UNIVERSITY OF
PLYMOUTH

Teaching Robots Social Autonomy From In Situ Human Supervision

by

Emmanuel Senft

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

DOCTOR OF PHILOSOPHY

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

July 2018

Acknowledgements

Thanks Séverin, Paul and of course Tony

And Charlotte, Maddie and Chris for being nice and proofreading the whole thing!

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

Authors 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: _____

Contents

List of Figures

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

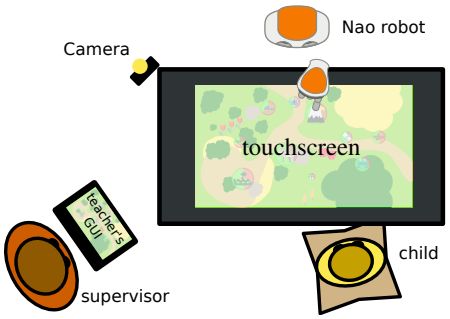


Figure 1.1: The setup used in the study: a child interacts with the robot tutor, with a large touchscreen sitting between them displaying the learning activity; a human teacher provides guidance to the robot through a tablet and monitors the robot's learning.

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?**

  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 and terminology

Throughout this thesis, the terms ‘wizard’, ‘supervisor’ and ‘teacher’ have been used interchangeably to represent the people in control of a robot’s action and teaching that robot an action policy.

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 social Human-Robot Interaction (HRI) and presents related research in agent and robot control. The first section introduces the different fields of application of social HRI and draws from them requirements for controlling a robot interacting with humans (the robot should: only execute appropriate actions and have a high level of adaptivity and autonomy). The second section provides the current state of the art in robot control for HRI and analyses it through the requirements presented in the previous section. And finally, the third section presents Interactive Machine Learning (IML), an alternative method holding promises 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

HRI covers the full spectrum of interactions between humans and robots ranging from physical assistance and teleoperation to social companions. However, this thesis being focused on teaching robots to interact socially with humans, the following criteria on the interactions were used to select the subfields of HRI relevant to this research:

- Presence of an interaction between a robot and a human: both the human and the robot are influencing each other's behaviour.
- The robot is "socially interactive" (Fong et al., 2003). The *social* element of the interaction between the robot and the human is key. As such, it rules out purely physical human-robot interaction, such as exoskeleton, physical rehabilitation or pure teleoperation as in robotic assisted surgery.

Applying these constraints on HRI resulted on five subfields involving social interactions between robots and humans:

- Socially Assistive Robotics (SAR)

- Education
- Search and Rescue, and Military
- Hospitality and Entertainment
- Collaborative Robots in Industry

Socially Assistive Robotics

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

One of the main applications of SAR is care for the elderly. In the near future, the ageing of population will have large impacts on the world, and societies will have to find solutions to tackle this challenge. The United Nations' Department of Economic and Social Affairs (2017) reports that the "population aged 60 or over is growing faster than all younger age groups". This unbalance of growth will decrease the support ratio (number of worker per retiree) forcing societies to find ways to provide care to an increasing number of people using a decreasing staff. Robots represent a unique opportunity to compensate for this lacking workforce potentially allowing elderly 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).

The second main application of SAR is Robot Assisted Therapy (RAT). Robots might might be used to provide therapies or to follow and support a patient during their rehabilitation to improve their health, their acceptance in society or recover better from an accident. In therapies, robots were first used as physical platforms to help patient in rehabilitation therapy during the 80s (Harwin et al., 1988). During this period, robots were primally used as mechatronic tools helping humans to accomplish repetitive task. But in the late 90s, robots 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 other projects, such as the DREAM project[1], have started all around the world to use robots to help patient with ASD (Diehl et al., 2012; Esteban et al., 2017).

RAT is not limited to ASD only, the use of robots is also explored in hospitals, for

[1] https://www.dream2020.eu

example to support children with diabetes (Belpaeme et al., 2012), to support elderly with dementia (Wada et al., 2005) and stroke recovering patients (Matarić et al., 2007) or to monitor and provide encouragement in cardiac rehabilitation therapies (Lara et al., 2017).

These examples demonstrate that already today, robots are interacting with sensible populations (the elderly or patients in therapy). This implies that robots' social behaviours need to be constantly correct to ensure that no harm will be caused to these populations. And due to the shortage of workforce, these robots need to be as autonomous as possible.

Education

Social robots are being used in education, supporting teachers to provide learning content to children, transforming the teaching process from passive learning to active learning (Linder et al., 2001). In education, robots can take multiple roles such as peer, tutor or tool (Mubin et al., 2013). It should be noted that Mubin et al. stress that the intentions of the robotic community is not to replace teachers by robots, but provide them with a new form of technological teaching-aid. Nevertheless, Verner et al. (2016) presented positive results from an early stage study using a tall humanoid robot to deliver a science lesson to children. The remaining of the section will describe more in details the roles of tutor and peer robots (the 'tool' role has been excluded due to the lack of social interaction between the robot and the students).

