TEACHING ROBOTS SOCIAL AUTONOMY FROM IN-SITU HUMAN GUIDANCE

by

EMMANUEL SENFT

A thesis submitted to Plymouth University
in partial fulfilment for the degree of

DOCTOR OF PHILOSOPHY

School of Computing, Electronics and Mathematics
Faculty of Science and Engineering

?? July 2018

Abstract

TEACHING ROBOTS SOCIAL AUTONOMY FROM IN-SITU HUMAN GUIDANCE
Emmanuel Senft

Here is the abstract

my original contribution to knowledge is a new teaching paradigm for robotics: Supervided Progressive Autonomous Robot Competenties (SPARC) which has been validated in three studies: interaction with a model, interaction with a fixed environment and interaction with humans in the context of robotic tutors.

First demonstration of teaching a robot to interact with humans

Contents

List of Figures

List of Tables

Acknowledgements

Thanks Séverin, Paul and of course Tony

Author's declaration

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award. Work submitted for this research degree at Plymouth University has not formed part of any other degree either at Plymouth University or at another establishment.

This work has been carried out by Emmmanuel Senft under the supervision of Prof. Dr. Tony Belpaeme, Dr. Paul Baxter, and Dr. Séverin Lemaignan. The work was funded by European Union FP7 projects DREAM (grant no.: 611391).

Parts of this thesis have been published by the author:

Word count for the main body of this thesis:

Signed: ______________________

Date: ______________________

Chapter 1

Introduction

Human-Robot Interaction (HRI) what it is and what it matters and why it is challenging

Robots will inhabit human spaces and need to interact with them in decent ways

1.1 Scope

1.1.1 Frame

1.1.2 Environment

Robot interacting in an environment shared with humans / directly with humans presence of a supervisor who can provide feedback/commands

1.1.3 Type of interaction

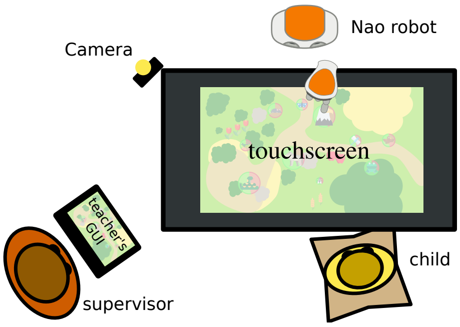


Figure 1.1: The setup used in the study: a child interacts with the robot tutor, with a large touchscreen sitting between them displaying the learning activity; a human teacher provides guidance to the robot through a tablet and monitors the robot's learning.

1.1.4 Algorithms

Three different algorithms have been used in the progress of this research. The first study presented in chapter 4 uses feed forward neural network with a single hidden layer. the second study in chapter 5 used Reinforcement Learning combining human and environmental rewards into a single reward source for the algorithm. And the last study presented in chapter 6 uses instance based algorithm adapted from Nearest Neighbours to enable quick and efficient learning. More details about each algorithms and their related work can be found in the associated chapters.

1.2 The Thesis

The main thesis that this document seeks to put forward is as below.

A robot can learn how to interact meaningfully with humans by receiving teaching content from a human supervisor which leads to an efficient, safe and low human-workload interaction policy.

Additional research questions have been explored during the progress of this work and are introduced here.

- **Does adding a learning component to a supervised robot can reduce the human-workload of the supervisor?**

  Wizard-of-Oz (WoZ) is an approach widely used in HRI (Riek, 2012), whereby a human teleoperates a robot to have it interact with other humans. However, this method applies a high workload on the operator and is not scalable. Using Machine Learning (ML) to learn from this operator online might decrease the operator's workload without decreasing the quality of the robot behaviour.

- **How control of a teacher over the robot's action impacts the robot's learning?**

  In the context of Interactive Machine Learning (IML), a human can provide inputs to an agent to speed up the learning. Other IML (Thomaz & Breazeal, 2008; Knox & Stone, 2009) focus on feedback from the human with limited or no control over the agent's actions. However increase the control should speed up the learning and reduce the number of errors made by the robot.

- **How could a human teach a robot how to interact with other humans?**

  SPARC has been designed to allow non-experts in ML to teach agents how to interact while interacting. Human-robot interactions provide a perfect test for this approach: using a human to teach a robot how to behave in this complex and non-deterministic environment.

1.3 Approach and Experimentation

1.4 Key Concepts

1.5 Challenges

1.6 Contributions

- Something

1.7 Structure

The structure of this thesis is outlined below to provide an overview of the content and context for each chapter. A summary of key experimental findings are included at the start of each relevant chapter for ease of reference.

- This chapter provided an introduction to the general field of this research (robot tutors for children), the research questions including the central *thesis*, scope, and contributions of the work presented in later chapters.
- Chapter 2
- Chapter 8 concludes the thesis with a summary of the main contributions.

Chapter 2

Background

This chapter describes the application fields of HRI and related research in agent and robot control. The first section presents different fields of application of social HRI and then draws from them requirements for controlling a robot interacting with humans. The second section provides the current state of the art in robot control for HRI and analyses it through the constraints presented in Section 2.1.2. And the third section presents an alternative method to teach agents how to interact and how it could be applied to HRI.

2.1 Social Human-Robot Interaction

2.1.1 Fields of application

Human-Robot Interaction covers the full spectrum of interactions between humans and robots. However, as this thesis is focused on teaching robots to interact socially with humans, we used the following criteria on the interactions to select the subfields of HRI relevant to our research:

- presence of an interaction between a robot and a human: both the human and the robot are influencing each other's behaviour
- the social side of the interaction is key, the interaction involves a socially interactive robot as defined in Fong et al. (2003) (no physical human-robot interaction, such as an exoskeleton, physical rehabilitation or pure teleoperation as in robotic assisted surgery).

Socially Assistive Robotics

Socially Assistive Robotics (SAR) is a term coined in Feil-Seifer & Matarić (2005) and refers to a robot providing assistance to human user through social interaction and has been defined by Tapus et al. (2007) as one of the grand challenges of robotics.

One of the principal application of SAR is care for the elderlies. Ageing of population is a known challenge for today and the near future: the United Nations. Department of Economic and Social Affairs (2017) reports than 'population aged 60 or over is growing faster than all younger age groups'. This will decrease the support ratio (number of worker per retiree) forcing society to find ways to provide care to an increasing number of persons using a decreasing workforce. Robots represent a unique opportunity to provide for this lacking workforce potentially allowing elderlies to stay at home rather than joining elderly care centres (Di Nuovo et al., 2014) or simply to support the nursing home staff (Wada et al., 2004).

However, the acceptance of these robots in elderly care facilities or at their homes, is still a complex task. Multiple studies report a good acceptance of robot and positive effects on stress in home cares both for the elderly and the nursing staff (Wada et al., 2004), but as mentioned in Broadbent et al. (2009), this acceptance could be increased by matching more closely the behaviour of the robots to the actual needs of the patients.

The second main application of SAR is Robot Assisted Therapy (RAT). In addition of providing social support, robots can also be used in the process of therapies, following a patient during their rehabilitation to improve their health, their acceptance in society or recover from an accident. In therapies, robots have first been used as physical platform to help patient in rehabilitation therapy in the eighties (Harwin et al., 1988). They were primally used as mechatronic tools helping humans to accomplish repetitive task. But in the late 90s, robots have started to be used for their social capabilities. For example the AuRoRA Project (Dautenhahn, 1999) started in 1998 to explore the use of robot as therapeutic tools for children with Autism Spetrum Disorder (ASD). Since AuRoRA, many projects have started all around the world to use robot to help patient with ASD, as shown by the review presented in (Diehl et al., 2012) or more recently the Development of Robot-Enhanced Therapy for Children with Autism Spectrum Disorders (European FP7 project) (DREAM) (Esteban et al., 2017). RAT is not limited to ASD only, the use of robots is also explored in hospitals, for example to support children with diabetes (Belpaeme et al., 2012), to support elderly with dementia (Wada et al., 2005) or to provide encouragement and monitor cardiac rehabilitation (Lara et al., 2017).

Education

Social robots are also being used in education, supporting teachers to provide learning content to children. As presented in Mubin et al. (2013), the robot can take multiple roles such as peer, tutor or tool, however this *tool role* has been excluded for this overview due to its lack of social interaction.

Robotic teachers providing class lessons. One of the most obvious role robots could have in a classroom is replacing the teacher to provide the lesson content to children as presented in Verner et al. (2016). However this approach is seldom used in robots for education. The aim is not to replace teachers but to support them with new tools to improve the teaching process both for the teachers and the children, the review by Mubin et al. (2013) do not even mention robot teachers as a possible role for robots in educations.

Robotics tutors providing tailored teaching content in 1 to 1 interaction. Studies reported that individualised feedbacks can increase the performance of students (Bloom, 1984), but due to the large number of students supervised by a single teacher in classes today, tutoring is complex to apply. Robotic tutors could provide this powerful one to one tailored interaction not available in current classroom without increasing the workload on teachers. Studies have also shown that adding robot in a classroom can improve the children learning outcomes (Kanda et al., 2004). Additionally, robots can be used at home to elicit advantages over web or paper based instructions (Han et al., 2005). However as pointed by Kennedy et al. (2015), social behaviours have to be carefully managed as a robot too social could decrease the learning for the children compared to a less social robot.

Peer robots learning alongside the children. A peer robot will not mentor the child to learn certain concepts, but will learn with or from the child. Unlike other roles of robots in education, peer robots can fit new roles vacant in education today. For example, in Co-Writer (Hood et al., 2015), the child has to teach a robot how to write, and as the child demonstrates correct handwriting, they improves their own. Peer robots can leverage the concept of learning by teaching (Frager & Stern, 1970) in a way hardly matchable by humans. The robot can take the role of a less knowledgeable agent with endless patience and encourage the student to perform repetitive tasks such as handwriting and improve.

Search and rescue, and military

Robots are already used during search and rescue missions. After a natural or artificial catastrophe, robots can be sent to analyse the damage area and report or rescue the surviving victims of the incident (Murphy et al., 2008). These robots have to interact socially with two kinds of human partners: the survivors and the rescue team. In both cases the social component of the interaction is key: the survivor can be in a shocked state and the robot could be the only link they received with the external world after the accident. In this case, a social response is expected from the robot and it has to be carefully controlled. On the other side, the rescue team monitoring the robots is under pressure to act quickly and faces traumatic events too. Even if the robot does not display social behaviour, rescuers interacting with it might develop some feeling toward the robots they are using during these tense moments and this has to be taken into account when designing the robot and its behaviour (Fincannon et al., 2004).

Similar human behaviours (emotional bonding with a robot) have also been observed in the army. Robots have been deployed as teleoperated drones and ground units alongside soldier to do scouting or demining tasks. And soldiers developed feelings toward the robot they use in a daily basis: taking pictures with it, introducing it to their friends and so on. This relation can go as far as soldier risking their life to in order to save the robot used by their squad (Singer, 2009). In that case, the social side of the robot has to be carefully managed to prevent it to have an opposite effect that the one desired: preventing the waste of human lives.

Hospitality and Entertainment

Robots are also interacting with humans in hotels around the world. The Relay robot (Savioke[1]) delivers commands for the guests of hotels from the reception to their room. Whilst the social interaction is still minimal today, these robots interact everyday with humans and provoke social reactions from them. On the research side, scientists have

[1] http://www.savioke.com/

explored robots as receptionist in a hall at Carnegie Melon University (Gockley et al., 2005) and robots as museums and exhibitions guide since the late 90s (Thrun et al., 1999; Burgard et al., 1999) and since, explore how to improve the social interaction for guide robots.

Similarly, robots guide and advice humans in shops and shopping mall. Long term studies explored how humans perceive and interact with robots in this environment (Kanda et al., 2009) and how robots should behave with clients (Kanda et al., 2008).

Robots have also entered homes and family circles. Twenty years ago, Sony created Aibo, a robotic dog to be used as pet in Japanese families. An analysis of online discussion of owners published 6 year after their introduction gives insights on the relationship that owners created with their robots (Friedman et al., 2003). For example 42% of the members assigned intentionality to the robot, like preferences, emotions or even feelings. Similar behaviours can also be observed when the robot is not presented as a pet, but just a tool. In Fink et al. (2013), authors reported that a participant was worried that their Roomba would feel lonely when they would be in holidays. More recently, the Pepper robot has been sold to family in Japan, however, as of March 2018 no english study has reported results of the interaction with family members or long term use.

Collaborative Robots in industry

Robots in industry used to be locked behind cages to prevent humans to interact with them and getting hurt. However recently, social robots such as Baxter (Guizzo & Ackerman, 2012) have been designed to collaborate with humans and share the same workspace interacting physically and socially with factory workers. As these robots need to be safe to human interacting with them, they are often softly actuated, using active or passive compliance (e.g. GummiArm [2]). Additionally, the legibility of motion is important, to interact efficiently and safely with humans, robots need to make their intentions clear to humans surrounding them Dragan et al. (2013) and reciprocally, they need to interpret human social cue to infer their goals and intentions.

Beyond the safety and legibility, another key challenge in collaborative robotics is task assignment: if a goal has to be achieved by a human-robot team, the repartition of tasks should be carefully managed to optimise the end result in term of task performance, but

[2] https://mstoelen.github.io/GummiArm/

also to ensure comfort for the human. Multiple sets of implicit rules have to be taken into account in a human-robot collaboration context, and so the task repartition system should be aware of them and follow them as proposed in Montreuil et al. (2007). Recent work done in the 3$^{\text{rd}}$ Hand project [3] exploring for example how a robot can learn to support human in a collaborative assembly task Munzer et al. (2017).

2.1.2 Constraints

The previous section demonstrated that robots are already interacting with sensitive populations: young children, elderlies, persons with handicap or in a stressful situations (victims of catastrophes, soldier or persons requiring healthcare for example). And, as pointed previously in the case of robots for the elderlies, this number of human-robot interactions is likely to rise for the other categories too. In this context, failure to meet expectations or lack of social norms or awareness might lead to physical injuries to surrounding humans or potentially to offence, anger, frustration or boredom. As such, the behaviour of robots interacting with humans need to be carefully managed to meet expectations and robots needs to behave in socially acceptable manner avoiding any inappropriate or confusing behaviours.

These undesired behaviours can have many different origins: lack of sensory capabilities to identify necessary environmental features, lack of knowledge to interpret human behaviours appropriately, failure to convey intentions, impossibility to execute the required action or incorrect action policy. Due to the wide range of origins of these potential social faux-pas, this research focuses only on the last point, obtaining a correct action policy: assuming a set of inputs, how a robot should select the most correct action. The other issues are either orthogonal and would lead to failure even with an 'optimal' action policy as external factors prevent the robot to solve a problem or can be handled by having a better action policy, generalising better, selecting suboptimal actions when the optimal ones cannot be used, or reacting correctly to human behaviours without having to interpret them on a higher level.

Appropriate actions are highly dependent on the interaction context: they could aim to match users' expectations of a robot behaviour or complete a specific task and actions correct in a context might be inappropriate in another one. Nonetheless, the intention

[3] http://3rdhandrobot.eu/

behind being *appropriate* here is that actions executed should be guaranteed not to present risk for the humans involved in the interaction; for example, preventing physical harm or mental distress while achieving the robot assigned objectives and help the robot to move toward its goal.

Additionally, interactions with humans 'in the wild' (Belpaeme et al., 2012) do not happen in well defined environments or rigid laboratory setups: in the real world, robots have to interact in diverse environments, with a large number of different persons, during extended periods of time or with initial incomplete or incorrect knowledge: as such the action policy also needs to be adaptable to the context, to the users and over time. Robots need to be adaptive, both to react to changing environments, and to improve their action policy over time.

Lastly, in many cases today, interactive robots are not autonomous but partially controlled by a human operator. We argue that to have a real Human-Robot Interaction, the robot needs to be as autonomous as possible. As pointed out in Baxter et al. (2016), by relying too much on a human to control a robot, we are shifting from a human-robot to a human-human interaction using a robot as proxy. As a community, HRI should strive toward more autonomy for robots interacting with humans.

We define three axes to analyse a robot behaviour and evaluate how suited its behaviour is in the interaction:

- Appropriateness of actions
- Adaptivity
- Autonomy

And we will use these axes to analyse current robots' controller types in Section 2.2.

HRI being a large field, other research axes are equally important such as the complexity or the depth of the interaction, the constraints put on the environment, the ability of the robot to set its own goals or the dependence and knowledge of social rules or the range of application to cite only a few. However, these axes are more influenced by the goal and the context of the specific human-robot interaction taking place rather than the action policy itself, so this research focuses on the three axis mentioned previously.

To be able to sustain meaningful social interactions, robots should score highly in the three axes presented before, following these three principles:

1. Only execute appropriate actions.
2. Have a high level of adaptivity.
3. Have a high level of autonomy.

Appropriateness of Actions

As argued previously, much of social human-robot interaction takes place in stressful or at least sensitive environments where humans have particular expectations about the robot behaviour or needs from the robot. Additionally, even in less critical situations, human-human interactions are subject to a large set of social norms and conventions resulting from precise expectations of the interacting partners (Sherif, 1936). Some of these expectations are also transferred to interactions with robots (Bartneck & Forlizzi, 2004).

Failing to produce appropriate actions, for example by not matching the users' expectations, can have a negative impact on the interaction, potentially compromising future interactions if the human feel disrespected, confused or annoyed. Similarly failing to behave appropriately can harm the persons interacting with the robots: not reminding an elderly to take their medication, not taking into account the state of mind of survivors after a disaster or behaving inconsistently with children with ASD can lead to dramatic consequences. We argue that robots require a way to ensure that all the actions they are executing do not present risks to the humans involved in the interaction and move the robot closer to its goal.

Being able to behave appropriately is a tremendous challenge: real interactions involve a large sensory space, with the human interactants often being unpredictable or at least highly stochastic. In addition, social interaction is grounded by a large number of often implicit norms, with expectations being highly dependent on the context of interaction. It seems unlikely that an action policy covering every possibility can be provided to the robot before the start of the interaction. For this reason, social robots need an action policy able to generate the appropriate reactions for the anticipated states, but also able to manage uncertainty, selecting a correct action even when facing a sensory state with not expected by the designers.

For the review in the next section, the appropriateness of actions axis is a continuous spectrum characterising how much the system controlling the robot is in control of the interaction and can act in a safe and useful way for the users at any moment of the interaction. For example, a robot selecting its action randomly would have a low appropriateness as no mechanism prevent the execution of unexpected or undesired action. On the other hand, a robot continuously selecting the action that a human expert would select would have a high value as domain experts have the knowledge of which action is the correct one in this interaction domain.

Adaptivity

For reasons similar to the ones stated in Section 2.1.2 and as pointed by other research groups (Argall et al., 2009; Hoffman, 2016), an optimal behaviour of a robot is unlikely to be programmable by hand. End users can express behaviours not anticipated by the designers, the environment is often not perfectly definable and the desired behaviour might need to be customisable by or for the end user or evolve over time. For these reasons, robots interacting with humans need to be able to update their action policy and improve their behaviour over time. We use the term *adaptivity* to represent this ability to express a behaviour suited to different conditions and refine it over time.

While many studies in HRI do not use adaptive robots, to interact meaningfully outside of lab settings or scientific studies, robots need to possess this adaptivity to extend their range of application and improve their interactions with users.

We propose two components for this adaptivity: the basic component is the adaptivity to the environment, i.e. the generalisation of the behaviour (reacting accordingly to different types of inputs) and the second component is the adaptivity in time which can be assimilated to learning (the possibility to enrich and refine the action policy at runtime).

The same robot might be expected to interact in different environments or the environment where it evolves might also change with time. For example a robot used as an assistant for elderly people will have to interact in the home of the owner, but can also have to follow the owner in the street or in a supermarket. In these different environments, different behaviour will be expected from the robot, as such to be able to behave accordingly, the robot has either to be programmed to be adaptive to the environment or be able to learn how to interact in this environment.

Additionally, in most of the application fields presented earlier, robots have to interact with a large number of users, and often, these interaction partners are not know in advance and can have different roles: in education the name and particularities of each child can hardly be specified in advance and the robot might also have to interact with the students and with the teacher and in entertainment or search and rescue, none of the user is known beforehand. Adaptivity can be a way to identify the diverse users and adopt an action policy that suit the current user more specifically.

Adapting to its users is especially important as in these fields where the interaction is social, robots deployed in the wild might interact over extensive periods with the same user, e.g. companion robots for the elderly, military robots for a squad or robots used in RAT. With these long-term interactions, learning allows the robot to tailor its behaviour to the current user and track the changes of preferences that could occur over long periods of interaction. This adaptivity in time can additionally the robot to learn from its errors and improve its action policy over time.

Furthermore, providing a robot with learning enables it to be used by non-experts in robotics, granting them a way to design their own human-robot interactions, and making use of their expertise and knowledge to have the robot interacting as they desire. This is crucial as many application of social robotics, such as RAT, happen in environment where non-technical person possess the domain expertise required to ensure that the robot is efficient, and as stated by Amershi et al. (2014), this reduces the requirements on expensive and time consuming rounds-trips between domain-experts and engineers and decreases the risks of confusion between these two communities.

For this review, adaptivity is a continuous scale ranging from no adaptivity at all (the robot has a linear script that it follows in all the interactions), to high adaptivity (the robot dynamically change its action policy during an interaction and learns new actions and task and how to improve its actions policy).

Autonomy

To be deployable in the world and interact with humans, robots have to be autonomous. A limited human supervision can support the robot and improve its behaviour, but the robot should not rely on humans for its action selection during the main part of the interaction. Today, many experiments are conducted using a robot tele-operated by a human (Baxter et al., 2016). Whilst having a human controlling the robot presents many advantages, e.g.

the human can provide the knowledge and the adaptivity required and has sensing and reasoning capabilities not yet implemented on the robot, multiple reasons push us away from this type of interaction (Thill et al., 2012). It is not suited for deploying robots in the real world: it does not scale to interact for a long period of time or on larger scale, the human-robot interaction tends to become a human-human interaction (Baxter et al., 2016) and it might introduce multiple biases in the robot behaviour (Howley et al., 2014). For these reasons among many, we argue that a robot used in social HRI should be as autonomous as possible.

