA was a good choice for optimising DNF parameters as it can escape local minima. On the other hand, using a global search approach has made the training highly computationally expensive. Although this could be alleviated by advances in computing power and parallel processing, at present, it has limited the training phase. At the same time, considering the dynamic behaviour of the DNF, it is highly resilient to variation in its input and, in most previous work (cited in Sect. 2.1.5) for less complex systems, its parameters were chosen manually through expert tuning. Another advantage of using DNF for modelling decision-making is that it can capture the dynamic process of human decision-making. For example, in our data from the pick-and-place task any changes in the human motion indicating a change in the decision would dynamically change DNF activation on-the-fly and the predicted decision would be updated. When using a 2D DNF this process can also be visualised for a better understanding of the decision-making process and evaluation of the model performance.

6 Conclusion and Future Work

In conclusion, a novel decision-making architecture was presented. Our results indicate that modelling complicated policies can be achieved by integrating single policies and that conflicts can be resolved or prevented in a joint action by means of internal simulation in the proposed Negotiation Layer. The structure can be used for different tasks provided that the relevant policies are modelled and integrated into the system. Partner's actions are always considered in the decision-making process for joint actions, hence making this system a good candidate to be embedded in robots for human–robot collaboration (HRC).

The current structure of the decision-making module is designed to be extendable by introducing new policies as needed, which is a clear advantage of our proposed architecture. Currently, it is assumed that both agents have an equal affordance for all actions, however, inspired by the role of canonical neurons, in the future a policy model could be added for the agents' affordances to have an architecture for heterogeneous agents. Another example can be modelling user preferences resulting in a personalised robot that can adapt to the user needs. This makes the architecture suitable for many applications; from production lines to care-working or companion robots for older adults. It is also possible to combine this architecture with the one proposed by Sarthou et. al [22] for considering tasks in which agents have different perspectives of the environment. This could be done either by using their proposed modelling approach or a unified approach of using DNF models for the agent's perspective.

Further improvements may be achieved by training the models by means of Partially Observable Monte Carlo Planning (POMCP) similar to Goldhoorn et al. [64].

Data Availability The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Human and Animal Rights All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards.

Informed Consent Informed consent was obtained from all individual participants included in the study.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Trafton JG, Cassimatis NL, Bugajska MD, Brock DP, Mintz FE, Schultz AC (2005) Enabling effective human-robot interaction using perspective-taking in robots. IEEE Trans Syst Man Cybernet-Part A Syst Hum 35(4):460–470
2. Moratz R, Tenbrink T (2008) Affordance-based human-robot interaction. Towards affordance-based robot control. Springer, Berlin, pp 63–76
3. Lohse M (2011) The role of expectations and situations in human-robot interaction. New Frontiers in Human-Robot Interaction, 35–56
4. Yamazaki A, Yamazaki K, Kuno Y, Burdelski M, Kawashima M, Kuzuoka H (2008) Precision timing in human-robot interaction: coordination of head movement and utterance. In: Proceedings of the SIGCHI conference on human factors in computing systems, ACM pp 131–140
5. Chao C, Thomaz AL (2011) Timing in multimodal turn-taking interactions: control and analysis using timed petri nets. J Human-Robot Interact 1(1):1–16
6. Wang Y, Ruhe G (2007) The cognitive process of decision making