Robotics tutors providing tailored teaching content in 1 to 1 interaction. Individualised feedback has been shown to increase the performance of students (Cohen et al., 1982; Bloom, 1984). However, tutoring requires a larger trained staff and as such is more costly than large classes supervised by a single teacher. For this reason, tutoring is seldom used in public education today. Tutoring is however often available through private teacher at additional cost for the parents, potentially increasing social inequalities (Bray, 2009). Robotic tutors could provide this powerful one to one tailored interaction to every student during school time, leading to higher learning gain for the children without dramatically increasing the workload on teachers (Kanda et al., 2004; Leyzberg et al., 2012; Kennedy et al., 2016; Gordon et al., 2016). In addition to classroom uses, robots could also support children at home as they have been shown to elicit advantages over web or paper based instructions (Han et al., 2005).

Peer robots learning alongside the children. Unlike other types of agents in education, peer robots have the opportunity to fit a special role seldom present in education today: the role of care-receiver rather than care-giver (Tanaka & Matsuzoe, 2012). Peer learning has demonstrated benefits both for the helper and those helped in Human-Human Interaction (HHI) (Topping, 2005). In HRI, a peer robot does not mentor a child to teach them new concepts, but learns alongside or from them, supporting them during the process and encouraging these children to produce behaviours improving their own learning. For example, in the Co-Writer project (Hood et al., 2015), a child has to teach a robot how to write, and as the child demonstrates correct handwriting to the robot, they improve their own skills at drawing letters. Peer robots are able to leverage the concept of learning by teaching (Frager & Stern, 1970) and peer learning (Topping, 2005) in a way hardly matched 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. In a similar position, an adult would not be a believable agent requiring learning and younger students might not have the compliance and the patience to learn from another child.

To provide efficient tutoring or peer support, robots need to be able to personalise their behaviour to the children they are interacting with in order to maximise the children's learning gain. Additionally, as pointed by Kennedy et al. (2015), a robot too social could decrease the learning for the children compared to a less social robot, consequently the robot's social behaviours have to be carefully managed to ensure its effectiveness.

Search and Rescue, and Military

Robots are already deployed in the real world, outside of labs and used during search and rescue missions (Murphy et al., 2008). For instance, after a natural or artificial catastrophe, robots have been sent to analyse the damaged area and report or rescue the surviving victims of the incident. During these missions, 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 is probably in a shocked state and the searching robot could be the first link they have with the external world after the accident (Murphy et al., 2008). 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 high pressure to act quickly and faces traumatic events

too. Even if the robot does not display a social behaviour, rescuers interacting with it might develop some feeling toward the robots they are using during these tense moments (Fincannon et al., 2004).

Similar human behaviours (i.e. emotional bonding with a robot) have also been observed in the army (Singer, 2009). Robots have been deployed as teleoperated drones and ground units alongside soldiers to complete scouting task or cleaning minefields. By interacting with a robot for extensive periods, some soldiers developed feelings toward this robot they used in a daily basis: taking pictures with it and introducing it to their friends. These relationships have gone as far as soldier risking their own life to in order to save the robot used by their squad (Singer, 2009).

These examples in these two fields demonstrate that the sociability of the robot has to be taken into account when interacting in such stressful environments. Overlooking the importance of social relationships human will form can lead to dramatic consequences (e.g. soldiers taking risks for the robot). As such, during the interaction, the robot's behaviour needs to be constantly appropriate not to create misleading expectations and to ensure that the goal of the interaction will be met.

Hospitality and Entertainment

Robots are also interacting with humans in hotels around the world; for example, the Relay robot (Savioke[2]) delivers amenities directly to the guests' rooms in more than 70 hotels in the US, Europe and Japan[3]. Whilst the social interaction is still minimal today, these robots interact everyday with humans and have been seen evoking social reactions from them[4].

On the research side, scientists have designed and tested a robot as a receptionist in a hall at Carnegie Melon University (Gockley et al., 2005). Since the late 90s, researchers also explored how robots could guide visitors in museums and exhibitions (Thrun et al., 1999; Burgard et al., 1999). These researches continue today to explore how to improve the social interaction between tourists and these guide robots (Bennewitz et al., 2005). Similarly, robots also inform and advise humans in shops and shopping malls. Long term studies have explored how humans perceive and interact with robots in this environment

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

[3] https://www.spectrum.ieee.org/view-from-the-valley/robotics/industrial-robots/ces-2018-delivery-robots-are-fulltime-employees-at-a-las-vegas-hotel

[4] https://www.fastcodesign.com/3057075/how-savioke-labs-built-a-robot-personality-in-5-days

(Kanda et al., 2009) and how robots should behave with clients (Kanda et al., 2008).