The third axis of this literature review is the autonomy. As stated by Beer et al. (2014), autonomy is organised following a spectrum of different levels of autonomy from no autonomy at all: a human is totally controlling the robot (doing sensory perception, analysis and action selection) to a full autonomy: the robot senses and acts on its environment without relying on human inputs. Levels exist between these extremes where a human and a robot share perception, decision or action: for example the robot can request information from a supervisor or the supervisor can override the action or goal being executed

2.2 Current robot behaviours in HRI

This section presents diverse approaches currently used in HRI to control a robot. As the number of individual techniques is too large for an exhaustive review, we organised the literature into broader categories. For each category, we will present the corresponding approach, indicate representative works done in this direction and qualitatively rate it on the three axes defined in the previous section.

2.2.1 Fixed preprogrammed behaviour

One of the simplest ways to have a robot interacting with a human is to have an explicit fixed behaviour. The robot is fully autonomous and follows a script or a finite state machine for action selection. This approach is dependent on having a well defined and predictable environment to have the interaction running smoothly. If the interaction modalities (possible range of behaviour and goal) are limited enough, a desired robot behaviour can be predefined for all (sensible) human actions. This approach is followed in a large number of research in HRI. Many studies being human-centred, the focus is not in the complexity of the robot's behaviour but on how different humans would interact with and react to a robot displaying a fixed behaviour (for comparison purposes). While being useful to explore human's reactions to robots, this method can hardly be used to deploy robots to interact with humans on a daily basis due to the lack of well defined environments in real world applications.

By essence, this type of controller has a no adaptivity as the robot is following a preprogrammed script, but scores highly on autonomy as no external human is required to control the robot and when the application domain is highly specified, the behaviour can be mostly appropriate.

2.2.2 Adaptive preprogrammed behaviour

To go beyond a script, the robot can also react to the human behaviour. The robot could be programmed with different behaviours, and then select the one corresponding to the current interaction following rules given prior the experiment. For example, in Leyzberg et al. (2014) the robot can deliver some predefined content according to the current performance of the participant, presenting personalised behaviour as long as the participants is behaving within expectations.

Reactive controllers mapping directly input data to behaviour without trying to reach specific high level goals have been also used successfully in HRI. For example, homeostasis, the tendency to keep multiple elements at equilibrium, is constantly used by living systems to survive and have been used to control robot in social interaction. Breazeal (1998) uses a set of drives (social, stimulation, security and fatigue) which are represented by a variable each and have to be kept within a predefined range. If these values are outside the desired homeostatic range, the robot is either over or under-stimulated and this will affect its emotion status and it will display an emotion accordingly. Homeostasis approaches have also been extended to robotic pets (Arkin et al., 2003) or RAT (Cao et al., 2017).

Due to the implicit description of behaviours, homeostasis-based methods are more robust in unconstrained environments than a purely scripted controller, while remaining totally autonomous. However the action policy is not adaptive in time and as the behaviours are not totally defined and controlled, there is no guarantee against the robot acting in inconsistent way in some specific cases limiting the appropriateness of actions.

Both predefined adaptation and homeostasis-based methods score highly in autonomy and can have a moderate to high level of appropriateness, but the adaptation is low as they can only adapt to the environment within predefined and anticipated boundaries and the robot does not learn.

2.2.3 Wizard of Oz

Wizard-of-Oz (WoZ) is a specific case of tele-operation where the robot is not autonomous but at least partially controlled by an external human operator to create the illusion of autonomy in an interaction with a user. It outsources the difficulty of action selection and/or sensory interpretation to a human operator. This technique has emerged from the Human-Computer Interaction (HCI) field in 1983 (Kelley, 1983) and is today common practice in HRI (Riek, 2012). Similarly to scripted behaviours, WoZ is highly used in human-centred studies to explore how humans react to robot and not as a realistic way to control robots in the environment. WoZ can assume different levels corresponding to the involvement of the human in the action selection process. Baxter et al. (2016) differentiate two main levels: perceptual WoZ (the human only replace sensory system and feed information to the robot controller) and cognitive WoZ (the human makes decisions about what the robot should do next). This method can also be used to gather data to develop a robot controller from human demonstration (cf. Section 2.2.4).

Likewise the different levels of autonomy presented earlier, systems can combine human control and predefined autonomous behaviour. Shiomi et al. (2008) proposes a semi-autonomous communication robot. This robot is mainly autonomous, but has the ability to make explicit request to a human supervisor in predefined cases where the sensory input is not clear enough to make a decision. The human can also provide additional information on the state of the world to inform the robot of events or do natural language recognition, to inform an autonomous system about the sentences said by the users.

With WoZ, the adaptivity and the appropriateness are provided almost exclusively by the human, so these characteristics are dependent of the human expertise but are generally high. However, due to the reliance on human supervision to control the robot, the autonomy is low. For semi-autonomous robots, the picture is more complex: as explained by Beer et al. (2014), the initiative, the human's role and the quantity of information and control shared will influence the level of autonomy. For example, in Shiomi et al. (2008) the robot explicitly makes requests to the human, but the human cannot take the initiative to step in the interaction limiting the adaptivity (especially as the robot policy is fixed) and as no mechanism preventing the robot to make undesired decision, the appropriateness of actions is lower compared to classical WoZ.

2.2.4 Learning from Demonstration

As stated by numerous researchers, explicitly defining a robot behaviour and manually implementing it on a robot can take a prohibitive amount of time or not be possible at all (Argall et al., 2009; Billard et al., 2008). This statement applies both to manipulation tasks and social interaction. However in both cases, humans have some knowledge or expertise that should be transferred to the robot. However in many case for social robots, experts of the field do not have the technical knowledge to implement this behaviour on a robot requiring numerous iteration of design between the users and engineers to reach a consensus.

The field of Learning from Demonstration (LfD) aims to tackle these two challenges: implementing behaviours too complex to be specified in term of code and empowering end-users with limited technical knowledge to transfer an action policy to a robot. Humans demonstrate a correct behaviour through different control means (Argall et al., 2009), and then offline batch learning is applied to obtain an action policy for the robot and if required, reinforcement learning can complement the demonstrations to reach a successful action policy (Billard et al., 2008).

In Learning from Demonstration (LfD) the interaction between the robot and the human teacher is key, however in most of the cases the object of the learning is not social interaction with humans, but manipulation or locomotion tasks. For example, Abbeel & Ng (2004) used telecontroled helicopter experts to demonstrate flying acrobatics, and extracted from these demonstrations a reward function used to train a controller to reproduce these acrobatics at a super-human level through Inverse Reinforcement Learning.

However, two approaches have explored using LfD to teach robots a social policy to interact with humans.

The first one is using human-human interactions to collect data about human behaviours and transfer them to a robot. Liu et al. (2014) present a data driven approach taking demonstration from human-human interactions to gather relevant features defining human social behaviour. Authors recorded motion and speech from about 180 interactions in a simulated shopping scenario, then behaviours were clustered into *behaviour elements*. Finally, during the interaction, the robot used a variable-length Markov model predictor to estimate the selection probability of each actions by the human demonstrator, and then selects the one with the highest probability. According to the authors, the final performance of the robot was not perfect, but if this approach was scaled using a larger dataset gathered from normal human-human interactions in the real world, the performance should improve and become closer to natural human behaviours.

Alternatively, the data can be collected from a Wizard-of-Oz setup. Knox et al. (2014) coined this approach *Learning from Wizard*: starting for a purely WoZ control to gather data, and then apply machine learning to derive an action policy. But this paper presents no description of which algorithm could be use or how, and gives no evaluation of the approach, but instead only offers a reflection on the application of this idea. An implementation and evaluation is briefly discussed by the authors in Knox et al. (2016), but the lack of detail reduces the usability of the paper.

This Learning from Wizard is widely used in HCI and especially dialogue management (Rieser & Lemon, 2008), and has been implemented by other groups of researchers in robotics. Sequeira et al. (2016) extended the idea to a full methodology to obtain a fully autonomous robot tutor. This method is composed of multiple steps starting with the observation of a human teacher performing the task. Then, the different features used by the teacher to select their actions as well as the actions themselves are encoded and implemented in a robot. The next step is setting up a WoZ experiment where the operator has access to the same features than the robot to make his decisions and controls the robot's action. Then, a combination manually derived rules and machine learning is applied on the data from the restricted-perception experiment and finally the robot is tested autonomously. Additional offline refinement steps are possible if the behaviour is not exactly the one desired.

Both Knox et al. (2014) and Sequeira et al. (2016) stress the importance of using similar features for the Wizard of Oz part than the ones available to the robot during the autonomous part: whilst decreasing the performance in the first interaction, it allows more accurate learning due to the similarity of inputs for the robot and the human controlling it.

Clark-Turner & Begum (2018) decided to bypass these limitations by using a deep Q-network (Mnih et al., 2015) to learn an Applied Behaviour Analysis policy for RAT.

They recorded videos, microphone inputs and actions selected in a WoZ interaction with participants. Then, the network learns from the raw inputs and the actions selected an action policy to control the robot and deliver the therapy. However in their study, the performance of the system was low (less than 70%) which means that the robot would provided inconsistent feedback at some point and they required limited human inputs to reach that level.

As these methods are based on real interactions either between humans, or between humans and robots controlled by humans, with enough demonstrations the robot should be able to select the appropriate actions. However, the efficiency is limited by the type of inputs recorded and the capabilities of the learning algorithm with the inputs space, which might limit the appropriateness of the action policy. After the learning phase, the robot behaviour is mostly static, without any additional learning provided. As such, the adaptivity is also reduced once the robot is deployed. Sequeira et al. propose that additional offline learning steps could be added, but online learning would allow a smoother transitions and improvement of behaviours. And finally, all these methods require the presence of humans in a first phase but the robots are fully autonomous later in the interaction, so the autonomy is null in the first phase and then high during the main part of the interaction.

2.2.5 Planning

An alternative way to interact in complex environments is to use planning. The robot has access to a set of actions with preconditions and postconditions and a defined goal. To achieve this goal state, it follows the three planning steps: sense, plan and act. The first step, sense, is to acquire information about the current state of the environment. Then, based on the set of actions available and the goal, a plan is created. This plan is a trajectory in the world, a succession of action and states which, according to the defined pre and postconditions, will lead to the goal. Finally, the last step is to execute the plan. The plan can be reevaluated at each state or only if a state differs from the expected one, in that case the robot update its plan to the new conditions and continue trying.

The efficiency of planning relies on having a precise and accurate set of pre and postconditions for each actions. And as humans are complex, if not impossible, to model precisely, planning have seen limited use for open social interactions with humans. However, due to the nature of planning, reaching a specific goal, it has been applied successfully to human-robot collaborative task achievement. Additionally, limiting the interaction to a joint task also simplifies the modelling of the human behaviour as the interaction is more constrained. One example of such an application is the Human Aware Task Planner (Alili et al., 2009). One property of this planner is the ability to take into account predefined social rules, such as reducing human idle time, when creating a plan specifying what the human and robot should do.

Planning performance depends heavily on the model of the environment the robot has access to. A detailed model can ensure that the robot will select the appropriate action whilst being totally autonomous. Similarly, the adaptivity depends on the model the robot has access to and whether it can update it in real time. However, in many cases when interacting with humans, the model is static, only covering a subset of the different tasks that the robot can be required to achieve and the different contexts it is expected face.

Planning have also been extended with learning, which then allows for adaptive action policies. This has been done in motion planning, to obtain a better trajectory (Jain et al., 2013; Beetz et al., 2004) and action selection planning (Kirsch, 2009). Munzer et al. (2017) also present an online learning of action policy in a collaborative task, adapting to each user and learning their preference. However, while planning is well suited for strictly defined and mostly deterministic task, many social HRI cannot be totally specified symbolically and with clear actions and outcomes.

2.2.6 Summary

Table X presents a summary of the different approaches currently used in HRI with their rating on each of the categories and how widely their are applied. As shown in the paper, the most promising type of control is *Learning from the interaction*, but it is seldom used to learn how to interact with humans. However, other fields (Reinforcement Learning and HCI) have explored ways of controlling agents included in this larger approach and we will present them in the next sections.

A last type of controller seldom used for creating action policies to interact with humans is learning from the interaction. The robot can create an action policy by exploring and learning from interacting with the world. This method assumes that the agent can update its action policy online, or regularly, and refine its knowledge of the world or at least of how to interact within it.

However, due to the complexity to create such a system and to the challenges of social interactions with humans this approach is almost never used. One example where a robot learns a non social policy online by receiving spoken human commands and explanation is using the DIARC cognitive architecture (Scheutz et al., 2017). In this paper, authors describe how a robot can learn concepts and mapping actions to verbal commands by receiving human information.

This approach is the one with the most potential as the humans could provide only the required supervision or guidance and let the robot be autonomous most of the time. Learning online an action policy provides potentially open-ended adaptivity and finally, with enough data points and the presence of a human in the action selection loop if required, the appropriateness of actions could be guaranteed.

2.3 Interactive Machine Learning

Machine learning is a promising method to provide a robot with an adequate action policy without having to implement in advance all the features used by the action selection mechanism. Offline learning is a technique allowing the robot to change its action policy over time by updating the action policy outside of the interaction. Between interactions, a learning algorithm is used to create a new action policy derived from the previous experiences.

However online learning (such as Reinforcement Learning (RL)) have the advantage of benefiting from multiple updates, constantly refining the agent behaviour, rather than a single monolithic definition or update of a behaviour. As mentioned in the previous section for the case of RL, including humans in the action selection loop can provide tremendous advantages compared to fully autonomous learning.

Interactive Machine Learning (IML), as coined by Fails & Olsen Jr (2003), differs from Classical Machine Learning (CML) by integrating an expert end-user in the learning process. In classical supervised learning, such as deep learning (LeCun et al., 2015), the learning phase happens offline once to obtain a classifier for later use. On the other hand, IML is an iterative online process using a human to correct the errors made by the algorithm as they appear.

Amershi et al. (2014) presents an introduction to IML by reviewing the work done and presenting classical approaches and challenges faced when using humans to support machine learning.

2.3.1 Goal

The main goal behind IML is to leverage the human knowledge during the learning process to speed it up. To extend the use of classifiers from static algorithms trained only once, to evolving agents learning from humans and refining their policies over time. IML can be applied in a RL approach or Supervised Learning (SL) but tries to combine advantages from both worlds. As explained in Fails & Olsen Jr (2003), classifiers gain to be fast rather than highly inductive and RL can gain from using humans to provide rewards (Knox & Stone, 2009).

By allowing a human user to see the output of an algorithms and provide additional inputs, the learning can be faster and tailored to this human desires. Using human expert knowledge and intuition, the system can achieve a better performance faster.

Additionally, a key of IML is also to empower end-users. These users are often non-technical, but possess valuable knowledge about what the robot should do. IML provides an opportunity to allow these users to design the behaviour of their robot, to teach it to behave the way they desire.

2.3.2 Active learning

Active learning is a form of teaching used in education aiming to increase student achievement by giving them a more active role in the teaching process (Johnson et al., 1991). This approach has been transferred to machine learning, and especially classifiers by allowing the learner to ask questions, query labels from an oracle for specific datapoints with high uncertainty (Settles, 2009). The typical application case is when unlabelled data are plentiful, but labels can be limited in number or costly to obtained. As such a tradeoff arises between the performance of the classifier and the quantity of queries made by the algorithm. Often this oracle would be a human annotator with the ability to provide a correct label to any datapoint.

Using an oracle to provide the label of specific points aims to both improve accuracy and speed up the learning as obtaining label of point with high uncertainty increase the precision of the algorithm and should highlight missing features in the current classifier. However, this relation between the learner and the human teacher poses questions such as:

- Which points should be selected for the query?
- How often the human should be queried?
- Who controls the interaction? (i.e. who has the initiative to trigger a query?)

Researchers have explored optimal strategies for dealing with the relation between the learner and the oracle, and this research has been especially active in HRI with robots directly asking questions to human participants and how the robot's queries could inform the teacher about the knowledge of the learning (Chao et al., 2010). In a follow up study, Cakmak et al. (2010) showed that most users preferred the robot to be proactive and involved in the learning process but they also wanted to be in control of the interaction, deciding when the robot could ask questions even if it imposed a higher workload on the teacher. Authors proposed that when teaching a complex task requiring a high workload on the teacher, the robot would probably be expected or should be encouraged to take a more pro-active stance requesting samples to take over some workload from the teacher.

Active learning, being able to select a specific sample for labelling, can dramatically improve the performance of the learning algorithm. However, in interaction, the learner is not in control of which sample to submit to an oracle to obtain a label. Datapoints are provided by the interaction and are influenced by the learner actions and the environment reaction. To tackle this issue: not being able to decide which sample to evaluate but still profiting from an active stance of the learner, Chernova & Veloso (2009) presented the Confidence Based Algorithm. The robot is initially provided with demonstrations of a correct action policy and then has to interact in the world under supervision from an human user. Authors propose that the teacher should still be able to provide corrective demonstrations in case of incorrect behaviour, but the learner should also be able to request a demonstration if the confidence of which action to select is below a threshold. That way, the learner can mitigate autonomous behaviour and human support. However, the effectiveness of this approach is bounded by the capacity of the learner to estimate this confidence in the desired policy, to request demonstrations only to prevent incorrect behaviour without relying excessively on the human oracle. Timing issues can also arise when requesting a demonstration or a label.

2.3.3 Learning from interaction

Reinforcement Learning

Concept Young infants and adults learn by interacting with their environment, by producing actions, analysing how the environment reacts and measuring progress toward a goal. Similarly, the field of Reinforcement Learning (RL) aims to empower agents by making them learn by interacting, using results from trials and errors and potentially delayed rewards to reach an optimal, or at least efficient, action policy (Sutton & Barto, 1998).

RL considers the time to be discrete, the life to be a sequence of states and actions. The simplest version of RL is modelled as a finite Markov Decision Process (MDP), a 5-tuples $(S, A, P_a(s,s'), R_a(s,s'), \gamma)$, with:

- S: a finite set of states defining the agent and environment states
- A: a finite set of actions available to the agent
- $P_a(s,s')$: the probability of transition from state s to s' following action a
- $R_a(s,s')$: the immediate reward following transition s to s' due to action a
- γ: a discount factor applied to future rewards

The goal of the RL agent is to find the optimal policy π_* maximising the discounted sum of future rewards. The agent is not aware of all the parameters of the model, and only observes the transitions between states and the rewards provided by the environment and has to update its policy to maximising this cumulated reward. Different algorithms exist to reach this policy, but the main features present in all of them is the concepts of exploration and exploitation.

Exploration reflects the idea of trying out new actions to learn more on the environment and potentially gain knowledge improving the policy whilst *exploitation* is the execution of the current best policy to maximise the current gain of rewards. All the algorithms have to balance these two features to reach an optimal action policy. One way to deal with this trade-off is to start with high probably of exploration to collect knowledge on the environment and then decrease this probably to converge toward a policy using this knowledge to make better choice of actions.

The more complex the environment is, the longer the agent has to explore before converging to a good action policy. It is not uncommon to reach numbers such as millions of iterations before reaching an appropriate action policy. And during this exploration phase, the agent's behaviour might seem erratic as the agent tries actions often randomly to observe how the environment is reacting.

Limitations As explained in the previous section, traditional RL has two main issues: requirement of exploration to gather knowledge about the environment and large number of iteration before converging. Generally, RL copes with these issues by having the agent interacting in a simulated word. This allows the agent to explore safely in an environment where its actions have no impact on the real world and where the speed of the interaction can be highly increased to gather the required datapoints in a reasonable amount of time. However, no simulator of human beings exists today which would be accurate enough to learn an action policy applicable in the real world. Learning to interact with humans by interacting with them would have to take place in the physical world, with real humans, and this implies that these issues would have direct impacts.

To gather informations about the environment, the agent needs to explore, trying out random actions to learn how the humans respond to them and if the agent should repeat them later. When interacting with humans, executing random actions can have dramatic effect on the users, presenting risk of physical harm as robot are often stiff and strong or cause distress as explained before. This reliance on random exploration presents a clear violation of the first principle to interact with humans presented earlier (all actions need to be appropriate).

Even if random behaviour were acceptable, humans are complex creatures, behaving stochastically, with personal preferences and desires. And as such, learning to interact with them from scratch would would require large number of datapoints and as interactions with humans are slow (not many actions can be tested per minute) the time required to reach an acceptable policy can prohibitive.

RL has been used in robotics despite these limitations (Kober et al., 2013), and, similarly to LfD, mostly to manipulation, locomotion or navigation tasks but not to learn social behaviours for HRI.

Opportunities Despite the limitations presented in the previous section, changes can be made to RL to increase its application to HRI.

García & Fernández (2015), insist on *safe* RL, ways to ensure that even in the early stages of the interaction, when the agent is still learning about the world, its action policy still achieve a minimal acceptable performance. Authors present two ways to achieve this safety: either use a mechanism to prevent the execution of non-safe actions or provide the agent with enough initial knowledge to ensure that it is staying in a safe interaction zone. These two methods are not limited to RL but can also be used with other machine learning techniques.

The first method (preventing the agent to execute undesired actions) can be implemented by explicitly preventing the agent to execute specific actions in predefined states, however it seems unlikely that every case could be specified in advance. As such, the easiest way in to include a human in the action selection loop, as a way to be sure that undesired actions are pre-empted before being executed. Learning by interacting with humans is called Interactive Machine Learning and is described more in depth in Section 2.3.

The second method (providing enough initial knowledge) can be achieved by carefully engineering the features used by the algorithm or starting from a initial action policy to build upon. Abbeel & Ng (2004) propose to use humans demonstrations in a fashion similar to LfD but to learn a reward function and an initial working action policy. This method, Inverse Reinforcement Learning has been applied successfully to teach a flying behaviour to a robotic helicopter. Once the initial policy and the reward function are estimated, RL is applied around the provided policy to explore and optimise the policy. That way, only small variation of the policy will happen around the demonstrated one. This small variations can ensure that policies leading to incorrect behaviours can be negatively reinforced and avoided before creating issues (such as crashing in the case of the robotic helicopter).

Whilst being promising and having been applied for agents in human environments (such as for personalised advertisement - Theocharous et al. 2015) these approaches have not been used social behaviour or to have robot interacting with humans.