7. Langley P, Laird JE, Rogers S (2009) Cognitive architectures: research issues and challenges. Cogn Syst Res 10(2):141–160
8. Curioni A, Knoblich G, Sebanz N (2016) Joint action in humans: a model for human-robot interactions. In: Goswami A, Vadakkepat P (eds) Humanoid Robot A Ref. Springer, Switzerland, pp 1–19
9. Wolpert DM, Doya K, Kawato M (2003) A unifying computational framework for motor control and social interaction. Philosop Trans Royal Soc Lond B Biol Sci 358(1431):593–602
10. Gordon RM (1986) Folk psychology as simulation. Mind Lang 1(2):158–171
11. Shanton K, Goldman A (2010) Simulation theory. Wiley Interdiscip Rev Cogn Sci 1(4):527–538
12. Gallese V, Goldman A (1998) Mirror neurons and the simulation theory of mind-reading. Trends Cogn Sci 2(12):493–501
13. Bekkering H, De Bruijn ER, Cuijpers RH, Newman-Norlund R, Van Schie HT, Meulenbroek R (2009) Joint action: neurocognitive mechanisms supporting human interaction. Top Cogn Sci 1(2):340–352
14. Winfield A.F (2014) Robots with internal models: a route to self-aware and hence safer robots
15. Vanderelst D, Winfield A (2018) An architecture for ethical robots inspired by the simulation theory of cognition. Cogn Syst Res 48:56–66
16. Umilta MA, Kohler E, Gallese V, Fogassi L, Fadiga L, Keysers C, Rizzolatti G (2001) I know what you are doing: a neurophysiological study. Neuron 31(1):155–165
17. Iacoboni M, Molnar-Szakacs I, Gallese V, Buccino G, Mazziotta JC, Rizzolatti G (2005) Grasping the intentions of others with one's own mirror neuron system. PLoS Biol 3(3):79
18. Pacherie E, Dokic J (2006) From mirror neurons to joint actions. Cogn Syst Res 7(2–3):101–112
19. Sebanz N, Knoblich G (2009) Prediction in joint action: what, when, and where. Top Cogn Sci 1(2):353–367
20. Vinciarelli A, Esposito A, André E, Bonin F, Chetouani M, Cohn JF, Cristani M, Fuhrmann F, Gilmartin E, Hammal Z, Heylen D, Kaiser R, Koutsombogera M, Potamianos A, Renals S, Riccardi G, Salah AA (2015) Open challenges in modelling, analysis and synthesis of human behaviour in human-human and human-machine interactions. Cogn Comput 7(4):397–413. https://doi.org/10.1007/s12559-015-9326-z
21. Bicho E, Erlhagen W, Louro L, e Silva EC (2011) Neuro-cognitive mechanisms of decision making in joint action: a human-robot interaction study. Hum Mov Sci 30(5):846–868
22. Sarthou G, Mayima A, Buisan G, Belhassein K, Clodic A (2021) The director task: a psychology-inspired task to assess cognitive and interactive robot architectures. In: 2021 30th IEEE international conference on robot and human interactive communication (RO-MAN), pp 770–777. https://doi.org/10.1109/RO-MAN50785.2021.9515543
23. Metta G, Sandini G, Natale L, Craighero L, Fadiga L (2006) Understanding mirror neurons: a bio-robotic approach. Interact Stud 7(2):197–232
24. Beraldo G, Tonin L, Millán JDR, Menegatti E (2022) Shared intelligence for robot teleoperation via bmi. IEEE Trans Human-Mach Syst 52(3):400–409
25. Anderson JR, Bothell D, Byrne MD, Douglass S, Lebiere C, Qin Y (2004) An integrated theory of the mind. Psychol Rev 111(4):1036
26. Laird JE (2012) The soar cognitive architecture. MIT press, Cambridge, Massachusetts
27. Fan X, Sun S, Yen J (2005) On shared situation awareness for supporting human decision-making teams. In: AAAI Spring Symposium: AI Technologies for Homeland Security, pp 17–24
28. Zhang Z, Russwinkel N, Prezenski S (2018) Modeling individual strategies in dynamic decision-making with act-r: a task toward decision-making assistance in hci. Procedia Comput Sci 145:668–674
29. Kahneman D, Tversky A (1979) Prospect theory: an analysis of decision under risk. Econom: J Econom Soc, 263–291
30. Swere E, Mulvaney DJ (2003) Robot navigation using decision trees. Electronic systems and control division research
31. Huang H-P, Liang C-C (2002) Strategy-based decision making of a soccer robot system using a real-time self-organizing fuzzy decision tree. Fuzzy Sets Syst 127(1):49–64
32. Hausman D.M (1999) The handbook of economic methodology, In: John Davis, D Wade Hands, Uskali Mäki (eds.) Edward Elgar, 1998, xviii+ 572 pages. Economics and Philosophy 15(02), 289–295
33. Rosenblatt JK (2000) Optimal selection of uncertain actions by maximizing expected utility. Auton Robots 9(1):17–25
34. Wagner A, Briscoe E (2016) Psychological modelling of humans by assistive robots, 273–296
35. Bellman R (1957) A markovian decision process. Technical report, DTIC Document
36. Pineau J, Gordon GJ (2007) Pomdp planning for robust robot control, 69–82
37. Png SCOSW, Lee DHWS (2009) Pomdps for robotic tasks with mixed observability
38. Spaan MT, Spaan N (2004) A point-based pomdp algorithm for robot planning. In: Robotics and automation, 2004. Proceedings. ICRA'04. 2004 IEEE international conference On, IEEE vol. 3, pp 2399–2404
39. Foka A, Trahanias P (2007) Real-time hierarchical pomdps for autonomous robot navigation. Robot Auton Syst 55(7):561–571
40. Cassandra AR (1998) A survey of pomdp applications. In: Working Notes of AAAI 1998 fall symposium on planning with partially observable Markov decision processes, vol. 1724 . Citeseer
41. Busemeyer JR, Townsend JT (1993) Decision field theory: a dynamic-cognitive approach to decision making in an uncertain environment. Psychol Rev 100(3):432
42. Townsend JT, Busemeyer JR (1989) Approach-avoidance: return to dynamic decision behavior. In: Current Issues in Cognitive Processes: The Tulane Flowerree Symposia on Cognition, pp 107–133 . Psychology Press
43. Roe RM, Busemeyer JR, Townsend JT (2001) Multialternative decision field theory: a dynamic connectionst model of decision making. Psychol Rev 108(2):370–392
44. Smith PL (1995) Psychophysically principled models of visual simple reaction time. Psychol Rev 102(3):567
45. Nosofsky RM, Palmeri TJ (1997) An exemplar-based random walk model of speeded classification. Psychol Rev 104(2):266
46. Schöner G (2008) Dynamical systems approaches to cognition. In: Sun R (ed) The Cambridge handbook of computational psychology. Cambridge University Press, Cambridge, pp 101–126
47. Amari S-I (1977) Dynamics of pattern formation in lateral-inhibition type neural fields. Biol Cybern 27(2):77–87
48. Wilimzig C, Schneider S, Schöner G (2006) The time course of saccadic decision making: dynamic field theory. Neural Netw 19(8):1059–1074
49. Giese MA (2012) Dynamic neural field theory for motion perception, vol 469. Springer, New York
50. Jancke D, Erlhagen W, Dinse HR, Akhavan AC, Giese M, Steinhage A, Schöner G (1999) Parametric population representation of retinal location: neuronal interaction dynamics in cat primary visual cortex. J Neurosci 19(20):9016–9028
51. Erlhagen W (2003) Internal models for visual perception. Biol Cybern 88(5):409–417
52. Erlhagen W, Mukovskiy A, Bicho E (2006) A dynamic model for action understanding and goal-directed imitation. Brain Res 1083(1):174–188
53. Schöner G, Dose M, Engels C (1995) Dynamics of behavior: theory and applications for autonomous robot architectures. Robot Auton Syst 16(2):213–245