Robots have also entered homes and family circles: from vacuum cleaners (e.g Roomba) to companion robot (e.g Pepper) passing by pet robots (e.g Aibo). A notable example is Aibo: twenty years ago, Sony created Aibo, a robotic dog to be used as pet in Japanese families and a new version was released in early 2018. An analysis of online discussions of owners published 6 years after Aibo's first introduction gives insights on the relationship that owners created with their robots (Friedman et al., 2003). For example 42% of the community members assigned intentionality to the robot, such as preferences, emotions or even feelings. Similar behaviours have also been observed when the robot is not presented as a pet, but even just a a tool. For instance, Fink et al. (2013) report that one of their participants was worried that their Roomba would feel lonely when they would be away on holiday. More recently, the Pepper robot has been sold to families in Japan[5]. However, as of early 2018, no study in English has reported results of the interactions with families or long term use and acceptance.

Similarly to previous fields, robots' behaviour (social or not social) will impact their users in ways hardly predictable in advance. This reinforces the necessity for their behaviour to remain appropriate in any step of the interaction. Additionally, when interacting with unknown users, the robot will have to face a wide range of users' expectations, and react to different unanticipated behaviours. This forces the robot to adapt to these different users and react according to their needs and desires. Finally, robots will also be deployed to interact with the same people over long periods. To sustain engagement over such time scales, these robots need to change their behaviour over time to overcome possible boredom due to the vanishement of the novelty effect in repeated interactions (Salter et al., 2004), for instance by referring to previous experiences and enrich their behaviour (Leite et al., 2013).

Collaborative Robots in Industry

In industry, robots 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; they share the same workspace and interact physically and socially with factory workers. As these robots need to be safe to human interacting with them, they are often softly actuated,

[5] https://www.softbankrobotics.com/emea/en/robots/pepper

using active or passive compliance to avoid the risk related to collision between a stiff robot and a human. Additionally, when collaborating with people, the legibility of motion is fundamental. In other words, 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 also need to interpret human social cue to infer their goals and intentions.

Beyond safety and legibility, another key challenge in Human-Robot Collaboration (HRC) is task assignment: if a goal has to be achieved by a human-robot team, the partitioning of tasks should be carefully managed to optimise the end result in term of task performance, but also to ensure comfort for the human. Explicit and implicit rules and personal preferences describe the role and behaviours of each participant and have to be taken into account in HRC. As such, the task repartition system and other planners used in HRC should be aware of these social norms and follow them (Montreuil et al., 2007). For example, recent work done in the 3rd Hand project[6] explored how a robot should support a human in a collaborative assembly task by adapting its behaviour to this human's personal preferences and how this adaptation improve the team's efficiency (Munzer et al., 2017).

As demonstrated in Munzer et al. (2017), intelligent systems involved in HRC should adapt their behaviour to the interaction partners, be aware of preferences and rules to follow to ensure that the robot's behaviour is always appropriate, efficient and safe for the humans involved in the interaction. An additional challenge faced by robots with the recent advances in Machine Learning (ML), and especially with the omnipresence of neural networks and deep learning, is Explainable Artificial Intelligence (XAI) (Wachter et al., 2017). As agents learning to interact will make mistakes and behave unexpectedly from time to time, they have to be able to provide explanations for these errors in a way understandable by humans. This challenge is especially visible in HRC where both humans and robots aim to collaborate to complete a task together. Hayes & Shah (2017) propose to achieve transparency through policy explanation, by allowing robots to answer questions and explain their behaviour by self observation and logic deduction. This transparency aims at increasing trust between the agents involved in the interaction and improve the robot's efficiency.

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

2.1.2 Requirements on Robots Interacting with Humans

The review in Section 2.1.1 demonstrated that robots are already interacting with vulnerable populations: young children, the elderly, people requiring healthcare or in a stressful situations (victims of catastrophes or soldiers for example). Additionally, people tend to a create emotional bonding with these robots even if they are not interacting socially with their users. As such, the behaviour of robots interacting with humans need to be carefully controlled to manage humans' expectations and ensure their safety. In other words, robots need to constantly behave in socially acceptable manners, avoiding any confusing, inappropriate or dangerous behaviour. Failure to do so might prevent the interaction to fit its goal or even lead to physical injuries or potentially elicit offence, anger, frustration, distress or boredom.

These undesired behaviours may come from 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 policies. Due to the wide range of origins of these potential social faux-pas, this research focuses only on the last point, obtaining an appropriate action policy: assuming a set of inputs, finding a way to have the robot select an appropriate action. The other issues are either orthogonal and would lead to failure even with an 'optimal' action policy as external factors prevent the robot from solving the problem or could be handled by having a better action policy (e.g. a policy generalising more efficiently or selecting suboptimal actions when the optimal ones are not available).

Appropriate actions are highly dependent on the interaction context: they could aim to match or reduce users' expectations of a robot's behaviour or complete a specific task. Additionally, actions correct in a certain context might be inappropriate in another one. Nonetheless, the intention behind being *appropriate* here is that actions executed should be guaranteed not to present risks for the humans involved in the interaction (for example, preventing physical harm or mental distress), while helping the robot to move toward its goal and achieving its objectives.

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 people, for extended

periods of time or with initial incomplete or incorrect knowledge. As such, the action policy also needs to be adaptable to the context and users as well as evolve over time. In summary, robots need to be adaptive, to react to changing environments, cover a larger field of application and improve their action policy over time.