2.3.4 Human as a source of feedback on actions

When the learner is agent learning an action policy (rather than a classifier) and the learner is expected to improve its behaviour by interacting with the environment, an intuitive way to steer the behaviour in the desired direction faster is to use human rewards. This approach is an adaptation of "shaping": tuning a animal's behaviour by providing rewards. In ML, using rewards from a human teacher is a *simple* way to bias the learning in the desired direction: the interface is easy, the teacher just need a way to provide a scalar or a binary evaluation of an action to steer the learning. However, this simplicity of interaction can be complex to interpret, ongoing research evaluates the best way to use this reward and combine it with environmental ones if existent (Knox & Stone, 2010) or how to understand the meaning of these rewards.

When used on their own, human rewards enable an agent to learn an action policy even in the absence of any environmental rewards, which is specially interesting to robots as it can be complex to define a clear reward function applicable to HRI or robotics in general. Early work in that field came from Isbell et al. (2006) who designed an agent to interact with a community in the LambdaMOO text based environment. Cobot, the agent had a statistical graph of users and their interactions and could execute some actions in the environment. Users of LambdaMOO could either reinforce positively or negatively Cobot's action by providing rewards. Isbell et al. presented the first agent to learn social interactions in a complex human online social environment.

While the goal of Cobot was to create an entity interacting with humans, Knox & Stone (2009) explored how a human can teach an agent an action policy with TAMER (Training an Agent Manually via Evaluative Reinforcement). The agent uses a supervised learner to model the human reward function and then takes the action that would receive the highest reward from the model. Other work explored how an agent can infer more knowledge from a reward than only its value. Advice (Griffith et al., 2013) models the confidence a learner can have in its teacher to make better use of rewards. Loftin et al. (2016) explore the strategy used by the teacher in the reward delivery: the meaning of not rewarding an action can vary between teachers, from a implicit acknowledgement of the correctness of an action to the active refusal to provide a positive reward (indicating the incorrectness of an action). MacGlashan et al. (2017) proposed COACH (Convergent Actor-Critic by Humans) to adapt the interpretation of feedback to the current policy, for example, a suboptimal policy could receive positive feedback early on when it compares positively to the average behaviour, while receiving negative feedback later on the teaching when the average agent performance is better.

Other approaches combine traditional RL with critique from humans. A human is requested to evaluate specific behaviours (Judah et al., 2010) or compare to policies to select the most appropriate one (Christiano et al., 2017).

Thomaz & Breazeal (2008) aimed to explore how humans would use feedback to teach a robot how to solve a task, here baking a cake. They used Interactive Reinforcement Learning as a way to directly combine environment rewards and human ones. However, during early studies, Thomaz et al. discovered that participants tried to use rewards to convey intention, informing the robot which part of the environment it should interact with. The next study involved two communication channels, a reward one to provide feedback on the actions and a guidance one to provide information about which part of the environment the robot should interact with. This guidance has been actively decided to be ambiguous, participants could not explicitly control the robot, but just bias the exploration. Adding this second channel improved the performance of participants.

2.3.5 Importance of control

Results from both active learning and research using human to provide feedback have shown that human teacher desire control (Amershi et al., 2014). Humans are not oracle, enjoying providing labels and evaluating an agent's actions they desire to be in control of the learning and provide richer information to the agent. Kaochar et al. (2011) have shown than when given choice between different teaching methods, humans will never choose to limit themselves to use only feedback, but they want to teach using more modalities.

In addition to improve the teacher's experience in the teaching, providing the teacher with more control can improve the learning. By allowing the teacher to demonstrate online an action policy or pre-empt undesired actions the robot is about to execute, the learner can interact mostly in useful states of the environment, learn faster and improve its performance in early stages of the learning.

However providing the teacher with this control presents challenges for designing the interaction. Unlike a simple scalar for reward, being able to control the robot requires the teacher to be able to ask the robot to execute any actions, which can be complex when the action space is bigger than few actions. Similarly, to give the opportunity to the teacher to pre-empt undesired actions, the learner needs to communicate its intentions to the teacher.

2.4 Summary

This chapter presented first an overview of fields where robots interact socially with humans. From these case of application, three principle have been defined that a robot controller should follow to interact efficiently with humans, the robot should:

1. Only execute appropriate actions.
2. Have a high level of adaptivity.
3. Have a high level of autonomy.

A review of current controller for robots in HRI reported that no approach applied today in the field validates these principles. The review was extended to more general methods in Machine Learning with potential to satisfy these principles. Interactive Machine Learning show promises for enabling a robot to learn how to interact with humans, however while humans have been used to teach robot behaviours or concepts, teaching them to interact with human in an interactive, online fashion has not been demonstrated in the field so far and could satisfy all these requirements.

Chapter 3

Supervised Progressive Autonomous Robot Competencies

Key points:

- Novel interaction framework to teach robots an action policy while interacting.
- A human teacher is in control of the robot actions whilst the robot learns from this supervision.
- The teacher provides feedback on intentions rather than actions.
- The robot behaviour (under supervision) can be assumed to be optimal.
- Workload on the teacher decreases over time as the robot learns.

Parts of the work presented in this chapter have been published in Senft et al. (2015) and Senft et al. (2017a). The final publications are available from Springer and Elsevier via `http://dx.doi.org/10.1007/978-3-319-25554-5_60` and `https://doi.org/10.1016/j.patrec.2017.03.015`.

As presented in Chapter 2, robots would profit from being able to learn from humans how to interact with other humans. Using IML to achieve this transfer of social and task knowledge from the human domain-expert to the robot would result in a faster learning than slow iterative update of behaviour by engineering an action policy or learning by trials and errors as with RL.

However, as stated in that chapter, IML is seldom applied to learning to interact with humans and no current system provides the teacher with enough control over the robot action to ensure that the first principle presented in Section 2.1.2 (“Only execute appropriate actions”) is validated. Techniques relying solely on feedback cannot prevent the robot to execute an incorrect action, but only reward negatively incorrect actions after their execution (Senft et al., 2017a) and with techniques based on LfD the teacher relinquishes its control over actions executed by the robot.

In order to provide a robot with an appropriate action policy, adaptive to different context or behaviours and requiring a low workload on the teacher or supervisor, we introduced in Senft et al. (2015) the Supervided Progressive Autonomous Robot Competenties (SPARC) framework of interaction to allow end-users to safely teach a robot an action policy applicable to HRI.

3.1 Principles

SPARC defines an interaction between a learner and a teacher following these principles:

- The learner has access to a representation of the state and a set of actions.
- The teacher can select actions for the robot to execute.
- The learner can propose actions to the teacher.
- The teacher can enforce or cancel actions proposed by the learner and actions non evaluated are implicitly validated and executed after a small delay.
- The learner improves its action policy using the teacher’s commands and feedback on propositions.

q This way of keeping a human in the loop and in control of the robot’s actions is similar to the level 6 on the Sheridan scale of autonomy: "A computer selects action, informs human in plenty of time to stop it" (Sheridan & Verplank, 1978) with the addition that the human can also select actions for the robot to execute. In addition, a learning algorithm improve the correctness of suggested actions decreasing the probability of requiring the teacher to correct actions or having to select new actions. Additionally, keeping the human in the loop also gives them opportunities to provide additional information to the algorithm to speed up the learning.

This approach can be compared to predictive texting as seen on phone nowadays. The user can select the words proposed by the algorithms (or implicitly accept then by pressing space) or write their own. The algorithm learns the user's preferences and habits and aims to suggest words more and more appropriate to the user. However, SPARC enables a more complex interaction between learner and teacher and is directed to interactive environments in continuous time where the context changes with time both dependently and independently to the agent actions.

3.2 Goal

The goal of SPARC is to allow non-experts in computer science to teach quickly an action policy to a robot by guiding its interaction within an environment, without requiring constant input from a human and whilst ensuring that the robot's behaviour is constantly appropriate.

Figure 3.1 presents an idealist comparison of the expected workload, performance and autonomy of three methods: autonomous learning (such as RL - Sutton & Barto 1998), feedback based teaching (such as TAMER - Knox & Stone 2009) and SPARC. Unlike other methods, by following the principles presented in the previous section, SPARC is expected to maintain a constant high performance even during early stages of learning, and to see a decrease of workload on the human teacher as the agent improve its action policy using machine learning and the suggestions become more accurate.

Once the behaviour is deemed appropriate enough by the teacher, the agent can be deployed to interact autonomously in the real world if this output is desired. Alternatively, in context where a human expert cannot be removed from the control loop, such as Robot Assisted Therapy, the teacher can stay in control of the robot actions in a supervised autonomous way (similar to the teaching phase) and only intervenes when the agent

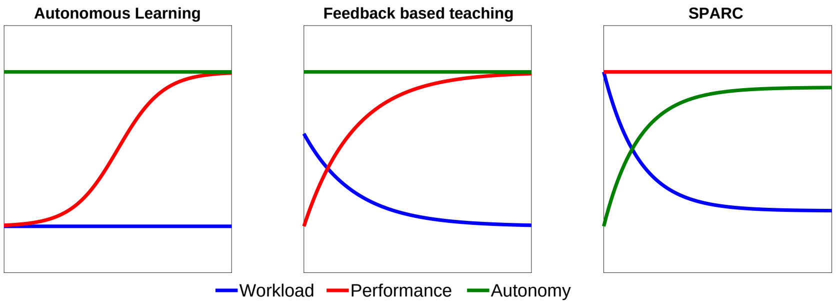


Figure 3.1: Idealistic comparison between autonomous learning, feedback base teaching and SPARC.

is about to do an error. This approach is similar to safety drivers behind autonomous vehicles but with information about the car's intentions rather than solely the car's actions.

3.3 Frame

Similarly to other applications of IML, SPARC requires the interaction with a teacher to learn an action policy to interact with the world. In this framework, the robot interacts with two entities: the target and the teacher, which results into two interlinked interactions: the application interaction (task the robot learns to achieve) and the control interaction (relation with the teacher), as demonstrated in Figure 3.2. In the generic case, the overall interaction is a triadic interaction (Teacher-Robot-Human target), such as a teacher teaching a robot tutor how to support child learning (as implemented in Chapter 6). But in specific cases, the overall interaction can be only dyadic (Teacher-Robot-Teacher or Teacher-Robot-Environment), such as robot at home learning from its user how to support them better.

However, unlike some other IML approaches, the requirement of a human in the action selection loop limits the timescale of interaction. As the human has to be provided with few seconds to react to the proposition, the rate of actions has to be 0.5 Hz or below. However, this can be mitigated by using higher level actions and this has the advantage to ensure that only correct actions will be executed without requiring the human to select them all.

SPARC presents many similarities with Learning from Demonstration (cf. Section 2.2.4) as it uses human demonstration of policies to learn. However, most of the applications of LfD (Argall et al., 2009; Billard et al., 2008) are focused on learning a manipulation skill in a mostly deterministic environment. LfD has seldom been used to teach an action policy to interact with humans (Liu et al., 2014; Sequeira et al., 2016) and never in an online fashion.

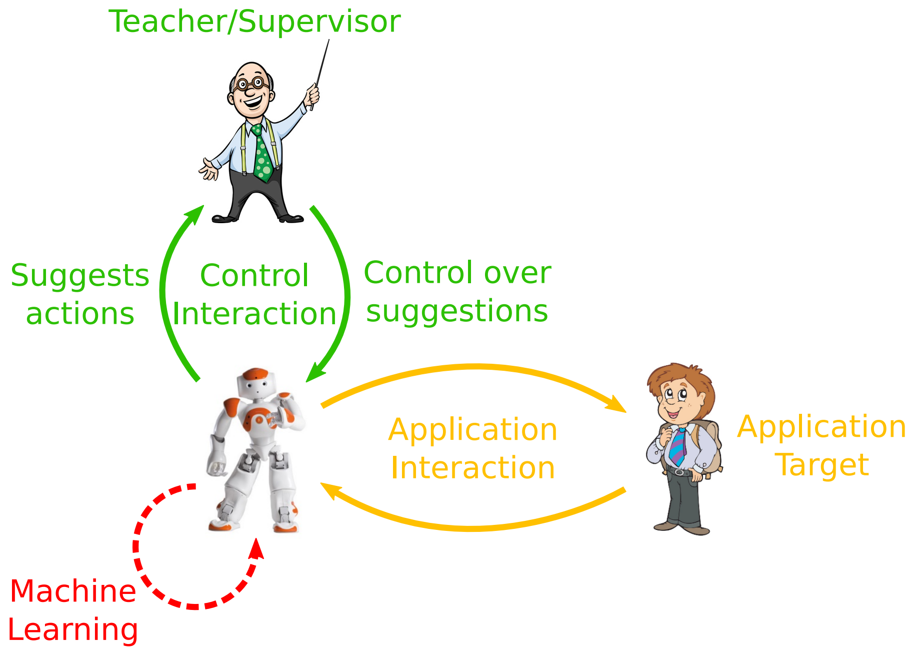


Figure 3.2: Frame of the interaction: a robot interact with a target and suggests actions and receive commands from a teacher.

Additionally, SPARC differs from Active Leaning (cf. Section 2.3.2) by the fact that the agent cannot decide which sample will be evaluated by the teacher. As the robot interacts with humans, the datapoints provided to the teacher for *labelling* will emerge from the interaction, and cannot be selected at will.

3.4 Implications

The principles described in section 3.1 have also implications on the interactions between the teacher and the agent and between the agent and the environment.

As the teacher can evaluate the actions proposed by the agent before their executions, it actually evaluates the intentions of the agent rather than its behaviour, and this difference is key as traditional IML approaches only evaluate the actions or their effects, but not the intentions. This implication is fundamental in HRI as we cannot accept to have a robot executing non appropriate actions when interacting with humans (cf. first principle in Section 2.1.2) and correctly evaluating intentions and not actions gives the opportunity to the teacher to pre-empt incorrect actions preventing the execution of undesired behaviours. The learner learns the expected impact of actions without having to handle the results of the execution of this action.

Most importantly, the control given to the teacher on the agent's actions ensures that every action executed by the learner in the world has been either actively or passively validated by the teacher. This implies that each action executed can be assumed to be appropriate to the current state, potentially simplifying the learning algorithm.

3.5 Interaction with Machine Learning Algorithms

The principles of SPARC define it at the crossroads between Supervised Learning and Reinforcement Learning. The goal of the algorithm could be either to reproduce the teacher's policy, in a supervised way or using the teacher to bias the exploration and learn an action policy from the environment under the supervision of the teacher.

As such, SPARC only defines an interaction framework between a teacher and a learner, and is agnostic to the learning mechanism: it can be combined with any algorithms used in Supervised Learning or Reinforcement Learning. This research explored a combination with three types of algorithms: supervised learning using feed-forward neural networks in Chapter 4, reinforcement learning in Chapter 5 and supervised learning using instance based algorithm in Chapter 6. However SPARC could be combined with a wide range of other algorithms.

Similarly to Inverse Reinforcement Learning (Abbeel & Ng, 2004) or other techniques based RL and LfD (Billard et al., 2008), if combined with a reward function SPARC could go beyond the demonstrated action policy and achieve a performance higher than the demonstration. But this has not been evaluated in this work.

3.6 Summary

This chapter introduced Supervided Progressive Autonomous Robot Competenties (SPARC), a novel interaction framework to teach agent an action policy. This approach is suitable to teach a robot to interact with humans as it validates the principles defined in Section 2.1.2. SPARC starts in a similar fashion than WoZ: the teacher can select which action the robot should do, then using a learning algorithm, the learner proposes actions to the teacher who can let them be executed after a short time or cancel the action and select another one if appropriate. This suggestions/corrections mechanism provides the appropriateness of actions as a human could have cancel any inappropriate action in the learning phase and provides the adaptivity as the teacher can extend the behaviour beyond the current action policy. The learning algorithm with the auto-executions ensure that once the robot has learnt, the human workload is low and the robot could even be deployed to interact autonomously if this is desired.

Chapter 4

Relation with Wizard of Oz

Key points:

- An experiment was designed to explore the influence of SPARC on the human workload and task performance compared to an approach based on WoZ.
- Application target replaced by a robot to ensure repeatability of target behaviour.
- Design of a robot model exhibiting probabilistic behaviour with a non-trivial optimal interaction policy.
- Results show that SPARC achieves a similar performance than WoZ while requiring a lower workload from the teacher.

Parts of the work presented in this chapter have been published in Senft et al. (2015) [1] . The final publication is available from Springer via `http://dx.doi.org/10.1007/978-3-319-25554-5_60`.

[1] Note about technical contribution in this chapter: the author used software from the DREAM project for the touchscreen and the robot functionalities. The author contributed to the material used within the robot control and the Graphical User Interface. Algorithm used from the OPENCV neural network library.

4.1 Motivation

SPARC as been designed to enable end-users non-experts in computer science to teach a robot an action policy while interacting in a sensitive environment. The argument behind this way of interacting is that SPARC allows a field expert to transfer their knowledge to an autonomous agent without wasting time to teach the agent and having to enforce each actions manually. As the agent is interacting in the target environment, displaying a appropriate action policy, the time spend to teach it is not lost as the desired interaction takes place also during the learning phase. For example, using the context of RAT, the therapist would teach the robot during a therapy session. As the therapist is in total control of the robot's action, the behaviour expressed by the robot fits the desired behaviour desired for the therapy.

In essence, SPARC, as a principle, allows to start a robotic application in a WoZ fashion, and then move away from it as the robot gains autonomy. The aim of SPARC is two fold: maintaining a high level of performance while reducing the workload of the teacher over time until reaching a point where the robot is autonomous or only necessitate minimal supervision to interact successfully. As explained in Chapter 3, SPARC involves two interactions the control interaction and the application ones. And when the goal is learning how to interact with humans, the robot is interacting simultaneously with two humans. These two dependent interactions complexify the evaluation of the approach, especially as both humans are impacting each other. The first step to evaluate SPARC was to focus on the relation between the robot and its teacher. To evaluate this aspect of the interaction, we decided to replace the target of the application by a robot running a model of a child and observe the impact of SPARC on the teacher. The setup ends up with two robots interacting together (the wizarded-robot and the child-robot) whilst the wizarded-robot is controlled by a participant. The child-robot has some inner variables (motivation and engagement) and has to keep them high to achieve a good performance.

4.2 Hypotheses

To evaluate the validity of SPARC and the influence of such an approach, four hypotheses were devised:

H1 A 'good' teacher (i.e. keeping the motivation and engagement of the child-robot high) will lead to a better child-robot performance.

H2 When interacting with a new system, humans will progressively build a personal strategy that they will use in subsequent interactions.

H3 Reducing the number of interventions required from a teacher will reduce their perceived workload.

H4 Using SPARC allows the teacher to achieve similar performance with fewer interventions than WoZ.

H1 represents a sanity check for the model, ensuring that the expressed performance represents the efficiency of the action policy demonstrated by the teacher. H2 tests that human teachers are not static entities, they adapt their learning target and their teaching strategy. H3 tests one of the motivations behind SPARC: does reducing the number of physical actions required for a robot to interact while requiring the teacher to monitor the robot suggestions lead to a lower workload. And finally, H4 is the main hypothesis, does SPARC enable a robot to learn a useful action policy: reducing the teacher's workload while maintaining a high performance.

4.3 Methodology

This study is based on a real scenario for RAT for children with ASD based on the Applied Behaviour Analysis therapy framework. The aim of the therapy is to help the child to develop/practice their social skills. The child has to complete an emotion recognition task involving a child playing a categorisation game with a robot on a mediating touchscreen device Baxter et al. (2012). And the robot can provide feedback and prompts to encourage the child and help them to classify emotions. Images of faces or drawings are shown to the child, and they have to categorise them by moving the image to one side or the other depending on whether the picture shown denotes happiness or sadness (e.g. fig. 1.1). In the therapy, the robot is remote controlled by an operator using the Wizard-of-Oz paradigm, and does not interact with the child directly.

This study explores if SPARC can be used to teach the robot a correct action policy to interact with the child. As timing in human-robot interactions is complex, for simplification

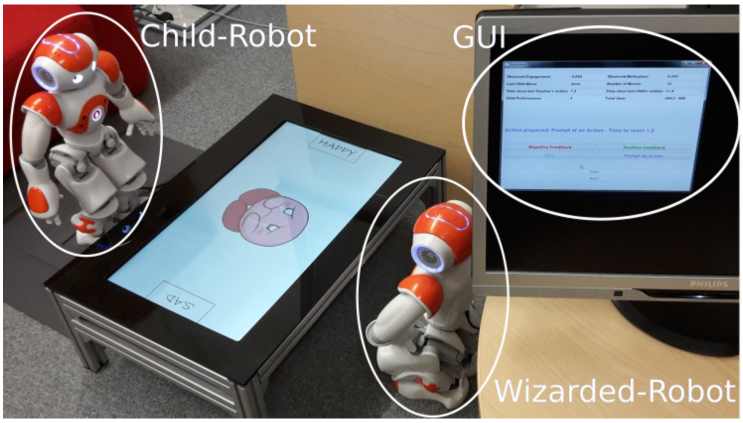


Figure 4.1: Setup used for the user study from the perspective of the human teacher. The *child-robot* (left) stands across the touchscreen (centre-left) from the *wizarded-robot* (centre-right). The teacher can oversee the actions of the *wizarded-robot* through the GUI and intervene if necessary (right).

reasons, the interaction has been discretised to have clear steps when the robot has to select an action. The basic interaction structure following SPARC is as follows:

1. the robot suggests an action to the teacher
2. the teacher can select an action for the robot to execute or let the proposed action be executed
3. the robot executes the selected action
4. both robot and teacher observe the outcome of the action until the next action selection step

Using SPARC, over time, the robot learns to replicate the teacher's policy by matching the inputs (child's state) to the outputs (action selected by the teacher).

Two conditions are compared: SPARC, where the robot learns from the human corrections and the WoZ condition, where the robot proposes a random actions instead of learnt actions to simulate a WoZ setup where the teacher would have to select every the actions.