54. Erlhagen W, Mukovskiy A, Bicho E, Panin G, Kiss C, Knoll A, Van Schie H, Bekkering H (2006) Goal-directed imitation for robots: a bio-inspired approach to action understanding and skill learning. Robot Auton Syst 54(5):353–360
55. Faubel C, Schöner G (2008) Learning to recognize objects on the fly: a neurally based dynamic field approach. Neural Netw 21(4):562–576
56. Bicho E, Louro L, Erlhagen W (2010) Integrating verbal and nonverbal communication in a dynamic neural field architecture for human-robot interaction. Front Neurorobot 4:5
57. Erlhagen W, Bicho E (2014) A dynamic neural field approach to natural and efficient human-robot collaboration, 341–365
58. Sandamirskaya Y (2014) Dynamic neural fields as a step toward cognitive neuromorphic architectures. Front Neurosci 7:276
59. Sandamirskaya Y, Schöner G (2010) An embodied account of serial order: how instabilities drive sequence generation. Neural Netw 23(10):1164–1179
60. Cosiniva: COSINIVA: Dynamic Field Theory MATLAB Toolbox. https://dynamicfieldtheory.org/cosivina/ (2019-02-05)
61. Igel C, Erlhagen W, Jancke D (2001) Optimization of dynamic neural fields. Neurocomputing 36(1):225–233
62. Mitchell M (1998) An introduction to genetic algorithms. MIT press, Cambridge, Massachusetts
63. Wang Z (2020) Modelling decision-making in a joint action for picking an object. Master's thesis, University of Bristol and University of West of England
64. Goldhoorn A, Garrell A, Alquézar R, Sanfeliu A (2018) Searching and tracking people with cooperative mobile robots. Auton Robots 42(4):739–759

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Mehdi Sobhani has been a member of Bristol Robotics Laboratory since 2011 and worked on many EC-FP7 and EPSRC funded projects on Medical Robotics, safe interactive robots, Human-Robot Interaction, Robot Adaptive Behaviour, and Decision Making in joint-action. He had also close collaboration with Experimental Psychologist and is experienced with Machine Learning and data analysis. He is currently a Research Fellow in the National Centre for Nuclear Robotics where his research is focused on the user-centred design of novel teleoperation interfaces using Virtual Reality.

Jim Smith received the first degree in electrical sciences from Cambridge University, Cambridge, U.K., and the Ph.D. degree from the University of the West of England, Bristol, U.K., in 1998. He spent several years in industry before earning his Ph.D. He is currently a Professor of interactive artificial intelligence with the University of the West of England, Bristol, U.K. His current research interests include the theory and application of intelligent systems that adapt their learning strategies in response to experience, and the interface between humans and adaptive intelligent systems.

Tony Pipe received his BSc (Hons.) degree in Electronic Engineering from Warwick in 1979 and, after an eight-year period in industry, returned to academia and received his PhD in Robotics from the University of the West of England (UWE) in 1997. Since 2010 he has been a Professor of Robotics and Autonomous Systems at UWE. He was a founding member of the Bristol Robotics Laboratory (BRL), and was one of its Deputy Directors from its inception in 2006 until 2020. His research interests focus on safe and correct operation, normally in close-proximity to human beings. He has co-authored over 200 international refereed publications, supervised 35 PhDs, and created a BRL income of 12M since 2013.

Angelika Peer is Full Professor at the Free University of Bozen-Bolzano (Italy) since November 2017. From 2014 to 2017 she was Full Professor at the Bristol Robotics Laboratory, University of the West of England, Bristol, UK. Before she was senior researcher and lecturer at the Institute of Automatic Control Engineering and TUM-IAS Junior Fellow of the Institute of Advanced Studies of the Technical University of Munich, Germany. She received the Diploma Engineering degree in Electrical Engineering and Information Technology in 2004 and the Doctor of Engineering degree in 2008 from the Technical University of Munich. Her research interests include robotics, control and human system interaction.