Lastly, in many cases today, interactive robots are not autonomous but partially controlled by a human operator (Riek, 2012). We argue that to have a real and useful HRI, the robot needs to be as autonomous as possible. Robot are expected to be used in area where the workforce is already in shortage (e.g. healthcare) and requiring humans to control these robots decrease widely their applicability. As a community, HRI should strive toward more autonomy for robots interacting with humans.

Based on these considerations, we define three axes to analyse robot controllers and evaluate how suited these controller are to interact with humans. Each axis is associated to a principle the robot has to follow to sustain meaningful social interactions:

1. Appropriateness of actions - The robot should only execute appropriate actions.
2. Adaptivity - The robot should be adaptivity to its environment and in time.
3. Autonomy - The robot should be as autonomous as possible.

We will use these axes to analyse current robotic controller types in Section 2.2.

As HRI is 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, the dependence and knowledge of social rules or the range of application of a robot to cite only a few. However, we ignored these axes in the current work as they are more influenced by the goal and the context of the specific human-robot interaction taking place than the action policy itself. Additionally, an appropriate, adaptive, and autonomous robot should be able to safely and autonomously learn to interact in deeper and more complex interactions and learn to extend its abilities beyond the ones it initially started with to increase its range of applications.

Appropriateness of Actions

As argued previously, much of social human-robot interaction takes place in stressful or sensitive environments, where humans have particular expectations about a robot's behaviour. 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). And some of these expectations are also transferred to interactions with robots (Bartneck & Forlizzi, 2004).

We define appropriate actions, as actions taking into account the social side of the interaction, and producing a correct robot behaviour at the right time. This behaviours needs to be safe for surrounding humans and help the robot to reach its goal. Failing to produce these appropriate actions, for example by not matching the users' expectations, may have a negative impact on the interaction, potentially compromising future interactions if the human feel disrespected, confused or annoyed. Furthermore, failing to behave appropriately can even harm the people interacting with the robot: 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 might lead to dramatic consequences. Robots require a way to ensure that all the actions they execute do not present risks to the humans involved in the interaction while moving the robot closer to its goal.

However, being able to behave appropriately is a tremendous challenge: real interactions involve a large sensory space, with the people participating often being unpredictable or at least highly stochastic. In addition, social interactions are grounded by a large number of (often implicit) norms, with expectations being highly dependent on the interaction context (Sherif, 1936). As such, it seems unlikely that every possible case of interaction and reactions from the humans interacting with the robot could be anticipated beforehand (Dautenhahn, 2004). But regardless of the complexity of the interaction, social robots need an action policy able to generate the appropriate reactions for the expected states and human behaviours. This action policy also needs to be able to manage uncertainty, selecting a correct action even when facing a sensory state not explicitly anticipated by the designers.

For the review in Section 2.2, the appropriateness of actions axis is a continuous spectrum characterising how much the system controlling the robot ensures that the robot acts in a safe and useful way for the users at any moment of the interaction. For example, a robot selecting its actions randomly has a low appropriateness as no mechanism prevents the execution of unexpected or undesired actions. On the other hand, a robot continuously selecting the action a human expert would select has a

high value as domain experts know of which action is the correct one in this application domain.

Adaptivity

As stated before, humans are complex, indeterministic and unpredictable agents, as such an optimal robot behaviour is not likely to be known in advance and programmable by hand (Dautenhahn, 2004; Argall et al., 2009). Specifically, end users will express behaviours not anticipated by the designers, the interaction environment is often not perfectly definable and the desired behaviour might also need to be customisable by or for the end user or evolve over time. While many studies in HRI use robots following a static script, to interact meaningfully outside of lab settings or scientific studies, robots need this flexibility to extend their range of application and improve their interactions with users. In other words, robots interacting with people need to be able to adapt their action policy to the environment 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.

We propose three components for this adaptivity. The basic one is the adaptivity to the environment, i.e. the generalisation of the behaviour (reacting accordingly to unseen inputs). The second one is personalisation and adaptation: being able to adapt a behaviour to the current user or context. Finally, the last component is the adaptivity in time which is in essence learning (the possibility to enrich and refine the action policy over time).

Generalisation: Robots are interacting in human centred environments which are complex and highly stochastic. These environments are often under specified and robot designers cannot explicitly anticipate every single possible human reactions or occurring events. Furthermore, the state representations often use large vectors with multiple possibilities for each values. As such, predefining a specific robot reaction for each state possibility or each possible human behaviour is not feasible. Consequently, robots should have an action policy able to generalise to unseen and unexpected situations and react appropriately to different environments.