The focus of the study being on the *control interaction* (the relation between the teacher and the robot), the second interaction (the application one) has been kept constant by replacing the child by a robot. A minimal model of child behaviour is therefore used to stand in for a real child. A second robot is employed in the interaction to embody this child model: we term this robot the *child-robot* while the robot being directly guided by the human teacher is the *wizarded-robot* (Figure 4.1).

4.3.1 Child model

The purpose of the child model is not to realistically model a child (with or without autism), but to provide a means of expressing some characteristics of the behaviours we observed in interactions with children in a repeatable manner. The child-robot possesses an internal model encompassing an *engagement* level and a *motivation* level, together forming the *state* of the child. The engagement represents how often the child-robot will make categorisation moves and the motivation gives the probability of success of the categorisation moves. Bound to the range $[-1, 1]$, these states are influenced by the behaviour of the wizarded-robot, and will asymptotically decay to zero without any actions from the wizarded-robot. These two states are not directly accessed by either the teacher or the wizarded-robot, but can be observed through behaviour expressed by the child-robot: low engagement will make the robot look away from the touchscreen, and the speed of the categorisation moves is related to the motivation (to which gaussian noise was added). There is thus incomplete/unreliable information available to both the wizarded-robot and the teacher, making the task non-trivial.

The influence of the wizarded-robot behaviour on the levels of engagement and motivation are described below (Section 4.3.2). In addition to this, if a state is already high and an action from the wizarded-robot should increase it further, then there is a chance that this level will sharply decrease, as an analogue of *frustration*. When this happens, the child-robot will indicate this frustration verbally (uttering one of eight predefined strings). This mechanism prevent the optimal strategy to be straightforward: always making actions aiming to increase motivation or engagement. The optimal strategy includes these actions but also waiting times to prevent the state values to overshoot. This non-trivial optimal strategy approximates better a real human-robot interaction scenario requiring a more complex strategy to be employed by the teacher.

4.3.2 Wizarded-robot control

The wizarded-robot is controlled through a GUI and has access to multiple variables characterising the state of the interaction as used by the learning algorithm:

- Observed engagement
- Observed motivation

- Type of last move made by the child-robot (good/bad/done)

Additionally, other metrics are displayed to the teacher but not used in the algorithm:

- Number of categorisations made by the child-robot
- Time since last teacher's action
- Time since last child's action
- Child's performance
- Total time elapsed

The wizarded-robot has a set of four actions, with one button each on the GUI:

- Prompt an Action: Encourage the child-robot to do an action.
- Positive Feedback: Congratulate the child-robot on making a good classification.
- Negative Feedback: Supportive feedback for an incorrect classification.
- Wait: Do nothing for this action opportunity, wait for the next one.

The impact of actions on the child-robot depends on the internal state and the type of the last child-robot move: good, bad, or done (meaning that feedback has already been given for the last move and supplementary feedback is not necessary). A *prompt* increases the engagement, a *wait* has no effect on the child-robot's state, and the impact of positive and negative feedback depends on the previous child-robot move. Congruous feedback (positive feedback for correct moves; negative feedback for incorrect moves) results in an increase in motivation, but incongruous feedback can decrease both the motivation and the engagement of the child-robot. The teacher therefore has to use congruous feedback and prompts.

However, as mentioned in Section 4.3.1, if engagement or motivation cross a threshold, their value can decrease abruptly to simulate the child-robot being frustrated. This implies that the optimal actions policy provides congruous feedback and prompts, but also requires wait actions to prevent the child-robot becoming frustrated and keep its state-values close to the threshold without crossing it. A 'good' strategy keeping the engagement and motivation high, leads to an increase in performance of the child-robot in the categorisation task.

To simplify the algorithm part, the interaction has been discretised: the teacher can not selects actions for the wizarded-robot at any time, actions can only be executed at specific times triggered by the wizarded-robot: two seconds after each child-robot action, or if nothing happens in the interaction for five seconds. When these selection windows are hit, the wizarded-robot proposes an action to the teacher by displaying the action's name and a countdown before execution. Only after this proposition has been done can the teacher select a different action for the wizarded-robot or let the proposed one be executed: if the teacher does nothing in the three seconds following the suggestion, the action proposed by the wizarded-robot is executed. This mechanism allows the teacher to passively accept a suggestion or actively make an *intervention* by selecting a different action and forcing the wizarded-robot to execute it.

4.3.3 Learning algorithm

In the SPARC condition, the robot learns to reproduce the action policy displayed by the teacher. For this study, the robot learns using a Multi-Layer Perceptron (MLP): with five input nodes: one for the observed motivation, one for the observed engagement and three binary (+1/-1) inputs for the type of the previous move: good, bad, or done. The hidden layer had six nodes and the output layer four: one for each action. The suggested action is selected applying a Winner-Take-All strategy on the value of the output node and then displayed on the GUI before execution. The network is trained with back propagation: after each new decision from the teacher a new training point is added with the selected action node having +1 while the others -1. The network is fully retrained with all the previous state-action pairs and the new one at each selection step.

This learning algorithm, MLP is not optimal for a real time interaction as the online learning should happens quickly between learning iteration. However as the length of interaction (and so the number of datapoints) is limited and the desired learning behaviour is purely supervised learning, this type of algorithm has been deemed suitable for this study.

4.3.4 Participants

In RAT scenarios using WoZ to control the robot, the wizard is typically a technically competent person with previous experience controlling robots or at least significant training controlling this robot for the therapy. As such, to maintain consistency with the target user group, the participants of this study (assuming the role of the teacher) have been taken from a robotics research group. Ten participants were used (7M/3F, age *M*=29.3, 21 to 44, *SD*=4.8 years).

4.3.5 Interaction Protocol

The study compared two conditions: a learning robot adapting its propositions to its user (the SPARC condition) and a non-learning robot constantly proposing random actions (the WoZ condition). The child-robot controller was kept constant in both conditions, while the state is reset between interactions. The design was a within subjects comparison with balancing of order: each participant interacted with both conditions, with the order balanced between participants to control for any ordering effects. In *order LN* the participants first interact with the learning wizarded-robot in the SPARC condition, and then with the non-learning one in the WoZ condition; in *order NL* the order of interaction is inverted. Participants were randomly assigned to one of the two orders.

The interactions took place on a university campus in a dedicated experiment room. Both robots were Aldebaran Nao, one of which had a label indicating that it was the *Child-Robot*. The robots faced each other with a touchscreen between them, and participants assuming the role of the teacher sat at a desk to the side of the wizarded-robot, with a screen and a mouse to interact with the wizarded-robot (fig. 4.1). Participants were able to see the screen and the child-robot.

A document explaining the interaction scenario was provided to participants with a demographic questionnaire (cf Annexes.). After the information had been read, a 30s video presenting the GUI in use was shown to familiarise the participants with it, without biasing them towards any particular control strategy. Then participants clicked a button to start the first interaction which lasted for 10 minutes. The experimenter was sat in the room outside of the participants' field of view. After the end of the first interaction, a post-interaction questionnaire was administered. The same protocol was applied in the second part of the experiment with another post-interaction questionnaire following. Finally, a post-experiment questionnaire asking participants to explicitly compare the two conditions was administered.

4.3.6 Measures

Two types of measures have been recorded for this study: interaction data representing objective behaviours and performance of the participants and subjective data through questionnaires.

Interaction data The state of the child-robot and the interaction values were logged at each step of the interaction (at 5Hz). All of the human actions were recorded: acceptance of the wizarded-robot's suggestion, selection of another action (intervention), and the states of the child-robot (motivation, engagement and performance) at this step.

The first metric is the performance achieved in each interaction measured by the number of correct categorisations made by the child-robot minus the number of incorrect ones. This represents how correct was the action policy executed by the wizarded-robot when controlled by a participant.

The second important metric is the intervention ratio: the number of times a user chooses a different action than the one proposed by the wizarded-robot, divided by the total number of executed actions. This metric represents how often in average a user had to correct the robot and could be related to the workload the user had to face to control the robot.

Questionnaire data Participants answered to four questionnaires: a demographic one, before the interaction, two post-interaction ones where they were asked to evaluate the last interaction with the robots and a post-experiment where they had to compare the two conditions and select the one corresponding to a description. All the rating questionnaires used seven items Likert scale.

Post-Interaction

- The child-robot learned during the interaction
- The performance of the child-robot improved in response to the teacher-robot actions
- The teacher-robot is capable of making appropriate action decisions in future interactions without supervision
- The teacher-robot always suggested an incorrect or inappropriate actions

- By the end of the interaction, my workload was very light
- What did you pay most attention during the interaction? (child-robot, touchscreen, GUI, other)

Post-experiment

- There was a clear difference in behaviour between the two teacher-robots
- There was a clear difference in behaviour between the two child-robots
- Which teacher-robot was better able to perform the task? (first, second)
- Which teacher-robot did you prefer supervising? (first, second)

4.4 Results

4.4.1 Interaction data

Figure 4.2 presents the aggregated (results collapsed between orders) performance and intervention ratio for both conditions. While the number of participants are not sufficient to perform statistical comparison, overall interaction results seem to show that both conditions lead to similar performance (SPARC: 32.6 (95% CI [27.89,37.31]) - WoZ: 31.4 (95% CI [25.9,36.9])) while the SPARC condition required less intervention (SPARC: 0.38 (95% CI [0.29,0.47]) - WoZ: 0.59 (95% CI [0.52,0.67])).

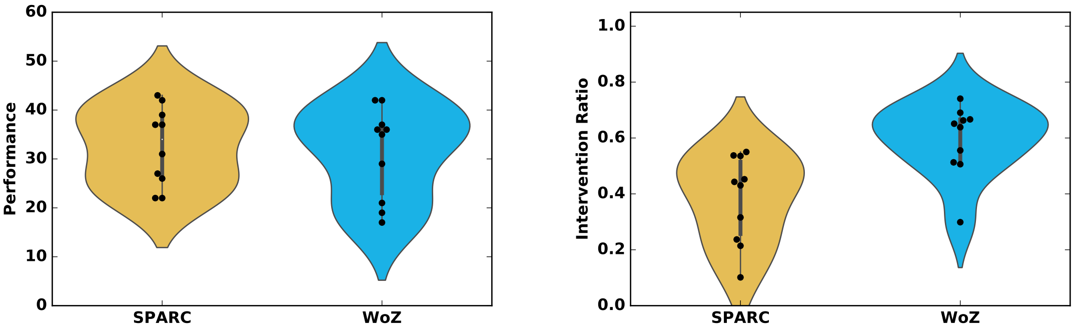


Figure 4.2: Aggregated comparison of performance and intervention ratio for both conditions

Figure 4.3 presents the evolution of intervention ratio for each condition and orders. During the first interaction, participants discover the interface and how to interact with

Table 4.1: Average performance and intervention ratio separated by condition and order.

	Order LN		Order NL	
	SPARC (int 1)	WoZ (int 2)	WoZ (int 1)	SPARC (int 2)
Performance M	29.6	38.2	24.6	35.6
95% CI	[23.61,35.59]	[35.46,40.94]	[18.1,31.1]	[29.34,41.86]
Intervention Ratio M	0.31	0.68	0.5	0.46
95% CI	[0.17,0.45]	[0.65,0.71]	[0.4,0.61]	[0.38,0.53]

it, which results in a high variation intervention ratio in the first 20 steps (each time the wizarded-robot proposes an action). However in the second phase of the interaction, when participants develop their teaching policy, there is a tendency of SPARC requiring a lower number of intervention than WoZ. This effect is higher in the second interaction, where as soon as 5 steps, the two conditions differentiate without overlap of the 95% CI of the mean, which would indicate that the two conditions differ in term of required interventions.

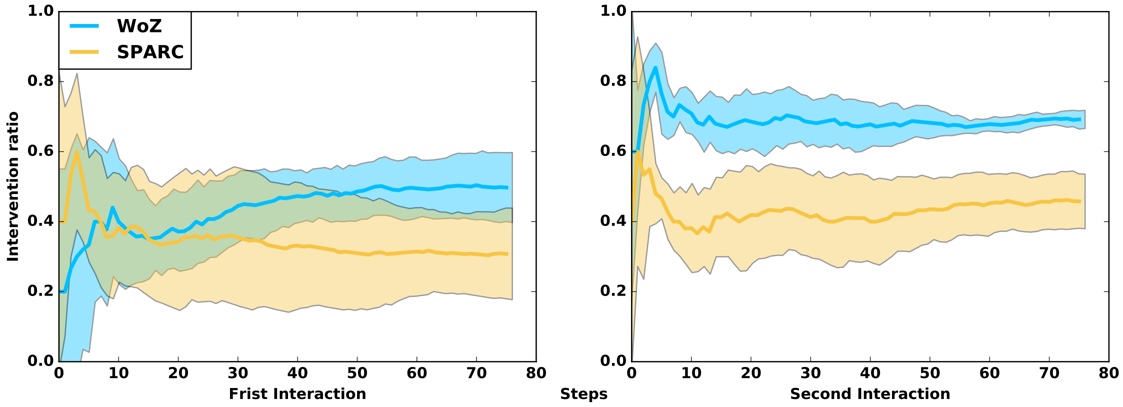


Figure 4.3: Evolution of intervention ratio over time for both conditions and both orders. Shaded area represents the 95% CI.

Both for the performance and the intervention ratio, a strong ordering effect have been observed. Figure 4.4 and Table 4.1 present the performance and intervention ratio separated by condition and order. In both orders, the second interaction had higher performance as the participants were used to the system and understood how to develop an efficient interaction policy. And the performance between condition is similar. However, regardless of the order, when only the interaction number is considered, the intervention ratio is lower when using SPARC compared to WoZ. This indicates that when the wizarded-robot learned using SPARC, a similar performance is attained as with WoZ, but the number of interventions required to achieve this performance is lower.

Additionally, a strong positive correlation (Pearson's r=0.79) was found between the average child-robot motivation and engagement and its performance which shows that

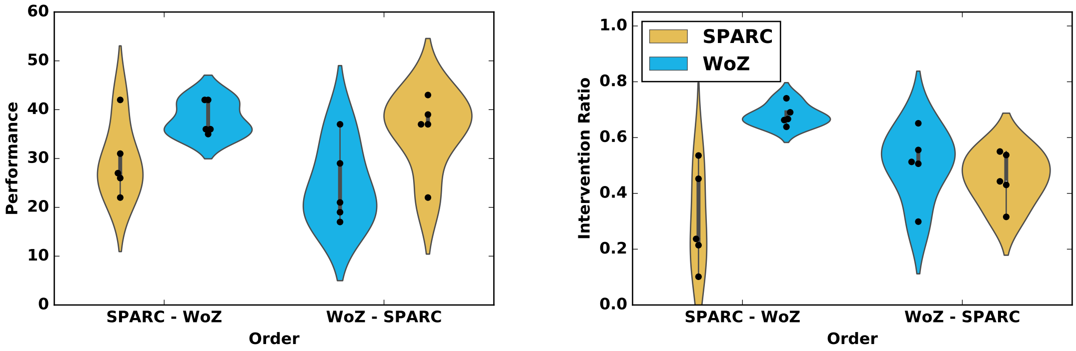


Figure 4.4: Performance achieved and intervention ratio separated by order and condition. For each order, the left part presents the performance in the first interaction (with one condition) and the right part the performance in the second interaction (with the other condition).

the performance achieved by the child-robot represents the capacity of the teacher to keep both engagement and motivation high.

4.4.2 Questionnaire data

The post-interaction questionnaires evaluated the participant's perception of the child-robot's learning and performance, the quality of suggestions made by the wizarded-robot, and the experienced workload. All responses used seven point Likert scales.

Table 4.2 presents separated results for the questions asked in the post-interaction questionnaires, with more details for the questions exhibiting differences in Figure 4.5.

Across the four possible interactions, the rating of the child-robot's learning was similar (*M*=5.25, 95% CI [4.8, 5.7]). As the child-robot was using the same interaction model in all four conditions, this result is expected. There is a slight tendency to rate the child's performance as being higher but the error margin is too high to conclude anything. This could indicate that the teachers were more aware of the child's behaviour as the workload was lighter to control the wizarded-robot.

Participants rated the wizarded-robot as more suited to operate unsupervised in the learning than in the non learning condition (confidence interval of the difference of the mean (CIDM) for LN ordering [-0.2, 2.6], CIDM for the NL ordering [1.6, 2.8]).

Similarly, a trend was found showing that learning wizarded-robot is perceived as making fewer errors than the non-learning robot (CIDM for LN ordering [1.3, 3.4], CIDM for the NL ordering [0.1, 1.1]).

Table 4.2: Average reporting on questionnaires separated by condition and order.

	Order LN		Order NL	
	SPARC (int 1)	WoZ (int 2)	WoZ (int 1)	SPARC (int 2)
Child learns M	5.2	5.2	5.2	5.4
95% CI	[3.69,6.71]	[3.8,6.6]	[4.18,6.22]	[4.7,6.1]
Child's performance M	4.6	5.0	5.0	4.4
95% CI	[3.41,5.79]	[3.25,6.75]	[4.04,5.96]	[3.7,5.1]
Wizarded-robot makes errors	1.6	4.0	2.6	2.0
M 95% CI	[0.9,2.3]	[2.76,5.24]	[1.9,3.3]	[2.0,2.0]
Wizarded-robot makes appropriate	4.8	3.6	3.0	5.2
decisions M 95% CI	[3.4,6.2]	[2.18,5.02]	[2.04,3.96]	[4.85,5.55]
Lightness of workload M	4.6	3.6	3.8	5.4
95% CI	[3.41,5.79]	[1.8,5.4]	[2.94,4.66]	[4.35,6.45]

The participants tended to rate the workload as lighter when interacting with the learning robot, and this effect is much more prominent when the participants interacted with the non-learning robot first (CIDM for LN ordering [-0.6, 2.6], CIDM for the NL ordering [0.7, 2.5]).

Most of the difference of mean interval exclude 0 or include it marginally, which would indicate tendency of difference (due to the low number of participants, no statistical tests are applicable).

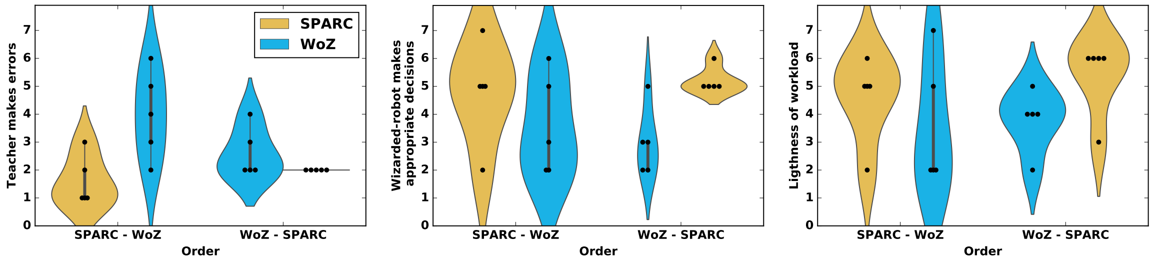


Figure 4.5: Questionnaires results on robot making errors, making appropriate decisions and on lightness of workload.

4.5 Discussion

Strong support for H1 (a good teacher leads to a better child performance) was found, a correlation between the average states (engagement and motivation) and the final performance for all of the 10 participants was observed (*r*=0.79). This sanity check confirms that the performance of the child robot reflects the performance of the teacher in this task: teaching the wizarded-robot an efficient action policy maximising the inner state of the child-robot. Additionally, the model of the child robot exhibited the desired behaviour:

allowing a wide range of performances without one obvious optimal action policy.

The results also provide support for H2 (teachers create personal strategies): all the participants performed better in the second interaction than in the first one. This suggests that participants developed a strategy when interacting with the system in the first interaction, and were able to use it to increase their performance in the second interaction. Looking in more detail at the interaction logs, different strategies for the wizarded-robot can be observed with a varying level of waiting actions compared to other types of actions.

H3 (reducing the number of interventions will reduce the perceived workload) is partially supported: the results show a trend for participants to rate the workload as lighter when interacting with the learning robot, and another trend between using a learning robot and the intervention ratio. However, when computing the correlation between the intervention ratio and the reported workload, a strong effect can only be observed in the second interaction ($\rho = -.622$). In the first interaction, the main effect of the workload is probably the discovery of the system and how to interact with it. Nevertheless, regardless of the order of the interactions, the learning robot consistently received higher ratings for lightness of workload which indicates that using SPARC could decrease workload on robot's supervisor compared to WoZ.

Finally, H4 (using learning keeps similar performance, but decreases interventions) is supported: interacting with a learning robot results in a similar performance than interacting with a non-learning robot, whilst requiring fewer active interventions from the supervisor. This has real world utility, it frees some time for the supervisor, to allow them to focus on other aspects of the intervention, e.g. analysing the child's behaviour rather than focusing on the robot control.

It should be noted that the actual learning algorithm used in this study is only of incidental importance, and that certain features of the supervisor's strategies may be better approximated with alternative methods – of importance for the present work is the presence of learning at all. Other algorithm and ways to handle time have been used in the following studies.

4.6 Summary

As expected, using SPARC to teach a robot to interact under supervision did allow the robot to partially learn an interaction policy which decrease the requirement on the teacher to physically enforce each robot's actions. Additionally, SPARC decreased also the workload imposed on the robot supervisors which has real world impact as today many subfields of HRI such as RAT still rely on WoZ for their robotic control.

Using a suggestion/intervention system, SPARC allowed online learning for interactive scenarios, thus increasing autonomy and reducing the demands on the supervisor. Results showed that interacting with a learning robot allowed participants to achieve a similar performance as interacting with a non-learning robot, but requiring fewer interventions to attain this result. This suggests that while there is always adaptation in the interaction (leading to similar child-robot performance given the two wizarded-robot controllers), the presence of learning shifts this burden of adaptivity onto the wizarded-robot rather than on the human. This indicates that a learning robot could allow the therapist to focus more on the child than on the robot, with improved therapeutic outcomes as potential result.

Chapter 5

Keeping the user in control

Key points:

- An experiment was designed to compared SPARC to another approach in IML: Interactive Reinforcement Learning (IRL) using feedback and partial guidance to teach a robot an action policy.
- Application domain is a replication of the world used in early studies on IRL.
- SPARC uses full control over the robot's action, implicit rewarding system and evaluation of intentions rather than actions.
- SPARC was combined with Reinforcement Learning.
- Results show that SPARC achieves a better performance, easier and faster than IRL.