Personalisation and adaptation: As robot are interacting with humans, they will encounter different type of environments, contexts of interactions and persons with

different roles. For example a robot used as an assistant for elderly people will have to interact in the home of the owner, but also follow them in the street or in a supermarket. In these different interaction contexts, distinct behaviour will be expected from the robot. Similarly, different humans being might have distinct roles and the robot needs to adapt its action policy to the type of person it is interacting with. For instance, in education, an autonomous would have to interact both with the students and the teacher, and its behaviour needs to take into account the role of the people it is interacting with. Additionally, the robot needs to personalise its behaviour to the person it interacts with: in entertainment or search and rescue, none of the users are known beforehand but providing a personalised behaviour adapted to the context may significantly impact the outcomes of the interaction. In summary, robots need to be able to adapt their actions policy to the environment and context they interact in and personalise their behaviour to the different users and their status.

Learning: When deployed in the wild, robots will be expected to interact over extensive periods of time with the same user, e.g. companion robots for the elderly, military robots for a squad or robots used in RAT (Leite et al., 2013). With these long-term social interactions, a key aspect in the engagement and efficiency is the co-adaptation between the user and the robot. Learning allows the robot to tailor its behaviour to the current user and track the changes of preferences and desires occurring over long-term interaction. Additionally, 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 people 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 additionally decreases the risks of confusion between these different communities. Finally, this adaptivity in time allow the robot to learn from its errors and improve its action policy over time. Furthermore, this learning can enrich the robot's action policy to allow it to tackle new task beyond its initial role, increasing its application and use.

In summary, 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 can generalise to unforeseen situations, dynamically changes its action policy according to the context of interaction and its partners, learns new actions and tasks and improves its action policy over time.

Autonomy

Today, many experiments in HRI are conducted using a robot tele-operated by a human (Riek, 2012; Baxter et al., 2016), and whilst having a human controlling the robot presents many advantages (e.g. the human provides 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). First, relying solely on tele-operation is not suited for deploying robots in the real world. Human control does not scale to interact for long periods of time or in many places: robots are expected to interact in fields already lacking workforce (e.g. healcare), so if robots needs to be continuously controlled, the advantage of automation is highly reduced. Second, the human-robot interaction tends to become "a human-human interaction mediated by a 'mechanical puppet' " (Baxter et al., 2016), which decrease the relevance of the robot as an agent and as such limits the knowledge gain for interactions with future fully autonomous robots. And finally, human control of a robot's actions introduces multiple biases in the robot's behaviour (Howley et al., 2014), and these biases from human operators will affect the robots' behaviour, decreasing the replicability of behaviours. For these reasons, among many, we argue that a robot used in social HRI should be as autonomous as possible. A limited human supervision could support the robot and be used to improve its behaviour, but the robot should not rely on humans for its action selection during the main parts of the interaction.

To analyse the different robot controller's autonomy, we take inspiration from Beer et al. (2014). Beer et al. define three components of autonomy: sensory perception, analysis and action selection. And autonomy is organised following a spectrum of different levels 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 and/or action: for example the robot can request information from a supervisor or the supervisor can override the action or goal being executed (Sheridan & Verplank, 1978).

Interdependence of factors

These three axes used for the review: appropriateness of action, adaptivity and autonomy are not independent. Especially, as a robot able to learn might be also able to improve its appropriateness of action and its autonomy as it refines its action policy. This impact of adaptivity on the two other axes is fundamental to increase the robot's fields of application, performance and usability. However, while a learning robot could eventually reach an optimal, perfect, and autonomous action policy, the behaviour expressed by the robot in early stages of the learning, while the action policy is not appropriate yet, is critical. Even during this learning phase, the robot's behaviour needs to be safe and useful for humans interacting with it. As such, adaptivity is a key element for a robot controller to improve and reach a correct and autonomous policy, but a mechanism must ensure that at every step of the interaction the robot's behaviour is appropriate regardless of the level of progress of the learning.

2.2 Current robot controllers in HRI

The previous section presented three axes to evaluate a robot controller: the appropriateness of actions, the range of adaptivity and the level of autonomy. Based on these three axes, this section presents and analyses high level categories of robot control currently used in HRI to define a robot behaviour. For each category, we will present the corresponding approach, indicate representative works done in this direction and qualitatively rate it on the three axes.

2.2.1 Scripted behaviour

One of the simplest ways to have a robot interacting with a human is probably to have explicit fixed behaviours. In this case, the robot is fully autonomous and follows a script for action selection. Success in using this approach is dependent on having a well defined and predictable environment to have the interaction running smoothly. However, if the interaction modalities (possible range of behaviours and goals) are limited enough, a constant action policy can be sufficient to handle all (sensible) human actions.

This approach is followed in a large number of research in HRI: as many studies are 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 behaviour with defined and controlled specificities. This also allows researchers to compare conditions

with controlled differences and analyse the impact of small variation of behaviour. Whilst this has advantages when exploring people's reactions to robots, this method can hardly be used to deploy robots to interact with humans on a daily basis. Real world applications take place in undefined and open environments where human potential behaviours are almost infinite. Additionally, a fixed robot behaviour might also create boredom in users once the novelty effect vanishes (Salter et al., 2004).

By essence, this type of controller has a no adaptivity as the robot is following a preprogrammed script, but is fully autonomous 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, robot can also be programmed to react in predefined ways to expected human actions. By adaptive preprogrammed behaviour, we denote a behaviour programmed before the interaction, but explicitly (or implicitly) including ways to adjust the action policy in reaction to anticipated human behaviours. This preprogrammed adaptation takes two forms: either having a fixed number of variables impacted by the actions of the partner and guiding the action policy (for instance using homeostasis), or explicitly planning for specific behaviours to be produced if predefined conditions are met (for example using a finite state machine).

Homeostasis, the tendency to keep multiple elements at equilibrium, is constantly used by living systems to survive and is also a good example of the first case of preprogrammed adaptation used in social HRI. For instance, Breazeal (1998) used 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 to represent a 'healthy' situation. If these variables reach values outside the desired homeostatic range, the robot is either over or under-stimulated, this will affect the robot's emotional 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).

On the other hand, a case of planned adaptation is clearly presented in Leyzberg et al. (2014). Participants have to play a cognitive game, and a robot delivers predefined advises on strategies depending on the performance and the current lack of knowledge of the participant. With these anticipated human behaviours, the robot can provide

personalised support as long as the participants behave within expectations.

Similarly to other behaviour-based methods used in robotic control (such as the subsumption architecture Brooks 1986), due to the indirect 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 fixed rules of actions limit the adaptability to unexpected event. Furthermore, with this indirect description of actions, there is no guarantee against the robot acting in inconsistent way in some specific cases which limits the appropriateness of actions. Similarly, planned adaptation provides adaptivity to the environment but only in highly limited cases expected by the designers. This limit the adaptability of such an approach as the robot does not learn and may face situations not expected by the designers, also reducing the maximum 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 adaptivity is low as they can only adapt to the environment within predefined, anticipated and limited 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 (Kelley, 1983) and is today common practice in HRI (Riek, 2012). Similar to scripted behaviours, Wizard-of-Oz (WoZ) is mostly used in human-centred studies to explore how humans react to robot and not as a realistic way to control robots in the wild. A second use of WoZ is to safely gather data to develop a robot controller from human demonstrations (cf. Section 2.2.4).

Even in WoZ, part of the robot's behaviour is autonomous, and combining this robot autonomy and human control can be done in multiple ways. Baxter et al. (2016) define two levels of WoZ related to the levels of autonomy presented by Beer et al. (2014) that correspond to the level of human involvement in the action selection process. Cognitive WoZ aims to provide a robot with human-like cognition or deliberative capabilities; while in perceptual WoZ, the human only replaces sensory system and feeds information

to the robot controller. Typically, perceptual WoZ replaces challenging features of the controller required for a study, but not relevant to the research question. One of such typical challenge is Natural Language Processing (NLP). Despite all the progress made in speech recognition, NLP is still a challenge in HRI, especially when interacting with children (Kennedy et al., 2017). And as some studies require a limited speech recognition element to test an hypothesis, using a human for that part of the interaction allows to run study without having to solve complex technical challenges (for instance, see Cakmak et al. 2010).

This level of human control impacts the autonomy of a system: a robot relying on human only to do perception has a higher autonomy than a robot fully controlled by an operator. Similar to the different levels of autonomy presented earlier, systems can also combine human control and predefined autonomous behaviour in mixed systems. For example, Shiomi et al. (2008) propose a semi-autonomous informative robot being mainly autonomous, but with 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.

With WoZ, the adaptivity and the appropriateness of actions 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 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 the human only has limited control over the robot's behaviour, no mechanism prevents the robot to make undesired decision, leading to a higher autonomous, but a lower appropriateness of actions and adaptivity compared to classical WoZ.

2.2.4 Learning from Demonstration

As stated by numerous researchers, explicitly defining a behaviour and manually implementing it on a robot can take a prohibitive amount of time or could not be possible at all (Argall et al., 2009; Billard et al., 2008; Dautenhahn, 2004). This statement applies equally well to manipulation tasks and social interaction. In both cases, humans have

some knowledge or expertise that should be transferred to the robot. However in social robotics, experts of the field often do not have the technical knowledge to implement efficient behaviours on a robot, which results in numerous design iterations 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. The learning process starts with a human demonstrations a correct behaviour (Argall et al., 2009), and then offline batch learning is applied to obtain an action policy for the robot. Later, if required, reinforcement learning can complement the demonstrations to reach a successful action policy (Billard et al., 2008). In LfD, the human-robot interaction is key, however in most of the cases this interaction is only in the learning process and the object of the learning is not social interaction with humans, but manipulation or locomotion tasks such as grasping and moving an object, using a racket to hit a ball or throwing tasks Billard et al. (2008).

However, two approaches have applied LfD to teach robots a social policy to interact with humans. The first one aims at learning directly from human-human interactions and analyse the human behaviours to replicate them on a robot. For example, Liu et al. (2014) present a data driven approach taking demonstrations from human-human interactions to gather relevant features defining human social behaviours. Liu et al. recorded motion and speech from about 180 interactions in a simulated shopping scenario and then clustered these behaviours into high-level actions. During the interaction, the robot uses a variable-length Markov model predictor to estimate probability of the human demonstrator to execute each actions and finally winner-take-all is applied to select the most probable action. 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 more human-human interactions in the real world, the performance should improve and become closer to natural human behaviours.