Parts of the work presented in this chapter have been published verbatim in Senft et al. (2017a) [1]. The final publication is available from Elsevier via `https://doi.org/10.1016/j.patrec.2017.03.015`.

[1] Note about technical contribution in this chapter: the author reimplemented every part of the system manually in Qt.

5.1 Motivation

Previous work in IML showed that humans want to teach robots not only with feedback but also by communicating what the robot should do (Thomaz & Breazeal, 2008). However, in most of the research teaching agents a policy using human guidance, the teacher is given little or no control at all on the agent's actions and has to observe the agent executing an action even when knowing that this action is incorrect. This chapter explores how these IML approaches could be pushed further by applying the principles of SPARC defined in Chapter 3, how it would influence the learning process, the agent performance and the user experience and how these results would compare to other IML approaches.

Additionally, the previous study explored how SPARC could be used with Supervised Learning, to replicate a teacher's action policy, but some of the most promising features of IML arise when combined with Reinforcement Learning (RL). As such this chapter proposes a way to apply the principles underlying SPARC to classical feedback based RL and evaluates how this human control over the robot's actions impacts the learning. A study involving 40 participants compares SPARC to another interaction approach offering less control but having been validated in previous studies: Interactive Reinforcement Learning (IRL) (Thomaz & Breazeal, 2008). The testing environment of Interactive Reinforcement Learning (IRL) have be reimplemented to stay as close as possible to the online version of the task.

5.2 Scope of the study

5.2.1 Interactive Reinforcement Learning

The IRL algorithms implements the principles presented in Thomaz & Breazeal (2008): the human teacher can provide positive or negative feedback on the last action executed by the robot and the robot combines this with environmental feedback into a reward which is used to update a Q-table: a table with a Q-values (the expected discounted reward) assigned to every state-action pair and used to select the next action. Three additions to the standard algorithm have been proposed and implemented by Thomaz and Breazeal and are used here as well: guidance, communication by the robot and an undo option (Thomaz & Breazeal, 2008). Teachers have two way to transmit information to the robot:

the reward channel (to give the numerical reward on the last action) and the guidance channel (to direct attention toward parts of the state and restrict exploration).

The guidance emerged from the results of a pilot study where participants assigned rewards to objects to indicate that the robot should do something with these objects. With the guidance, teachers can direct the attention of the robot toward certain item in the environment to indicate the robot that it should interact with them. This guidance behaviour offer partial control over the robot's actions and restricts the next action the robot is executing, but it cannot be used to set explicitly the robot behaviour.

The robot also communicates its uncertainty by directing its gaze toward different items in the environment with equally high probability of being used next. The aim of this communication of uncertainty is to provide transparency about the robot's internal state, for example indicating when a guidance should be provided.

Finally, after a negative reward, the robot tries to cancel the effect of the previous action (if possible), resulting in a undo behaviour. As shown in the original paper, these three additions improve the performance on the task.

5.2.2 SPARC

SPARC uses a single type of input similar to the guidance present in IRL but without using the reward channel. However with SPARC, the guidance channel controls directly the actions of the robot. The robot communicates every of its intentions (i.e the action it plans to execute next) to its teacher by looking to a part of the environment and the teacher can either not intervene, letting the robot execute the suggested action or step in and force the robot to execute an alternative action. This combination of suggestions and corrections gives the teacher full control over the actions executed by the robot. This also makes the rewards redundant: rather than requiring the human to explicitly provide rewards, a positive reward is directly assigned to each action executed by the robot as it has been either forced or passively approved by the teacher.

5.2.3 Differences of approaches

Unlike IRL, SPARC offers full control over the actions executed by the robot. SPARC changes the learning paradigm from learning from the human evaluation of actions' effects to learning from the human's expectation of these actions' effects and human's preferences. An expert in the task domain evaluates the appropriateness of actions before their execution and can guide the robot to act as they see fit. This implies that the robot does rely on observing negative effects of an action to learn that this action should be avoided, but rather what the best action is for each state. Even in a non-deterministic environment such as human-robot interactions, some actions can be expected to have a negative consequence and the human teacher can stop the robot from ever executing these actions, preventing the robot from causing harm to itself or its social or physical environment.

Another noticeable difference is the way in which the robot communicates with the user: in IRL, the robot communicates its uncertainty about an action and with SPARC its intention to execute an action. It should also be noted that the quantity of information provided by the user to the robot is similar for both IRL and SPARC: in SPARC the user can offer the whole action space as commands to the robot, which removes the need for explicit rewards. While in IRL, the teacher can guide the robot toward a subset of the action space but has to manually provide feedbacks to evaluate the robot's decisions.

5.2.4 Hypotheses

Three hypotheses have been tested in the study:

H1 *Effectiveness and efficiency with non-experts.* Compared to IRL, SPARC leads to higher performance, whilst being faster, requiring fewer inputs and less mental effort from the teacher and minimising the number of errors during the teaching phase when used by non-experts.

H2 *Safety with experts.* SPARC can be used by experts to teach an action policy safely, quickly and efficiently, unlike other IML methods lacking control.

H3 *Control.* Teachers prefer a method in which they can have more control over the robot's actions.

5.3 Methodology

The task used in this study is the same as Thomaz & Breazeal (2008): Sophie's kitchen, a simulated environment on a computer where a virtual robot has to learn how to bake a cake in a kitchen. As the source code was not available, the task was reimplemented to stay as close as possible to the description in the paper and the online version of the task[2].

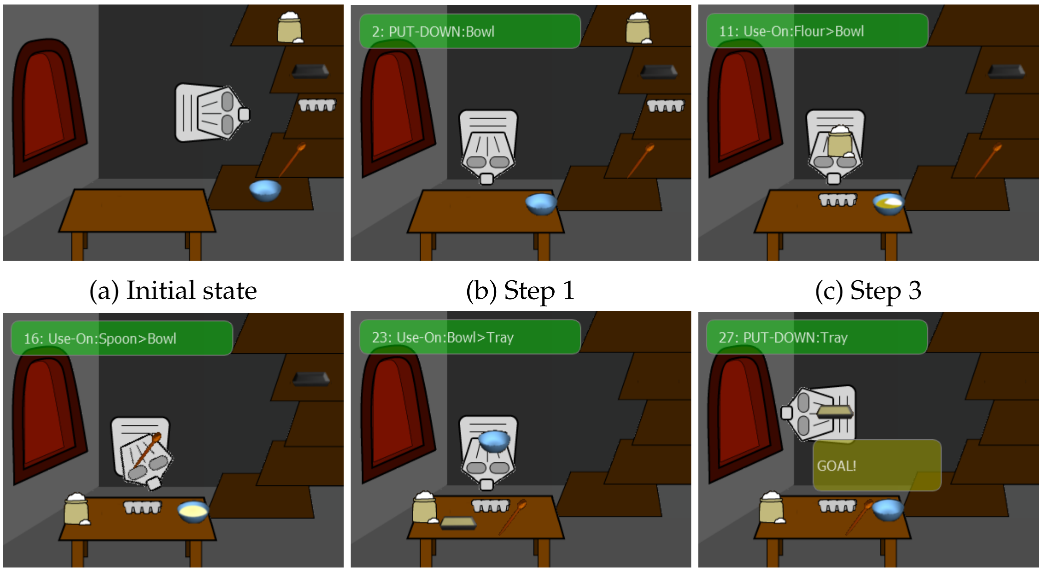
The scenario is the following: a robot, Sophie, is in a kitchen with three different locations (shelf, table and oven) and five objects (flour, tray, eggs, spoon and bowl) as shown in Figure 5.1a. Sophie has to learn how to bake a cake and the participant has to guide the robot through a sequence of steps while giving enough feedback so the robot learns a correct series of actions. As presented in Figure 5.1, there are six crucial steps to achieve a successful result:

1. Put the bowl on the table (Figure 5.1b).
2. Add one ingredient to the bowl (flour or eggs).
3. Add the second ingredient (Figure 5.1c).
4. Mix the ingredients with the spoon to obtain batter (Figure 5.1d).
5. Pour the batter in the tray (Figure 5.1e).
6. Put the tray in the oven (Figure 5.1f).

The environment is a deterministic Markov Decision Process, defined by a state, a set of actions (move left, move right, pick up, drop and use), a deterministic transition function and environmental reward function (+1 for success and -1 for failure and -0.04 for every other step to penalise long sequences). Different action policies can lead to success, but many actions end in a failure state, for example putting the spoon in the oven results in a failure. As argued by Thomaz and Breazeal, this environment provides a good setup to evaluate teaching methods to a robot due to the large number of possible states (more than 10,000), the presence of success and failure states and the sparse nature of the environmental reward function which increases the need for a teacher to aid the learning. More details on the environment are available in the original paper.

[2] `http://www.cc.gatech.edu/~athomaz/sophie/WebsiteDeployment/`

5.3.1 Task



(a) Initial state (b) Step 1 (c) Step 3

(d) Step 4 (e) Step 5 (f) Step 6

Figure 5.1: Presentation of different steps in the environment. 5.1a initial state, 5.1b step 1: bowl on the table, 5.1c step 3: both ingredients in the bowl, 5.1d step 4: ingredients mixed to obtain batter, 5.1e step 5: batter poured in the tray and 5.1f step 6 (success): tray with batter put in the oven. (Step 2: one ingredient in the bowl has been omitted for clarity)

5.3.2 Implementation

Two conditions are compared for this study: IRL and SPARC. The underlying learning is strictly identical for both conditions, only the way to interact with it (inputs to and from the algorithm) and how to provide rewards changes: with IRL teachers have to explicitly provide rewards, while they are implicit with SPARC.

The learning algorithm (see Algorithms 1 and 2) is a variation on Q-learning, without reward propagating[3]. This guarantees that any learning by the robot is due to the teaching by the human, and as such provides a lower bound for the robot's performance. By using Q-learning, the performance of the robot would be higher.

[3] In Q learning the update function is $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \gamma(\max_a Q(s_{t+1}, a)) - Q(s_t, a_t))$

Algorithm 1: Algorithm used for SPARC

```
while learning do
    $a$ = action with the highest $Q[s, a]$
    look at object or location used with $a$
    while waiting for correction (2 seconds) do
        if received command then
            $a$ = received command
            reward, $r = 0.5$
        else
            reward, $r = 0.25$
    execute $a$, and transition to $s'$
    $r = r + r_{environment}$
    Learn:
    $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \gamma(\max_a Q(s_t, a)) - Q(s_t, a_t))$
```

Algorithm 2: Algorithm used for IRL

```
while learning do
    $a$ = action with the highest $Q[s, a]$
    indicate confusion if multiple $a$ with similar high $Q[s, a]$
    while waiting for guidance and reward on last action (2 seconds) do
        if received command then
            Try: $a$ = received command
        if received reward $r'$ then
            $r = r + r'$
    Learn:
    $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \gamma(\max_a Q(s_t, a)) - Q(s_t, a_t))$
    execute $a$, and transition to $s'$
    $r = r_{environment}$
```

As shown in the algorithms, another minor difference between the conditions is that with SPARC, the algorithm learns just after executing an action (and only with positive rewards), while IRL learns on the last action before executing the next one based on the human evaluation of the last action.

Interactive Reinforcement Learning

We have implemented IRL following the principles presented in Thomaz & Breazeal (2008). The user can use the left click to display a slider providing rewards. The guidance is implemented by right-clicking on objects: it directs the robot's attention to the object if facing it (a click on objects in different locations has no effect). Following the guidance, the robot will execute the candidate action involving the object. The action space is not entirely covered by this guidance mechanism: for example, it does not cover moving from a location to another. This guidance if used correctly, limits the exploration for the current step to the part of the environment evaluated as more interesting by the user without preventing the robot to explore in further steps. The robot communicates its uncertainty by looking at multiple objects having similarly high probability of being used.

Some modifications were required to the original study due to the lack of implementation details in the original paper, one of them being the use of a purely greedy action selection instead of using softmax, due to the absence of parameters descriptions. The reliance on human rewards and guidance limits the importance of autonomous exploration, and thus, the greediness of the algorithm should assist the learning by preventing the robot to explore outside of the guided policy. Additionally, as the environment is deterministic and the algorithm is greedy, the concept of convergence is altered: once a trajectory has Q-Values high enough or a correct gradient of Q-Values on all state-action pairs, it will be reinforced automatically. But the teacher can manually force the robot to converge or diverge depending of their behaviours.

SPARC

SPARC uses the gaze of the robot toward objects or locations to indicate which action the robot is suggesting to the teacher. Similarly to the guidance in IRL, the teacher can use the right click of the mouse on objects to have the robot execute the action associated to this object in the current state and this has been extended to also cover locations. With SPARC, the command covers all the action space: at every time step, the teacher can specify, if desired, the next action executed by the robot. If an action is not corrected, a positive reward of 0.25 is automatically received (as it has the implicit approval from the teacher) and if the teacher selects another action, a reward of 0.5 is given to the correcting action (the corrected action is not rewarded). That way, actions actively selected are more reinforced and participants can still have give higher rewards when using IRL. This system allows for the use of reinforcement learning with implicit reward assignation, aiming to simplify the teaching interaction.

5.3.3 Interaction protocol

Participants are divided into two groups and interact first either with IRL or SPARC as shown in Figure 5.2. Before interacting, participants complete a demographic questionnaire and receive an information sheet explaining the task (describing the environment and how to bake the cake) and one explaining the system they are interacting with. Then they interact for three sessions with the assigned system. Each session is composed of a

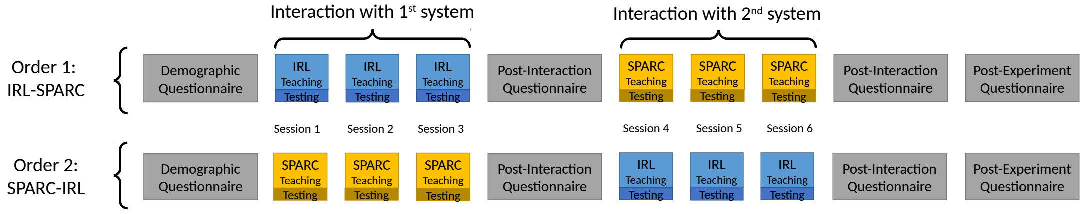

Interaction with 1st system
Interaction with 2nd system
Order 1:
IRL-SPARC
Order 2:
SPARC-IRL
Demographic Questionnaire
IRL Teaching Testing
SPARC Teaching Testing
Post-Interaction Questionnaire
Post-Experiment Questionnaire
Session 1
Session 2
Session 3
Session 4
Session 5
Session 6

Figure 5.2: Participants are divided into two groups. They first complete a demographic questionnaire, then interact for three independent sessions (with a teaching and a testing phase each) with a system (IRL or SPARC). After a first post-interaction questionnaire, participants interact for another three sessions with the other system before completing the second post-interaction questionnaire and a final post-experiment questionnaire.

teaching phase and a testing phase. The teaching phase is composed of as many teaching episodes as the participant desires, a teaching episode ends when a success or failure state has been reached which returns the environment to the initial state. In the same way as in the initial experiment by Thomaz and Breazeal, participants can decide to terminate the teaching phase whenever they desire by clicking on a button labelled 'Sophie is ready', however the session also stops after 25 minutes to impose an upper time limit to the study. After the end of a teaching phase, the robot runs a testing phase where the participant's inputs are disabled and which stops as soon as an ending state is reached or the participants decide to stop it (for example if the robot is stuck in a loop). This testing phase is used to evaluate the performance of the participants for this session. The interaction with a system consists of three repeated independent sessions with their own independent teaching and testing phases to observe how the interactions evolve as participants are getting used to the system.

After participants completed their three sessions with the first system, they are asked to interact for three more sessions with the other system. This way, every participant interacts three times with each system (IRL and SPARC) and the order of interaction is balanced. Additionally, participants have to complete post-interaction questionnaires distributed after the interaction with the first system and the second one and a final post-experiment questionnaire at the end of the experiment. All information sheets and questionnaires can be found online [4].

[4] https://emmanuel-senft.github.io/experiment2.html

5.3.4 Participants

A total of 40 participants have been recruited using a tool provided by the university to reach a mixed population of students and non-student members of the local community. All participants gave written informed consent, and were told of the option to withdraw at any point. All participants received remuneration at the standard U.K. living wage rate, pro rata. Participants were distributed randomly between the groups whilst balancing gender and age (age *M*=25.6, *SD*=10.09; 24F/16M). Participants were mostly not knowledgeable in machine learning and robotics (average familiarity with machine learning *M*=1.8, *SD*=1.14; familiarity with social robots *M*=1.45, *SD*=0.75 - Likert scale ranging from 1 to 5).

In addition to naive non-expert users, an expert user (the author) interacted five times with each system following a strictly optimal strategy in both cases. These results from the expert are used to evaluate H2 and show the optimal characteristics of each system (IRL and SPARC) when used by trained experts such as therapist in a context of assistive robotics.

5.3.5 Metrics

Interaction Metrics

We collected four metrics during the teaching phase: the teaching performance (how many steps participants reached in the teaching phase), the teaching time (from 0 to 25 minutes), the number of times a participant reached a failure state while teaching (which can be related to the risks taken during the teaching) and the number of inputs provided during the teaching, which can be seen as the efforts invested in the teaching. The testing phase is a single run of the taught action policy ending as soon as the robot reaches an ending state (failure or success) or if stopped by the participants. We use the performance achieved during this single test as evaluation of the success of the teaching. As not all participants reached a success during the testing phase, we used the six key steps defined in Section 5.3.1 as a way to evaluate the performance ranging from 0 (no step has been completed) to 6 (the task was successfully completed) during this testing run: for example a testing where the robot puts both ingredients in the bowl but reaches a failure state before mixing them would have a performance of 3.

Questionnaires

The post-interaction and post-experiment questionnaires provide additional subjective information to compare with the objective results from the interaction data. Two principal metrics are gathered: the workload on participants and the perception of the robot.

Workload is an important factor when teaching robots. As roboticists, our task is to make the teaching of robots as undemanding as possible, meaning that the workload for user should be minimal. Multiple definitions for workload exist and various measures can be found in the literature. Due to its widespread use in human factors research and clear definition and evaluation criteria, we used the NASA-Task Load Index (TLX) (Hart & Staveland, 1988). We averaged the values from the 6 scales (mental, physical and temporal demand, performance, effort and frustration) to obtain a single workload value per participant for each interaction. There are two measures for each participant, after the interaction with the first system and after the interaction using the other one.

Finally, the perception of the robot has been evaluated in the post-interaction and post-experiment questionnaires using subjective questions (measured on a Likert scale), binary questions (which robot did you prefer interacting with) and open questions on preference and naturalness of the interaction.

5.4 Results

Most of the results are non-normally distributed. Both ceiling and floor effects can be observed depending on the conditions and the metrics. For the teaching time, some participants preferred to interact much longer than others, resulting in skewed data. Likewise for the performance: often participants either reached a successful end state or did not hit any of the sub-goals of the task in the testing phase ending often in two clusters of participants: one at a performance of 6 and one at 0. Similarly, some participants who interacted a long time with the system did not complete any step, while others could achieve good results in a limited time. Due to the data being not normally distributed, Bayesian statistics have been used from the JASP software (JASP Team, 2018). Each test: mixed Anova for omnibus comparison between condition for each interaction (first or second), independent t-test for post-hoc comparison between participants and paired samples t-test for post-hoc comparison within participants have been performed using their Bayesian counterpart, which also remove the need of doing a correction on post-hoc test such as Bonferroni. As such, no p-value is reported, but a B factor representing how much a parameter impact of a parameter on the metric (if $B < 1/3$ there is no impact, if $B > 3$ the impact is strong) (Dienes, 2011; Jeffreys, 1998).

Initial results of the first interaction of the participants have been reported in Senft et al. (2016).

5.4.1 Interaction data

Five objective metrics (teaching performance, testing performance, teaching time, number of inputs provided and number of failures) and one subjective metric (workload) have been used to evaluate the efficiency of IRL and SPARC.

Teaching Performance

Figure 5.3 presents the performance of participants during the teaching phase, i.e how far in the steps they brought the robot during the teaching. It can be seen as how much control a method provide to the teacher and how easy it is to guide the robot to execute a desired action policy. It is also an upper bound for the testing performance as, due to the risk of failures or loop in the environment, the performance in the testing phase cannot (or has dramatically low probability to) achieve a higher performance than in the teaching phase.

In the first three sessions participants interacted with either IRL or SPARC and swapped for the remaining three sessions. The bayesian analysis shows the importance of the interacting condition on the teaching performance on the three sessions both for the first interaction and the second one ($B_1 = 2881$ and $B_2 = 76.2$). According to the medians and the graph, participants using SPARC achieved a higher teaching performance than the ones using IRL. The session number has no impact on the performance ($B_1 = 0.089$ and $B_2 = 0.105$). Table 5.1 presents descriptive statistics of the performance in the different conditions.

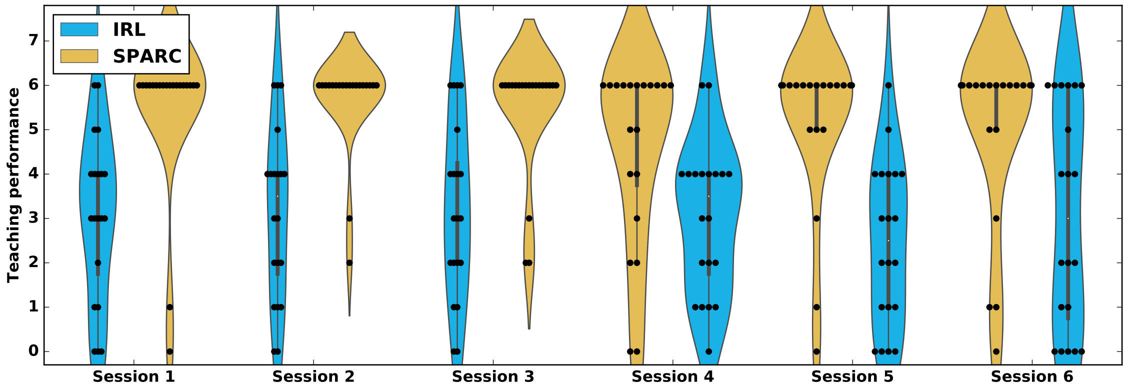


Figure 5.3: Comparison of the teaching performance for the six sessions (three with each system, IRL and SPARC, with interaction order balanced between groups). A 6 in teaching performance shows that the participant reached at least one success in the teaching phase.

Table 5.1: Medians of the performance in the teaching phase. Lines represent the condition in which participant interacted in a the first three sessions or the last three. It must be noted that between session 3 and 4 participants change condition.

	$\widetilde{X}_1$	$\widetilde{X}_2$	$\widetilde{X}_3$	$\widetilde{X}_4$	$\widetilde{X}_5$	$\widetilde{X}_6$
IRL	3.0	3.5	3.0	3.5	2.5	3.0
SPARC	6.0	6.0	6.0	6.0	6.0	6.0

Testing Performance

Figure 5.4 presents the performance of participants during the testing phase, how successful was the teaching. The bayesian analysis shows important factor of condition on the performance on the three sessions both for the first interaction and the second one ($B_1 = 8.8\text{x}10^5$ and $B_2 = 7340$). The descriptive statistics show that participants using SPARC reached a higher performance than the ones using IRL. The session number has no impact on the performance on the first interaction, but results are inconclusive for the impact of repetition on the second interaction ($B_1 = 0.084$ and $B_2 = 0.80$). Table 5.2 presents descriptive statistics of the performance in the different conditions.

As shown by table 5.2, in our study, only a limited number of participants succeeded in teaching the robot to complete the task using IRL, this observation will be discussed in more details in section 5.6.

Table 5.2: Medians of the performance in the testing phase.

	$\widetilde{X}_1$	$\widetilde{X}_2$	$\widetilde{X}_3$	$\widetilde{X}_4$	$\widetilde{X}_5$	$\widetilde{X}_6$
IRL	0.0	0.0	1.0	0.0	0.0	0.0
SPARC	6.0	6.0	6.0	4.5	6.0	6.0

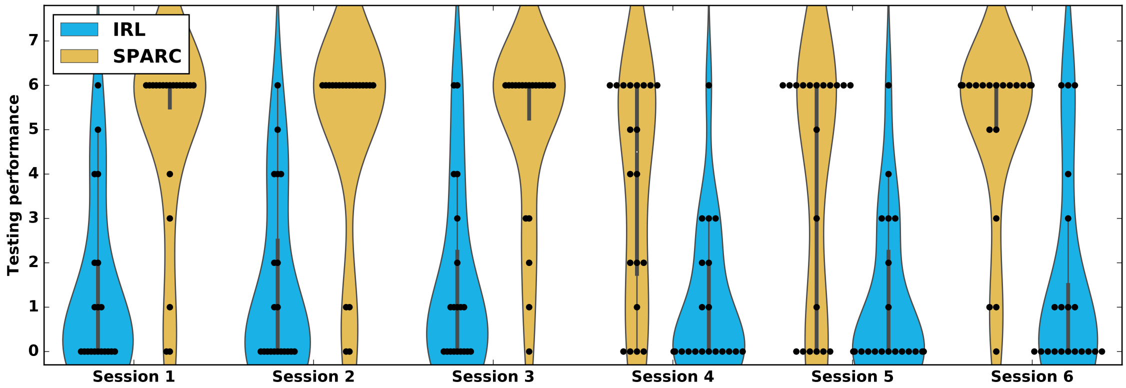


Figure 5.4: Comparison of the testing performance for the six sessions. A 6 in performance shows that the taught policy led to a success.

Teaching time

Figure 5.5 presents the time participants spent teaching. They could stop whenever they decided (if they think the robot masters the task or cannot learn further) or the session would stop after 25 minutes. The bayesian analysis shows the important role of condition ($B_1 = 31.4$ and $B_2 = 679$) and sessions on the time spend teaching ($B_1 = 8.3\text{x}10^9$ and $B_2 = 3188$). Additional post-hoc comparisons for sessions indicate that in the first interaction, the teaching time decreases between the first and the second session and then tends to stabilise between the second and the third sessions ($B_{12} = 4.4\text{x}10^5$, $B_{13} = 2.6\text{x}10^6$ and $B_{23} = 0.435$). Similar results happen in the second interaction ($B_{45} = 850$, $B_{46} = 382$ and $B_{56} = 0.172$) with more strengths on the stabilisation of teaching time between session 5 and 6.

Table 5.3 presents descriptive statistics of the teaching time in the different conditions.

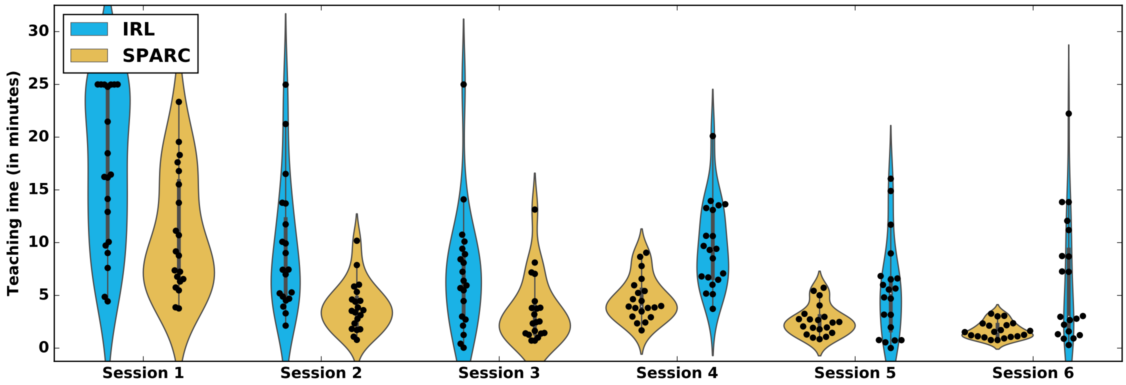


Figure 5.5: Comparison of the teaching time for the six sessions. At 25 minutes, the session stopped regardless of the participant stage in the teaching.

Table 5.3: Medians of the teaching time in each session (in minutes).

	$\widetilde{X}_1$	$\widetilde{X}_2$	$\widetilde{X}_3$	$\widetilde{X}_4$	$\widetilde{X}_5$	$\widetilde{X}_6$
IRL	16.34	7.43	6.16	9.36	5.18	3.0
SPARC	8.97	3.56	2.49	3.96	2.45	1.53

Inputs

Figure 5.6 presents the number of inputs the participants provided while teaching. The bayesian analysis shows that in both interaction, the condition impacts the number of inputs provided ($B_1 = 27.4$ and $B_2 = 34.1$), while sessions matter highly in the first interaction, but the results are inconclusive for the second interaction ($B_1 = 4.1x10^5$ and $B_2 = 1.5$). Additional post-hoc comparisons for sessions indicate that in the first interaction, the number of inputs used decreases between the first and the second session and then tends to stabilise between the second and the third one($B_{12} = 2707$, $B_{13} = 4.7x10^4$ and $B_{23} = 0.410$). However, for the second interaction, no strong difference is observed between session 1 and 2 and session 1 and 3 while the number of inputs stabilise between session 5 and 6 ($B_{45} = 2.6$, $B_{46} = 2.7$ and $B_{56} = 0.17$).

Table 5.4 presents descriptive statistics of the number of inputs used in the different conditions.

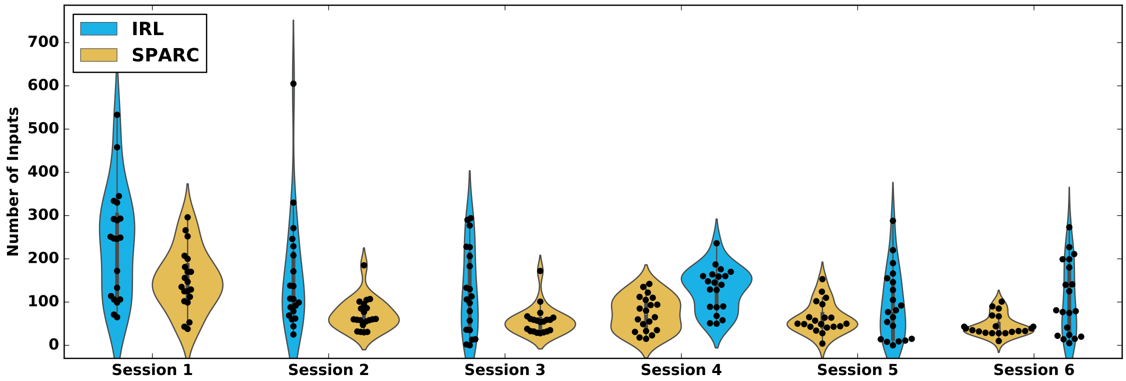


Figure 5.6: Comparison of the number of inputs provided by the participants for the six sessions.

Table 5.4: Medians of the number of inputs in the testing phase.

	$\widetilde{X}_1$	$\widetilde{X}_2$	$\widetilde{X}_3$	$\widetilde{X}_4$	$\widetilde{X}_5$	$\widetilde{X}_6$
IRL	248.0	107.5	109.5	142.5	79.0	80.0
SPARC	141.0	60.0	56.0	72.5	50.0	37.0

Number of failures

Figure 5.7 presents the number of failures participants faced during the teaching phase. The bayesian analysis shows that for both interactions, both the condition ($B_1 = 6.2x10^4$ and $B_2 = 2.6x10^4$) and sessions ($B_1 = 1.5x10^4$ and $B_2 = 11$) have an important role on the number of failures. Additional post-hoc comparisons for sessions indicate that in the first interaction, the number of failures decreases between the first and the second session and then stabilises between the second and the third one($B_{12} = 619$, $B_{13} = 1.7x10^3$ and $B_{23} = 0.25$). Similar results can be observed in the second interaction ($B_{45} = 3.3$, $B_{46} = 7.5$ and $B_{56} = 0.2$).

Table 5.5 presents descriptive statistics of the performance in the different conditions.

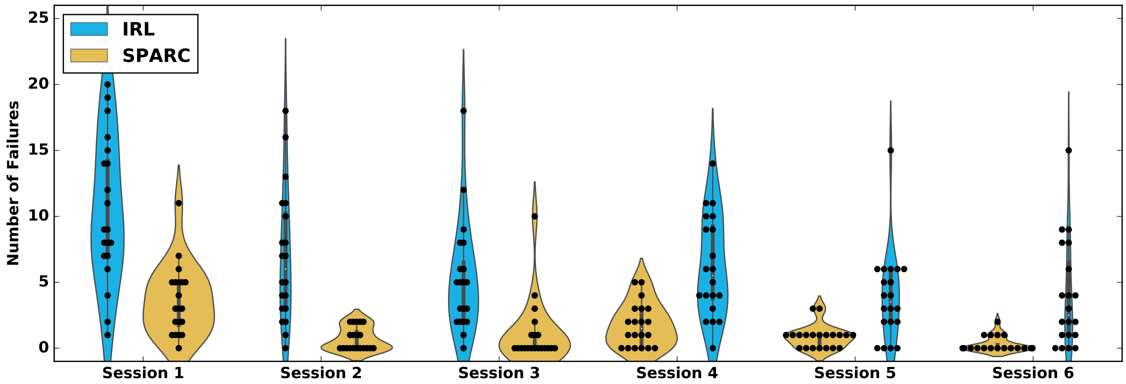


Figure 5.7: Comparison of the number of failures for the six sessions. A 6 in performance shows that the taught policy leads to a success.

Table 5.5: Medians of the number of failures in the testing phase.

	$\widetilde{X}_1$	$\widetilde{X}_2$	$\widetilde{X}_3$	$\widetilde{X}_4$	$\widetilde{X}_5$	$\widetilde{X}_6$
IRL	9.0	6.0	5.0	5.5	3.5	2.5
SPARC	3.0	0.0	0.0	1.5	1.0	0.0

5.4.2 Questionnaire data

The main task of the questionnaires was to assess the workload on participants when interacting with a condition. Figure 5.8 presents the workload for participants for each condition for both interactions. Bayesian analysis show a strong effect on the condition for both interactions ($B_1 = 462$ and $B_2 = 8.1x10^4$) between participants and for both interactions order within participants($B_{SPARC-IRL} = 1.1x10^4$ and $B_{IRL-SPARC} = 1.7x10^6$). Regardless of the comparison criteria, when interacting with SPARC, participants reported a lower workload than when interacting with IRL. In the first interaction, participants using IRL reported a workload of 12.9 (SD = 2.33), whereas the ones using SPARC reported 8.94 (SD = 3.01). In the second interaction, participants interacting with SPARC rated their workload as 7.44 (SD = 3.41) and the ones using IRL reported 13.87 (SD = 2.84).

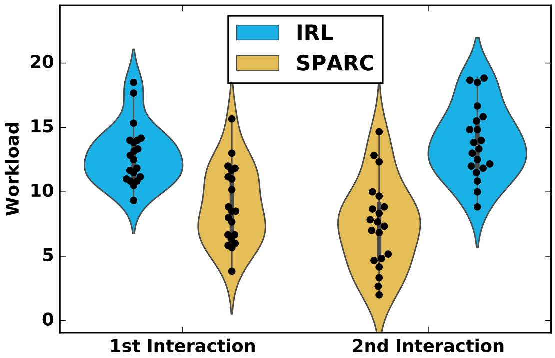


Figure 5.8: Workload as measured by the NASA-TLX for both conditions and both interaction order.

5.4.3 Expert

To evaluate the best case potential offered by SPARC and IRL, an expert (the author) interacted five times with each systems. In both cases, the expert followed a strictly optimal strategy. This shows the expected behaviours in optimal conditions, the best metrics achievable. Results of the interactions are presented in Table 5.6. In both cases, the expert successfully taught the robot (as indicated by a performance of 6), which indicates that both systems can be used to teach a robot an action policy. However the time required to teach the robot with IRL is higher than with SPARC (B = 7102).

Additionally, when using IRL, even an expert cannot prevent the robot from reaching failure states during the teaching due to the lack of control over the robot's action. This is prevented when interacting with SPARC, due to the full control and clear communication, the teacher can ensure that only desired actions are executed. So with sufficient knowledge, an expert can teach the robot to behave safely without having to explore and reach undesired states. This has real world applications, as random exploration is often impossible or undesirable, SPARC offers a way for the teacher to stop the robot from executing actions with negative consequences.

Similar results have been observed with the non-expert participants: in their last interaction with SPARC, both groups had a median of 0 failures for a performance of 6, meaning that more than half of the participants successfully taught the robot the task without ever hitting a failure state after understanding how SPARC can be used.

Table 5.6: Results of an expert interacting 5 times with each system following an optimal strategy. In case without variance, Bayes Factor cannot be computed.

	IRL *M(SD)*	SPARC *M(SD)*	B factor
Performance	6 (0)	6 (0)	NA
Time (minutes)	4.5 (0.67)	0.60 (0.03)	7102
Inputs	115.6 (8.4)	28 (0)	NA
Number of Failures	3.2 (0.84)	0 (0)	NA

5.5 Validation of the hypotheses

5.5.1 Effectiveness and efficiency with non-experts

The objective data (teaching and testing performance, teaching time, number of inputs and number of failures) show that despite spending a shorter time interacting with SPARC and using less inputs, participants reached a higher performance than with IRL whilst facing fewer failures during the teaching. Additionally, when interacting with SPARC, participants' time required to teach the robot decreased with successive sessions, without affecting the performance. This indicates that after the first session, participants understood the interaction mechanism SPARC and consistently managed to achieve a high performance whilst requiring less time to teach the robot the task. On the other hand, when interacting with IRL, participants' performance remains low over the session, and their teaching time decreases between session 1 and 2 but not between session 2 and 3. This might be due to a loss of motivation after session 1 where often participants did not succeed to teach the robot, reducing the desire to further interact in successive sessions.

The results suggest that teaching the robot using SPARC allows the robot to achieve a higher performance than with IRL, in a shorter time, while requiring fewer inputs and making fewer errors when teaching. These objective results are also supported by subjective measures: the workload on the teacher is lower when using SPARC than when using IRL. For these reasons, H1 ('Compared to IRL, SPARC can lead to higher performance, whilst being faster, requiring fewer inputs and less mental effort from the teacher and minimising the number of errors during the teaching when used by non-experts.') is supported.

5.5.2 Safety with experts

As presented in Section 5.4.3, when interacting with SPARC, experts can reach a success easily and safely (low number of inputs, quickly and without facing failures), and this effect is also observed after some training for the naive participants: most of them reached a success without encountering any failures in their last interaction with SPARC.

However, when interacting with IRL, even experts cannot prevent the robot to end in failures states even when applying a strictly optimal action policy. This effect is typical of feedback-based IML methods: as the teacher only rates the actions of the agent, they cannot prevent them to make errors, they can just reward negatively these errors.

This shows support for H2 ('SPARC can be used by experts to teach an action policy safely, quickly and efficiently, unlike other IML methods lacking control'). This also demonstrates how the principles presented in Chapter 3 can provide control to the teacher over the robot's actions and so improve the teaching and ensure that even in the early stages of teaching, the action policy of the robot can be appropriate, which is not the case of most other IML methods.

5.5.3 Control

One of the main differences between the two methods is the way in which the concept of teaching is approached. With IRL an exploratory individual learning approach is followed: the robot has freedom to explore, and it can receive feedback on its actions and hints about actions to pursue next from a teacher. This is to some extent inspired by how children are taught, where the learning process can be more important than the achieved results. This is supported by the behaviours observed by Thomaz and Breazeal: their participants gave motivational rewards to the robot, just as one would to do to keep children motivated during learning, despite the absence of effect or use in classical reinforcement learning.

On the other hand, SPARC promotes a more direct teaching process: the supervisor explicitly tells the robot what to do and expects it to obey and learn. The robot is not totally considered as a social agent from the supervisor's point of view, but rather as a tool having to learn an action policy. This does not mean that the robot cannot be social: the supervisor can teach in such a non-social way to the robot how to interact socially. This approach is more task oriented, and we argue that it better fits assistive robotics when the interaction with the teacher does not have to be social, and the task (such as interaction with a child with ASD) is more important than the social relation between the robot and its supervisor (a therapist for example).

The post-experiment questionnaire included the open question: 'which robot did you prefer interacting with and why?'. Almost all the participants (38 out of 40) replied that they preferred interacting with SPARC. Half of all the participants used vocabulary related to the control over the robot actions ('control', 'instruction', 'command', 'what to do' or 'what I want') to justify their preferences without these words being used in the question. Furthermore, multiple participants reported being frustrated to have only partial control over the robot's actions with IRL, they would have preferred being able to control each action of the robot.

To the question 'which interaction was more natural?', 10 participants rated IRL as being more natural, using justifications such as: 'The robots thinks for itself', 'Some confusion in the [IRL] robot was obvious making it more natural', 'More like real learning', 'Because it was hard to control the robot' or 'People learn from their mistakes faster'. But despite some participants acknowledging that IRL is more natural, closer to human teaching, they still preferred teaching using SPARC. This suggests that when humans teach robots, they are focused on the results of the teaching: can the robot do the new task requested. This relates to the role of robots, they often interact in human-centred scenario where they have to complete a task for their users. And due to the absence of life-long learning for robots today, it is not worth investing time and energy to allow the robot to improve its learning process or explore on its own. These comments from the participants show support for H3 ('Teachers prefer a method providing more control over the robot's actions.').

5.6 Discussion

Despite not being originally designed to be used in combination with Reinforcement Learning, SPARC achieved good results in this study. This shows that principles presented in Chapter 3 are agnostic to the learning algorithm and promote efficient teaching. Furthermore, SPARC achieves a higher performance, in a shorter time and facing less failures than IRL, whilst requiring a lower workload from the human teacher. And finally, when used by experts, SPARC demonstrates that teaching can be safe and quick: the full control over robot's action in the teacher's hands ensures that only desired actions will be executed. These results show an important feature of teaching robots: as robots interact in task oriented, human-centred environments, human teachers seem to prefer direct approaches focused on commands rather than letting the robot explore on its own.

5.6.1 Comparison with original Interactive Reinforcement Learning study

Unlike the original experiments evaluating IRL (Thomaz & Breazeal, 2008), in this study, most of the participants did not succeed in teaching the robot the full cake baking sequence using feedback and guidance (the IRL condition). In Thomaz and Breazeal's study, the participants were knowledgeable in machine learning (M=3.7, SD=2.3 - range: 1 to 7), but the population in the presented study was drawn from a more general public having little to no knowledge of machine learning (M=1.8, SD=1.13 - range: 1 to 5). This can explain why a much larger number of participants did not achieve success with IRL in this study whereas Thomaz and Breazeal only reported 1 participant out of 13 failing the task. In our study, only 12.5% of the participants and the expert did manage to teach the robot using IRL.

As demonstrated by the teaching performance, most of the participants did not manage to reach a single success during the teaching phase. We identify the lack on control over the robot's actions as limiting factor for the teaching, as participants did not manage to steer the robot to do correct actions, they could not reward it and teach it an action policy. Additionally, the requirement of explicit feedback made the learning task more complex. Future robot teachers need to have control over the robot's action and robots should also use implicit rewarding to ease the task for the teacher. This is consistent with Kaochar et al. (2011) who note that feedback is not well suited for teaching an action policy from scratch, but better for fine tuning. For teaching the basis of the action policy, they recommend using demonstrations, a method much closer to SPARC.

5.6.2 Advantages and limitations of SPARC

In the SPARC implementation for this study, the algorithm mostly reproduces actions selected by the teacher. So one can argue that no learning algorithm is required, instead the actions could just be blindly reproduced by the robot. However, when combined with reinforcement learning, SPARC does provide advantages: due to the Q-Table, all the loops in the demonstration are removed in following interactions and the algorithm provides a way to deal with variations in teaching. It also allows the robot to continue from any state in the trajectory. And finally, due to the suggestion/correction mechanism, the teacher can leave the robot to act on its own as long as it attempts correct actions, and the human only has to intervene when the robot is about to execute an incorrect action.