In the second approach, the data is collected through a WoZ setup and aims to learn to replicate the wizard action policy to reach an autonomous social behaviour. Knox et al. (2014) coined this approach “Learning from the Wizard (LfW)”. The method starts with a purely WoZ control study to gather data, and then, an action policy is derived by

applying machine learning on the collected data. However, this original paper presents no description of which algorithm could be used or how, and gives no evaluation of the approach, instead it 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 implementation details and results reduces the usability of the paper.

LfW is widely used in HCI (especially dialogue management; Rieser & Lemon 2008) and has also been implemented by other groups of researchers in robotics. For example, Sequeira et al. (2016) extended and tested this idea to a create a fully autonomous robot tutor. This method is composed of a series of steps:

1. Collect observations of a human teacher performing the task.
2. Define the different actions used by the teacher and the features used for the action selection.
3. Implement these actions and features on a robotic system.
4. Set up a restricted-perception WoZ experiment where an operator uses only the identified features to select actions for the robot.
5. Combine machine learning applied on the data and hand-coded rules to create an autonomous robot controller.
6. Deploy the autonomous robot.

(7. If required, add offline refinement steps to fine tune the robot's behaviour.)

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

Clark-Turner & Begum (2018) aimed 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 neurotypical participants to train the network with the raw inputs and the actions selected to obtain a controller able to deliver the therapeutic intervention. However, in their study,

the autonomous robot required limited human input to inform the algorithm of the state of the therapy and only reached a behavioural intervention with an accuracy inferior to 70%. This means that even with additional human input the robot would provide inconsistent feedback at some points in the interaction. However, this study only used a limited amount of data, and using more training examples should lead to better results.

LfD methods are based on real interactions either between humans, or between humans and robots controlled by humans; and, with enough demonstrations the robot should be able to select appropriate actions. However, efficiency is limited by the type of inputs recorded, the capabilities of the learning algorithm with the inputs space and the quality of the demonstrations which limit the appropriateness of the action policy (as seen in Clark-Turner & Begum 2018). Furthermore, after the learning phase, the robot's behaviour is mostly static, without any additional learning provided. As such, while possessing a good generalisation capability, LfD do not possess the adaptivity in time once the robot is deployed. Sequeira et al. (2016) propose to add offline learning steps could be added, but online learning would allow for smoother transitions and improvement of the behaviours. 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 low in the first phase and then total during the main part of the interaction.

2.2.5 Planning

An alternative way to interact in complex environments is to use planning (Asada et al., 1986). 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*, consist on acquiring 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 *act*, to execute the plan. The plan can be reevaluated at each time step or only if an encountered state differs from the expected one, in that case the robot updates its plan according to the new state of the environment and continues trying.

The efficiency of planning relies on having a precise and accurate set of pre and postconditions for each action. And as humans are complex and unpredictable, it is

a serious challenge, if not impossible, to model them precisely. As such, planning have seen limited use for open social interactions with humans. However, due to the nature of planning, reaching a specific goal in a known environment, it has been applied successfully to HRC. Additionally, limiting the interaction to a joint task also simplifies the modelling of the human: the interaction is more constrained and the human behaviour should limit to a number of expected task-related actions. The Human Aware Task Planner (Alili et al., 2009) is an example of planning used to assign task between a robot and human in a HRC scenario. One specificity of this planner is the ability to take into account predefined social rules (such as reducing human idle time) when creating a plan to allocate tasks to the human-robot team. Including these social norms in the plan construction is expected to improve the user experience and ensures maximised human compliance to the plan.

Planning performance depends heavily on the model of the environment the robot has access to. A precise and correct model ensures that the robot will autonomously select the appropriate action whilst an incorrect one would lead to non appropriate behaviours. 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 the tasks the robot has to complete the different contexts and states it is expected to face and as such presents limited generalisation capabilities to unanticipated situations or or non task-related human actions.

Planning has also been extended with learning, which then allows for more adaptive and personalised action policies. This type of learning planner has been mostly used in manipulation and navigation to obtain better trajectories (Jain et al., 2013; Beetz et al., 2004). In HRI, Munzer et al. (2017) presented a planner adapting its decisions to human preferences in a HRC scenario. With this approach, the robot estimates the risk of each actions and depending of the risk value will execute them, propose them (and waits for approval before executing an action), or wait for a human decision. Between repetitions of the task, the robot will update its planner to fit more precisely to the human preferences and improve its action policy for the next iteration of the task. Munzer et al. adopted principles from LfD to planning to improve quickly and efficiently the performance of the robot. However, while planning is well suited for strictly defined and mostly deterministic tasks, many social human-robot interactions cannot be totally specified symbolically and with clear actions and outcomes and as such the application

of planning to social HRI is limited.