However SPARC also has limitations in the current implementation, related to the quality of the human supervised guidance. If the teacher allows an action to be executed by mistake (through inattention or by not responding in time), this action will be reinforced and will have to be corrected later on. This might lead to loops when successive actions are cancelling each other (such as move left, then right). In that case, the teacher has to step in and manually guide the robot to break this cycle. Furthermore, due to the automatic execution of actions, the teacher has to be attentive at all times and ready to step in when a wrong action is suggested by the robot.

In this version, SPARC has been applied to a scenario where a clear strategy with optimal actions is present. The interaction also takes place in a virtual environment with a discrete time. Real human-robot interactions are stochastic, happen in real time and often there is no clear strategy known in advance. However, we argue that human experts in the application domain can know what type of actions should be executed when, and which features of the environment they used for their decision. As this knowledge can not be available to the robot's designers, robots should be able to learn from a domain user in an interactive fashion. In this chapter, SPARC mainly receives inputs from a teacher at predefined discrete times and still does not use the human knowledge to it's fullest: the learning algorithm is still simple and with limited inputs, but as presented in Chapter 6, SPARC has been tested in real-world interactions with humans.

5.6.3 Lessons learned on designing interactive machine learning for human-robot interactions

From observing the participants interacting with both systems, we derived four recommendations for future designs of interactive learning robot that we also used to develop the study presented in Chapter 6.

Clarity of the interface

Algorithms used in machine learning often need precisely specified inputs and outputs and require an internal representation of the world and policies. These variables are often not accessible to a non expert: the weights of a neural network or the values in a Q-table are not easily interpreted, if at all. The inner workings of the machine learning algorithms are opaque, and people only have access to inputs and outputs of the black box that is machine learning. As such, care needs to go into making the input and output intuitive and readable. For example, in this study (following Thomaz and Breazeal's original study), the communication between the robot and the teacher occurred through the environment: using clicks on objects rather than buttons on a graphical user interface. This design decision has important consequences as participants first have to familiarise themselves with the interface: how to interpret the robot's behaviour, what actions are available for each state and what is the exact impact of the actions? This lack of clarity leads to a high number of failures and high teaching time during the first session in our study. So we argue that to avoid this precarious discovery phase for the teachers, roboticists have to design interfaces taking into account results from the Human Factors community as advocated by Adams (2002) or find an intuitive way to train teachers to use these interfaces.

Limits of human adaptability

Human-Robot Interaction today is facilitated by relying on people adapting to the interaction, often making use of anthropomorphisation (Złotowski et al., 2015). Roboticists use people's imagination and creativity to fill the gaps in the robot's behaviour. However, human adaptivity has its own limits: in our study, often participants adopted one particular way of interacting with the system and they hold on to it for a large part of the interaction. For example, participants clicked on an object requiring two actions to interact with, assuming that the robot had planning capabilities which it did not. Or when the robot was blocked in some cycles (due to constant negative reward in IRL and undo behaviour or due to a loop created and not stopped with SPARC), participants kept on trying the same action to break the loop, without really exploring alternatives. For these reasons, if robots are to be used with a naive operator, they need a mechanism to detect these 'incorrect' uses and either adapt to these suboptimal human inputs or inform the user that this type of input is not supported and clarify what human behaviour is appropriate instead.

Importance of keeping the human in the learning loop

As argued in previous chapters, we think the presence of a human in the learning loop is key. This human in the loop can provide importance knowledge about the environment and allow the machine learning to deal with sensor errors or imperfect action policies. An expert supervising the robot should also be able to prevent the execution of specific actions or force the execution of others. This was one of the important points we considered when proposing SPARC: there is initially no distinction between a teaching and a testing phase, they are merged into a single phase. The teacher can correct the robot when needed and let it act when it behaves correctly. Participants used this feature of SPARC in this study: many participants corrected SPARC only when required rather than forcing every action, 37.5% of the participants even let the robot complete the task without giving a single command before starting the test to be sure that the robot is ready. In this study, SPARC has been used as a tool to provide online learning to a robot whilst keeping the teacher in control, but reducing the need of intervention over time.

Keeping people in control

Most of the scenario where a robot has to learn how to interact with humans are human-centred: the robot has to complete a task to help a human (such as in SAR). In these scenarios, the goal of the learning is to ensure that the robot can complete the task assigned to it, not to provide the robot with tools to learn more efficiently in further interactions. Similarly, participants in our study did not desire to have the robot exploring on its own and learn from its experience, they wanted to be able to direct the robot. Furthermore, a lack of control over the robot's actions can lead to frustration and loss of motivation for the teacher. This human control is especially critical when the robot is designed to interact with other people as undesired actions can have a dramatic impact, such as causing harm for the interaction partners or bystanders. For these reasons, we argue that when designing an interactively learning robot for HRI in human-centred scenario, it is critical to keep the human teacher in control.

However, this control does not mean that the robot cannot learn and become autonomous.

We take a stronger inspiration from LfD, using human input more efficiently to guide the learning, speeding it up and making it safer, especially in the early stages of the learning. The human is in control mainly when the robot is prone to make exploratory mistakes, and can prevent them before they occur, but once the action policy is appropriate enough, the teacher can leave the robot learn mostly on its own and refine its action policy with limited supervision from a human.

5.7 Summary

As presented in Chapter 3, SPARC has been designed to allow naive humans to teach an action policy to a robot while maintaining a constant appropriate behaviour. This chapter presents a study where SPARC was combined with RL to teach a simulated robot to complete a baking task. SPARC used communication of intentions from the robot, full control of the teacher over the robot behaviour and implicit rewarding mechanism to have participants teach the robot an action policy. This approach has been compared with IRL using communication of uncertainty, partial control and explicit rewarding to teach the robot in a study involving 40 participants. When interacting with SPARC, participants reached more successes, with less time and inputs and while facing fewer failures and a lighter workload than when interacting with IRL. From this user study involving 40 participants, SPARC has demonstrated being usable by naive participants to successfully teach an action policy and succeeded in its goal to enable humans to quickly and safely teach a robot to complete a task.

Based on these results and other observations, we also propose four guidelines to design interactive learning robots: (1) the interface to control the robot has to be intuitive, (2) the limits of human adaptability have to be taken into account (robots should detect deadlocks in human behaviours and adapt their way to be controlled or inform the human about it), (3) the operator should be kept in the learning loop and (4) teachers should stay in control of the robot behaviour when interacting in sensitive environment. The first two points can be seen to apply to all robot teaching methods, and should be addressed at the time of designing the interface. And by definition, SPARC aims to address these last two points: maintaining the performance of an adaptive system by remaining under progressively decreasing supervision.

Chapter 6

Teaching a robot to support child learning

Key points:

- An experiment was designed to test SPARC in the wild in a learning application in a school.
- Between participant study involving 70 children compared 3 conditions: passive robot, supervised robot and autonomous robot.
- .
- .
- .

Parts of the work presented in this chapter have been published verbatim in Senft et al. (2017b) [1]. The final publication is available from AAAI via `https://aaai.org/ocs/index.php/FSS/FSS17/paper/view/16011`.

[1] Note about technical contribution in this chapter: the author reimplemented every part of the system manually in Qt.

6.1 Motivation

Chapters 4 and 5 tested SPARC in interactions between robots or in a virtual world but not for human-robot interactions as it was developed for. As such a new study had to evaluate the application of SPARC to teach a robot an interactive behaviour in a real human-robot interaction. It has been decided to focus on robots in education to teach a food-web to children as it provides a constraints while rich and complex environment for the interaction. The scenario and the code is based on Lemaignan et al. (2017) but has been adapted to provide a robot controller and task goal for the children.

6.2 Setup of the study

Similarly to the study presented in Chapter 4, this study is based on the Sandtray paradigm (Baxter et al., 2012): a child interacts with a robot through a large touchscreen located between them. Additionally, a teacher can use a tablet to control the robot in one of the conditions (cf. Figures 6.1 and Figure 1.1 used to frame this research).

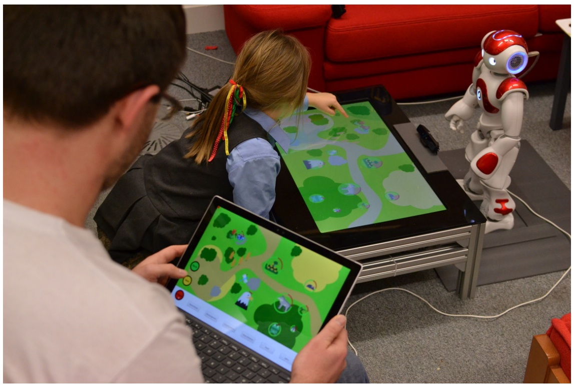


Figure 6.1: Setup used in the study: a child interacts with the robot tutor, with a large touchscreen sitting between them displaying the learning activity; a human teacher provides guidance to the robot through a tablet and monitors the robot learning.

6.2.1 Food chain game

To teach children a food web, they interacted with a game presented 10 animals and three types of plants. Animals have energy decreasing over time and they have to eat to stay healthy. Animals are immobile unless the child or the robot move them and eat or be eaten when entering in contact with another animal or a plant. Children have to feed animals by moving them to their food, and by feeding them can learn what food each animal eats.

Figure 6.2 presents an example of the game where after some time each animal has lost some energy

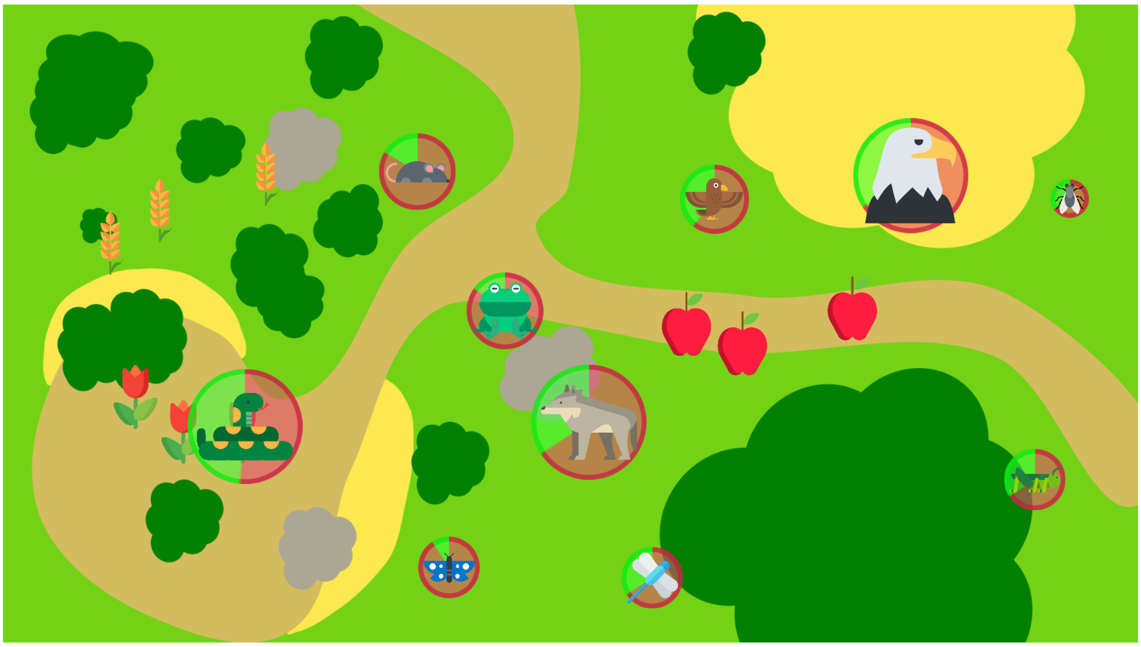


Figure 6.2: Example of the game. Animals have energy in red and have to eat plants of other animals to survive.

6.2.2 Robot behaviour

During the game, the robot can execute actions to provide hints and support to the child. The robot has access to five types of actions:

- Movements: moving any animal to, toward or away from other item (animal or plant) - the robot points to an animal and moves it on the game while describing its action (e.g. "The eagle needs help getting close to the mouse").
- Drawing attention: the robot points an item and says a reminder to the child (e.g. "Don't forget the frog").
- Reminding rules: the robot says one of 5 sentences on the game (e.g. "Move the animals to feed them" or "Don't feed animals with a lot of energy").
- Congratulation: the robot provides congratulations (e.g. "Well done").
- Encouragements: the robot provides encouragement (e.g. "You can do it").

For each utterance joining an action, multiple versions are available, and a random one not used recently is selected. Considering all the possible combination, the total number of actions adds up 655.

These actions represent different level of support, from general motivation and informations sentences to information about which animals the child should focus on or direct information about what animals eat. This should cover a large range of supportive feedback provided for such an application.

6.2.3 Wizard of Oz application

To allow the interaction between the teacher and the robot, a GUI has been developed representing the current state of the game exactly as the child sees it on the touchscreen 6.3. Buttons for the actions (excluding movements) allow the teacher to select which action the want the robot to execute. To provide additional features for the algorithms and precise which action the teacher is executed (on which animal should the robot draw the attention), the teacher can select animals or plant and provide them to the other components. For example, if the teacher highlights the frog and then press the "Draw attention" button, the actions *drawing attention to the frog* will be executed by the robot.

For the movements, the teacher can drag the the image of the animals, creating a *shadow* and the release of this shadow triggers the start of the motion. Depending the animal moved and the other items highlighted, the corresponding action will be inferred and sent to the robot. This gives access to the teacher to the full 655 actions without requiring as many buttons.

Additionally, the GUI is used by the teacher to respond to the propositions of the robot. Following the proposition of an action, a bubble describing the action will appear on top of the GUI and the corresponding item will be highlighted and if the action is a motion, an arrow will show the proposed motion. The teacher can react to the proposed action by pressing the "Do it", "Skip", "Cancel" or "Remove" buttons or let the action be executed. The action will be automatically executed after 2 seconds, during which the bubble will become greener to represent the passive acceptance of the action. The "Do it" button executes the action straight-away, the "Skip" button informs the algorithm that it should wait rather than doing the action, the "Cancel" button assigns a negative reward to this action in that case and finally, the "Remove" button looks for the closest previous instantiation of action in memory and removes it, preventing it to be executed later.

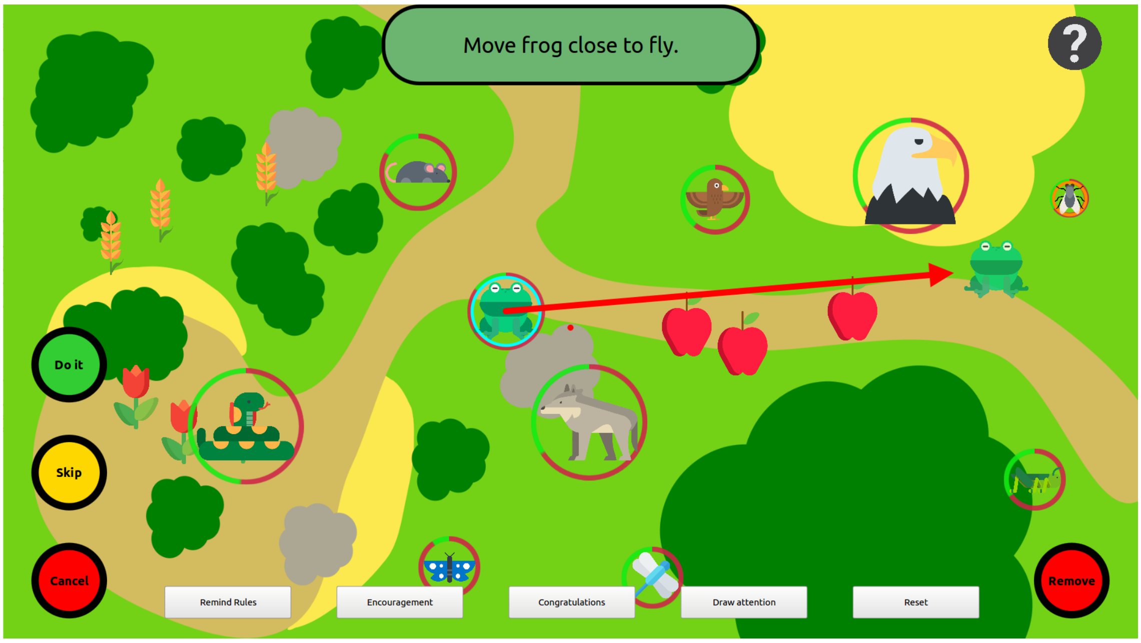


Figure 6.3: GUI used by the teacher to control the robot and respond to its suggestions. The game presents the same state as in Figure 6.2, and he robot proposes to move the frog close to the fly (text bubble, arrow, moving the *shadow* of the frog and highlight of the frog and the fly).

6.2.4 Algorithm

To learn a appropriate action policy, the algorithm has to map an action (or no action) to each possible state. The state used in this study represents the state of the game in a 210 dimensional vector, with value from 0 to 1. The dimensions include: distance between items, items' energy, time since events (child and robot touching each animal, robot's actions, interaction events: feeding an animal, death of animal...), progression in the game sessions and child face direction (toward the robot, the screen or away).

The actions and the state dimensions have been selected to be generic to many teaching task involving movable items: each item can have a value assigned to it (here energy, but this could be changed in other scenario), and some movable items (here animals) can be move toward or away from other items. Using these generic actions and this state definition, this implementation could be easily re-purposed to another teaching task.

The algorithm used for the learning is an adaptation of the one presented in Senft et al. (2017b). It is an instance based algorithm similar to the nearest-neighbours algorithm Cover & Hart (1967). However, two differences are notable compared to the initial algorithm. Firstly, instead of being defined on the full state space, instances are defined on a sliced version of the state. The intuition is that states needed to cover complex action policy require large number of dimensions, however for a single action, large parts of the state are irrelevant: for example if a robot needs to pick-up a cup, the colour of the cup does not impact the optimal motion. For this implementation, when selecting actions, the teacher can highlight features of the environment which will *activate* specific dimensions of the state space that are used to store the instance in memory. All *non-activated* dimensions are left as wild-card. Then when comparing the current state to the saved instances, the distance is only computed on the *activated* dimensions of the comparing instance. The second difference is that each instance saved has a reward assigned to it, if the teacher selected the action, a reward of $+1$ is assigned, and if the teacher cancelled the action (following an incorrect suggestion from the algorithm) a reward of -1 is assigned. When selecting an action, the algorithm looks through all the actions it have been using and for each action selects the closest instance and compute the expected reward as a multiplication of the distance with the reward assigned. Then the algorithm selects the action with the highest expected reward and proposes it if the value is higher than an adaptive threshold.

The algorithm runs at 2Hz while we would expect actions to be selected every 5 to 20 seconds, so unlike most of the discrete cases of action selections, in most of the steps, no actions are required. To handle this difference of timescale, a waiting action have been added (through the "skip button") and an adaptive threshold only proposes actions with an expected reward higher than the threshold. Selecting an action can reduce the threshold, and cancelling or skipping an action can increase it. This adapts the rate of action propositions to the desires of the teacher. Another mechanism filters propositions from the algorithm not to transfer them to the teacher when an action is already proposed to the supervisor or the robot is acting and also rewards negatively impossible actions (such as moving dead animals).

6.3 Methodology

6.3.1 Study design

Sixty children aged 8 to 10 participated in the study (age: *M*=8.9, *SD*=0.83; 11F/15M). Children were first introduced to the robot and the aim of the interaction, then had a first pre-test to evaluate their initial knowledge. Before starting the teaching game, children have to complete a tutorial where they are introduced to the mechanics of the game: animals have life and have to eat to survive and children can move animals to make them interact with other animals or plants. After this short tutorial, they have to complete two sessions of the game where the robot can provide feedback and advices depending which conditions they are in. After these initial sessions of the game they have to complete a mid-test before playing another 2 sessions of the game and a last post-test before concluding the study.

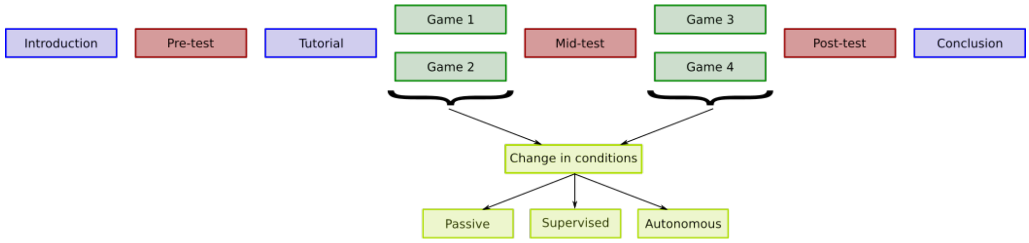


Figure 6.4: Methodology used for the study.

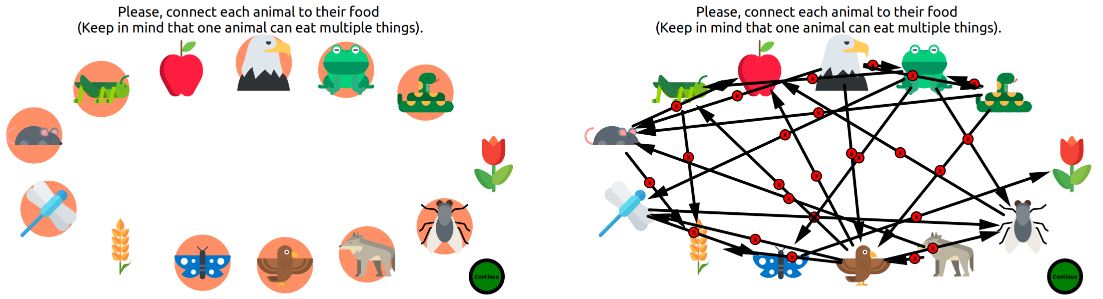
In all the conditions, the robot's behaviour during the introduction, tests, tutorial and conclusion is identical. The only change of behaviour happens during the games sessions. Figure **??** shows an example of the game screen. The child can move 10 animals across the game field and can have them interact with other animals or plants. Animals lose energy over time and by interacting with their food the can regain some. Animals that are eaten lose a chunk of their life. The goal for the children is to keep animals alive as long as possible by feeding them and they earn stars representing how healthy their animals have been during the session. The game stops when 3 or more animals run out of energy and each game session lasted 1.6 minutes in average.

6.3.2 Hypotheses

6.3.3 Metrics

Learning Evaluation

During the pre-test, the experimenter demonstrate how to connect animals by drawing an arrow from the frog to the fly, and they removing the arrow by pressing the *X* button. Then, children are asked to connect as many animals as possible. Figure 6.5 shows two examples of test, without or with all correct connections. When they think they are done, they can press the continue button, showing a screen asking confirmation to quit the test or give the opportunity to keep connecting animals. Additionally, the robot inform the child if not all the animals are connected to their food or that animal can eat many types of food if no more than one animal has been connected to two items. They are in total 25 different correct connections and 95 possible incorrect ones. As the child can connect as many arrows as desired, the performance is defined as the number of correct arrows above chance for the total number of connected arrows on the test divided by the maximum achievable performance to reach a score with a ceiling at 1.


Please, connect each animal to their food
(Keep in mind that one animal can eat multiple things).
Continue
Please, connect each animal to their food
(Keep in mind that one animal can eat multiple things).
Continue

(a) Empty screen that children face at each test. Red dots behind animals represent that they are not been connected to any food.

(b) Fully connected test with all the correct connections.

Figure 6.5: Test screen to evaluate children's knowledge, empty starting screen (a) and fully connected and correct test (b).

6.4 Results

6.5 Discussion

6.6 Summary

Chapter 7

Discussion

7.1 Experimental Limitations

7.1.1 Ecological Validity and Generalisability

7.2 Ethical Questions

7.3 Summary

Chapter 8

Contribution and Conclusion

This chapter seeks to provide an overview of the findings and topics covered in this thesis. The contributions to the field of social HRI are outlined and summarised. Following this, a conclusion is provided to briefly encapsulate the primary outcome of this work.

8.1 Summary

8.2 Contributions

This section will revisit the contributions outlined in the introduction (Chapter 1), with further expansion and explanation. The main contributions of this thesis are as follows:

- Something

8.3 Conclusion

Acronyms

ASD Autism Spetrum Disorder. 6, 12, 41

CML Classical Machine Learning. 25

DREAM Development of Robot-Enhanced Therapy for Children with Autism Spectrum Disorders (European FP7 project). 6, 39

GUI Graphical User Interface. vii, 42–46, 48

HCI Human-Computer Interaction. 15, 22

HRI Human-Robot Interaction. 1, 2, 5, 11, 13, 15–17, 21, 22, 24, 27, 28, 30, 32, 35, 53, 58, 71

IML Interactive Machine Learning. 2, 24–26, 30, 32, 34, 35, 55, 56, 58

IRL Interactive Reinforcement Learning. 55–58, 60–64

LfD Learning from Demonstration. 18, 24, 32, 34, 36

MDP Markov Decision Process. 22

ML Machine Learning. 2, 3, 28, 30

RAT Robot Assisted Therapy. 6, 14, 17, 19, 33, 40, 41, 45, 53

RL Reinforcement Learning. 2, 22–26, 29, 32, 33, 36, 55, 56

SAR Socially Assistive Robotics. 6

SL Supervised Learning. 26, 36, 56

SPARC Supervided Progressive Autonomous Robot Competenties. i, vii, 3, 32–36, 39, 41, 42, 46, 48, 49, 52, 53, 55–58, 60, 62, 63

WoZ Wizard-of-Oz. 2, 15, 16, 19, 36, 39–42, 45, 46, 48, 49, 52, 53

Appendices

Appendix A

A1

Bibliography

Abbeel, P., & Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*.

Adams, J. A. (2002). Critical considerations for human-robot interface development. In *Proceedings of 2002 AAAI Fall Symposium*, (pp. 1–8).

Alili, S., Alami, R., & Montreuil, V. (2009). A task planner for an autonomous social robot. In *Distributed Autonomous Robotic Systems 8*, (pp. 335–344). Springer.

Amershi, S., Cakmak, M., Knox, W. B., & Kulesza, T. (2014). Power to the people: The role of humans in interactive machine learning. *AI Magazine*, *35*(4), 105–120.

Argall, B. D., Chernova, S., Veloso, M., & Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and autonomous systems*, *57*(5), 469–483.

Arkin, R. C., Fujita, M., Takagi, T., & Hasegawa, R. (2003). An ethological and emotional basis for human–robot interaction. *Robotics and Autonomous Systems*, *42*(3-4), 191–201.

Bartneck, C., & Forlizzi, J. (2004). A design-centred framework for social human-robot interaction. In *Proceedings of the 13th IEEE International Workshop on Robot and Human Interactive Communication*, (pp. 31–33).

Baxter, P., Kennedy, J., Senft, E., Lemaignan, S., & Belpaeme, T. (2016). From characterising three years of hri to methodology and reporting recommendations. In *The Eleventh ACM/IEEE International Conference on Human Robot Interation*, (pp. 391–398). IEEE Press.

Baxter, P., Wood, R., & Belpaeme, T. (2012). A touchscreen-based âĂŸsandtrayâĂŹ to facilitate, mediate and contextualise human-robot social interaction. In *Human-Robot Interaction (HRI), 7th ACM/IEEE International Conference on*, (pp. 105–106).

Beer, J., Fisk, A. D., & Rogers, W. A. (2014). Toward a framework for levels of robot autonomy in human-robot interaction. *Journal of Human-Robot Interaction*, *3*(2), 74.

Beetz, M., Kirsch, A., & Muller, A. (2004). Rpllearn: Extending an autonomous robot control language to perform. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 3*, (pp. 1022–1029). IEEE Computer Society.

Belpaeme, T., Baxter, P. E., Read, R., Wood, R., Cuayáhuitl, H., Kiefer, B., Racioppa, S., Kruijff-Korbayová, I., Athanasopoulos, G., Enescu, V., et al. (2012). Multimodal child-robot interaction: Building social bonds. *Journal of Human-Robot Interaction*, *1*(2), 33–53.

Billard, A., Calinon, S., Dillmann, R., & Schaal, S. (2008). Robot programming by demonstration. In *Springer handbook of robotics*, (pp. 1371–1394). Springer.

Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational researcher*, *13*(6), 4–16.

Breazeal, C. (1998). A motivational system for regulating human-robot interaction. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, (pp. 54–62). AAAI.

Broadbent, E., Stafford, R., & MacDonald, B. (2009). Acceptance of healthcare robots for the older population: review and future directions. *International Journal of Social Robotics*, *1*(4), 319–330.

Burgard, W., Cremers, A. B., Fox, D., Hähnel, D., Lakemeyer, G., Schulz, D., Steiner, W., & Thrun, S. (1999). Experiences with an interactive museum tour-guide robot. *Artificial intelligence*, *114*(1-2), 3–55.

Cakmak, M., Chao, C., & Thomaz, A. L. (2010). Designing interactions for robot active learners. *IEEE Transactions on Autonomous Mental Development*, *2*(2), 108–118.

Cao, H.-L., Esteban, P. G., Simut, R., Van de Perre, G., Lefeber, D., Vanderborght, B., et al. (2017). A collaborative homeostatic-based behavior controller for social robots in human–robot interaction experiments. *International Journal of Social Robotics*, *9*(5), 675–690.

Chao, C., Cakmak, M., & Thomaz, A. L. (2010). Transparent active learning for robots. In *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*, (pp. 317–324). IEEE.

Chernova, S., & Veloso, M. (2009). Interactive policy learning through confidence-based autonomy. *Journal of Artificial Intelligence Research*, *34*(1), 1.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, (pp. 4302–4310).

Clark-Turner, M., & Begum, M. (2018). Deep reinforcement learning of abstract reasoning from demonstrations. In *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, (pp. 372–372). ACM.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, *13*(1), 21–27.

Dautenhahn, K. (1999). Robots as social actors: Aurora and the case of autism. In *Proc. CT99, The Third International Cognitive Technology Conference, August, San Francisco*, vol. 359, (p. 374).

Di Nuovo, A., Broz, F., Belpaeme, T., Cangelosi, A., Cavallo, F., Esposito, R., & Dario, P. (2014). A web based multi-modal interface for elderly users of the robot-era multi-robot services. In *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*, (pp. 2186–2191). IEEE.

Diehl, J. J., Schmitt, L. M., Villano, M., & Crowell, C. R. (2012). The clinical use of robots for individuals with autism spectrum disorders: A critical review. *Research in autism spectrum disorders*, *6*(1), 249–262.

Dienes, Z. (2011). Bayesian versus orthodox statistics: Which side are you on? *Perspectives on Psychological Science*, *6*(3), 274–290.

Dragan, A. D., Lee, K. C., & Srinivasa, S. S. (2013). Legibility and predictability of robot motion. In *Human-Robot Interaction (HRI), 2013 8th ACM/IEEE International Conference on*, (pp. 301–308). IEEE.

Esteban, P. G., Baxter, P., Belpaeme, T., Billing, E., Cai, H., Cao, H.-L., Coeckelbergh, M., Costescu, C., David, D., De Beir, A., et al. (2017). How to build a supervised autonomous system for robot-enhanced therapy for children with autism spectrum disorder. *Paladyn, Journal of Behavioral Robotics*, *8*(1), 18–38.

Fails, J. A., & Olsen Jr, D. R. (2003). Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces*, (pp. 39–45). ACM.

Feil-Seifer, D., & Matarić, M. J. (2005). Defining socially assistive robotics. In *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on*, (pp. 465–468). IEEE.

Fincannon, T., Barnes, L. E., Murphy, R. R., & Riddle, D. L. (2004). Evidence of the need for social intelligence in rescue robots. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 2, (pp. 1089–1095). IEEE.

Fink, J., Bauwens, V., Kaplan, F., & Dillenbourg, P. (2013). Living with a vacuum cleaning robot. *International Journal of Social Robotics*, *5*(3), 389–408.

Fong, T., Nourbakhsh, I., & Dautenhahn, K. (2003). A survey of socially interactive robots. *Robotics and Autonomous Systems*, *42*(3-4), 143–166.
URL `http://linkinghub.elsevier.com/retrieve/pii/S092188900200372X`

Frager, S., & Stern, C. (1970). Learning by teaching. *The Reading Teacher*, *23*(5), 403–417.

Friedman, B., Kahn Jr, P. H., & Hagman, J. (2003). Hardware companions?: What online aibo discussion forums reveal about the human-robotic relationship. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, (pp. 273–280). ACM.

García, J., & Fernández, F. (2015). A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, *16*, 1437–1480.

Gockley, R., Bruce, A., Forlizzi, J., Michalowski, M., Mundell, A., Rosenthal, S., Sellner, B., Simmons, R., Snipes, K., Schultz, A. C., et al. (2005). Designing robots for long-term social interaction. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, (pp. 1338–1343). IEEE.

Griffith, S., Subramanian, K., Scholz, J., Isbell, C., & Thomaz, A. L. (2013). Policy shaping: Integrating human feedback with reinforcement learning. In *Advances in Neural Information Processing Systems*, (pp. 2625–2633).

Guizzo, E., & Ackerman, E. (2012). How rethink robotics built its new baxter robot worker. *IEEE spectrum*, (p. 18).

Han, J., Jo, M., Park, S., & Kim, S. (2005). The educational use of home robots for children. In *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*, (pp. 378–383). IEEE.

Hart, S. G., & Staveland, L. E. (1988). Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Advances in psychology*, *52*, 139–183.

Harwin, W., Ginige, A., & Jackson, R. (1988). A robot workstation for use in education of the physically handicapped. *IEEE Transactions on Biomedical Engineering*, *35*(2), 127–131.

Hoffman, G. (2016). Openwoz: A runtime-configurable wizard-of-oz framework for human-robot interaction. In *2016 AAAI Spring Symposium Series*.

Hood, D., Lemaignan, S., & Dillenbourg, P. (2015). When children teach a robot to write: An autonomous teachable humanoid which uses simulated handwriting. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, (pp. 83–90). ACM.

Howley, I., Kanda, T., Hayashi, K., & Rosé, C. (2014). Effects of social presence and social role on help-seeking and learning. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, (pp. 415–422). ACM.

Isbell, C. L., Kearns, M., Singh, S., Shelton, C. R., Stone, P., & Kormann, D. (2006). Cobot in lambdamoo: An adaptive social statistics agent. *Autonomous Agents and Multi-Agent Systems*, *13*(3), 327–354.

Jain, A., Wojcik, B., Joachims, T., & Saxena, A. (2013). Learning trajectory preferences for manipulators via iterative improvement. In *Advances in neural information processing systems*, (pp. 575–583).

JASP Team (2018). JASP (Version 0.8.6)[Computer software].
URL `https://jasp-stats.org/`

Jeffreys, H. (1998). *The theory of probability*. OUP Oxford.

Johnson, D. W., Johnson, R. T., & Smith, K. A. (1991). *Active learning: Cooperation in the college classroom*. Interaction Book Company Edina, MN.

Judah, K., Roy, S., Fern, A., & Dietterich, T. G. (2010). Reinforcement learning via practice and critique advice. In *AAAI*.

Kanda, T., Glas, D. F., Shiomi, M., Ishiguro, H., & Hagita, N. (2008). Who will be the customer?: A social robot that anticipates people's behavior from their trajectories. In *Proceedings of the 10th international conference on Ubiquitous computing*, (pp. 380–389). ACM.

Kanda, T., Hirano, T., Eaton, D., & Ishiguro, H. (2004). Interactive robots as social partners and peer tutors for children: A field trial. *Human-computer interaction*, *19*(1), 61–84.

Kanda, T., Shiomi, M., Miyashita, Z., Ishiguro, H., & Hagita, N. (2009). An affective guide robot in a shopping mall. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, (pp. 173–180). ACM.

Kaochar, T., Peralta, R. T., Morrison, C. T., Fasel, I. R., Walsh, T. J., & Cohen, P. R. (2011). Towards understanding how humans teach robots. In *International Conference on User Modeling, Adaptation, and Personalization*, (pp. 347–352). Springer.

Kelley, J. F. (1983). An empirical methodology for writing user-friendly natural language computer applications. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, (pp. 193–196). ACM.

Kennedy, J., Baxter, P., & Belpaeme, T. (2015). The robot who tried too hard: social behaviour of a robot tutor can negatively affect child learning. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, (pp. 67–74). ACM.

Kirsch, A. (2009). Robot learning languageâĂŤintegrating programming and learning for cognitive systems. *Robotics and Autonomous Systems*, *57*(9), 943–954.

Knox, W. B., Spaulding, S., & Breazeal, C. (2014). Learning social interaction from the wizard: A proposal. In *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*.

Knox, W. B., Spaulding, S., & Breazeal, C. (2016). Learning from the wizard: Programming social interaction through teleoperated demonstrations. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, (pp. 1309–1310). International Foundation for Autonomous Agents and Multiagent Systems.

Knox, W. B., & Stone, P. (2009). Interactively shaping agents via human reinforcement: The tamer framework. In *Proceedings of the fifth international conference on Knowledge capture*, (pp. 9–16). ACM.

Knox, W. B., & Stone, P. (2010). Combining manual feedback with subsequent mdp reward signals for reinforcement learning. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, (pp. 5–12).

Kober, J., Bagnell, J. A., & Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, *32*(11), 1238–1274.

Lara, J. S., Casas, J., Aguirre, A., Munera, M., Rincon-Roncancio, M., Irfan, B., Senft, E., Belpaeme, T., & Cifuentes, C. A. (2017). Human-robot sensor interface for cardiac rehabilitation. In *Rehabilitation Robotics (ICORR), 2017 International Conference on*, (pp. 1013–1018). IEEE.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*.

Lemaignan, S., Edmunds, C., Senft, E., & Belpaeme, T. (2017). The free-play sandbox: a methodology for the evaluation of social robotics and a dataset of social interactions. *arXiv preprint arXiv:1712.02421*.

Leyzberg, D., Spaulding, S., & Scassellati, B. (2014). Personalizing robot tutors to individuals' learning differences. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, (pp. 423–430). ACM.

Liu, P., Glas, D. F., Kanda, T., Ishiguro, H., & Hagita, N. (2014). How to train your robot-teaching service robots to reproduce human social behavior. In *Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on*, (pp. 961–968).

Loftin, R., Peng, B., MacGlashan, J., Littman, M. L., Taylor, M. E., Huang, J., & Roberts, D. L. (2016). Learning behaviors via human-delivered discrete feedback: modeling implicit feedback strategies to speed up learning. *Autonomous agents and multi-agent systems*, *30*(1), 30–59.

MacGlashan, J., Ho, M. K., Loftin, R., Peng, B., Wang, G., Roberts, D. L., Taylor, M. E., & Littman, M. L. (2017). Interactive learning from policy-dependent human feedback. In *Proceedings of the 34th International Conference on Machine Learning*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533.

Montreuil, V., Clodic, A., Ransan, M., & Alami, R. (2007). Planning human centered robot activities. In *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, (pp. 2618–2623). IEEE.

Mubin, O., Stevens, C. J., Shahid, S., Al Mahmud, A., & Dong, J.-J. (2013). A review of the applicability of robots in education. *Journal of Technology in Education and Learning*, *1*(209-0015), 13.

Munzer, T., Toussaint, M., & Lopes, M. (2017). Efficient behavior learning in human–robot collaboration. *Autonomous Robots*, (pp. 1–13).

Murphy, R. R., Tadokoro, S., Nardi, D., Jacoff, A., Fiorini, P., Choset, H., & Erkmen, A. M. (2008). Search and rescue robotics. In *Springer Handbook of Robotics*, (pp. 1151–1173). Springer.

Riek, L. (2012). Wizard of Oz Studies in HRI: A Systematic Review and New Reporting Guidelines. *Journal of Human-Robot Interaction*, *1*(1), 119–136.

Rieser, V., & Lemon, O. (2008). Learning effective multimodal dialogue strategies from wizard-of-oz data: Bootstrapping and evaluation. *Proceedings of ACL-08: HLT*, (pp. 638–646).

Scheutz, M., Krause, E., Oosterveld, B., Frasca, T., & Platt, R. (2017). Spoken instruction-based one-shot object and action learning in a cognitive robotic architecture. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, (pp. 1378–1386). International Foundation for Autonomous Agents and Multiagent Systems.

Senft, E., Baxter, P., Kennedy, J., & Belpaeme, T. (2015). Sparc: Supervised progressively autonomous robot competencies. In *International Conference on Social Robotics*, (pp. 603–612).

Senft, E., Baxter, P., Kennedy, J., Lemaignan, S., & Belpaeme, T. (2016). Providing a robot with learning abilities improves its perception by users. In *The Eleventh ACM/IEEE International Conference on Human Robot Interation*, (pp. 513–514). IEEE Press.

Senft, E., Baxter, P., Kennedy, J., Lemaignan, S., & Belpaeme, T. (2017a). Supervised autonomy for online learning in human-robot interaction. *Pattern Recognition Letters*, *99*, 77–86.

Senft, E., Lemaignan, S., Baxter, P., & Belpaeme, T. (2017b). Toward supervised reinforcement learning with partial states for social hri. In *Proceedings of 2017 AAAI Fall Symposium - AIHRI*.

Sequeira, P., Alves-Oliveira, P., Ribeiro, T., Di Tullio, E., Petisca, S., Melo, F. S., Castellano, G., & Paiva, A. (2016). Discovering social interaction strategies for robots from restricted-perception wizard-of-oz studies. In *The Eleventh ACM/IEEE International Conference on Human Robot Interation*, (pp. 197–204). IEEE Press.

Settles, B. (2009). Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Sheridan, T. B., & Verplank, W. L. (1978). Human and computer control of undersea teleoperators. Tech. rep., MASSACHUSETTS INST OF TECH CAMBRIDGE MAN-MACHINE SYSTEMS LAB.

Sherif, M. (1936). *The psychology of social norms.*. Harper.

Shiomi, M., Sakamoto, D., Kanda, T., Ishi, C. T., Ishiguro, H., & Hagita, N. (2008). A semi-autonomous communication robot: a field trial at a train station. In *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, (pp. 303–310). ACM.

Singer, P. W. (2009). *Wired for war: The robotics revolution and conflict in the 21st century*. Penguin.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. MIT press.

Tapus, A., Mataric, M. J., & Scassellati, B. (2007). Socially assistive robotics [grand challenges of robotics]. *IEEE Robotics & Automation Magazine*, *14*(1), 35–42.

Theocharous, G., Thomas, P. S., & Ghavamzadeh, M. (2015). Personalized ad recommendation systems for life-time value optimization with guarantees. In *IJCAI*, (pp. 1806–1812).

Thill, S., Pop, C. A., Belpaeme, T., Ziemke, T., & Vanderborght, B. (2012). Robot-assisted therapy for autism spectrum disorders with (partially) autonomous control: Challenges and outlook. *Paladyn*, *3*(4), 209–217.

Thomaz, A. L., & Breazeal, C. (2008). Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence*, *172*(6), 716–737.

Thrun, S., Bennewitz, M., Burgard, W., Cremers, A. B., Dellaert, F., Fox, D., Hahnel, D., Rosenberg, C., Roy, N., Schulte, J., et al. (1999). Minerva: A second-generation museum tour-guide robot. In *Robotics and automation, 1999. Proceedings. 1999 IEEE international conference on*, vol. 3. IEEE.

United Nations. Department of Economic and Social Affairs (2017). *World Population Prospects: The 2017 Revision. Key findings and advance tables*. United Nations Publications.

Verner, I. M., Polishuk, A., & Krayner, N. (2016). Science class with robothespian: using a robot teacher to make science fun and engage students. *IEEE Robotics & Automation Magazine*, *23*(2), 74–80.

Wada, K., Shibata, T., Saito, T., Sakamoto, K., & Tanie, K. (2005). Psychological and social effects of one year robot assisted activity on elderly people at a health service facility for the aged. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, (pp. 2785–2790). IEEE.

Wada, K., Shibata, T., Saito, T., & Tanie, K. (2004). Effects of robot-assisted activity for elderly people and nurses at a day service center. *Proceedings of the IEEE*, *92*(11), 1780–1788.

Złotowski, J., Proudfoot, D., Yogeeswaran, K., & Bartneck, C. (2015). Anthropomorphism: opportunities and challenges in human-robot interaction. *International Journal of Social Robotics*, *7*(3), 347–360.