2.2.6 Summary

Table 2.1 presents a summary of the different approaches currently used in social HRI with their advantages and drawbacks for application in HRI and the evaluation on each of the three axes. The two most promising types of control are LfD and planning, however, both of them have their drawbacks: LfD is applied offline to create a monolithic controller with limited adaptivity after being deployed, and planning's reliance on a model of the world limits its application in open-ended social HRI in the wild.

An ideal controller would learn how to interact by interacting, using demonstrations from an expert to obtain an initial reasonable action policy, but still improving itself after being deployed using reactions from the environment or feedback from a teacher. This type of interaction is similar to IML: learning from the interaction and using a human teacher to speed up the learning. Researchers have explored how to learn interactively non-social action policy from interactions with humans (Scheutz et al., 2017; Cakmak et al., 2010), but as of April 2018, no controller exists in HRI applying IML to the challenge of learning social interaction with humans.

A supervised learning from interaction would be the approach with the most potential as this type of learning can validate the three requirements: appropriateness of actions, adaptivity and autonomy. By essence, this continuous online learning aims at providing open-ended adaptivity to the robot. Including a human with control over the robot's actions can ensure that actions are appropriate. And finally, as the robot learns, accumulates datapoints and demonstrations from the teacher it improves its action policy, reducing the reliance and workload on the human to reach high levels of autonomy while conserving the constant appropriateness of actions and the adaptivity.

2.3 Interactive Machine Learning

As seen with example in LfD, Machine Learning (ML) is a promising method to provide a robot with an adequate action policy without having to implement in advance all the decisions rules used to select an action. ML has two main trends referring to the synchronisation between the learning and the use of algorithm: offline and online learning.

In robotics, offline learning is a technique allowing the robot to change its action policy

Table 2.1: Comparison of robot controllers in HRI

Controller	Advantage	Drawbacks	Application in HRI	Appropriateness	Adaptivity	Autonomy
Fixed preprogrammed behaviour	Quick and easy to create Clear specified and repeatable behaviour	Limited to highly constrained interactions	Human-centred studies in highly constrained env.	Low	Null	Maximal
Adaptive preprogrammed behaviour	Relatively simple to program More robust and efficient than scripted behaviour	Only provide adaptability in limited anticipated context	Human-centred studies in constrained environments	Medium	Low	Maximal
Wizard of Oz	Use human expertise to select the best action	Require constant high workload from human	Human-centred studies Highly critical HRI	Maximal	Maximal	Null/Low
Learning from Demonstration	Transfer knowledge from human to robot in the real application environment	Lack of learning once deployed	HRI case by case	High	Medium	High
Planning	Complex behaviours and adaptable to variations in the environment	Human are too complex to have clear set of conditions Limited application to social interaction	Complex defined environments HRC	Medium	High	Maximal

over time by updating it outside of the interaction (such as Learning from the Wizard in Section 2.2.4). Between or before the interactions, a learning algorithm is used on a dataset previously accumulated to create a new action policy.

On the other hand, online learning (such as Reinforcement Learning (RL); Sutton & Barto 1998) learns during the interaction. Rather than single monolithic definitions or updates of the behaviour, this constant refinement of the agent behaviour benefits from a high number of updates, allowing the robot to learn even during the first interaction and never stop improving its behaviour.

Interactive Machine Learning (IML), as coined by Fails & Olsen Jr (2003), is a type of online learning with two specificities:

- Use of an end-user expert in the learning process.
- Learn by multitude of consecutive small updates of behaviour.

These two characteristics differ greatly from classical offline learning, such as deep learning (LeCun et al., 2015) which uses costly monolithic learning steps without human influence to define a static behaviour. On the other hand, IML is an iterative process where the behaviour is improved at each small steps, and where the end-user can provide feedback on the learner's performance during all these iterations. IML aims to learn faster, by continuously using the human expert to correct the errors made by the algorithm as they appear, provide additional useful information to the learner and improve the knowledge gained at each learning step.

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. As explained in Fails & Olsen Jr (2003), classifiers gain to use human knowledge to iterate quickly to reach a good solution and agents learning from the interaction would gain from using additional feedback from humans (Thomaz & Breazeal, 2008; Knox & Stone,

2009). IML aims to combine the advantages from both Supervised Learning (SL) and online learning and applies this new type of learner to classification or interaction tasks.

Furthermore, by allowing a human user to see the output of an algorithms and provide additional inputs, the learning has the potential to be faster and tailored to this human's desires. By using human expert knowledge and intuition, the system can achieve a better performance faster ((Thomaz & Breazeal, 2008)). Additionally, a key advantage of IML is also being able to empower end-users of robotic or learning systems. 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.

These human inputs take three forms: labels for specific datapoints (cf. active learning - Section 2.3.2), feedback over actions (similarly to reward in RL; cf. Section 2.3.3) or demonstrations to reproduce (cf. LfD - 2.2.4).

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 learning process (Johnson et al., 1991). This approach has been transferred to machine learning, especially to classifiers, by allowing the learner to ask questions and 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 numbers or costly to obtain. As such, a trade-off arises between the performance of the classifier and the quantity of queries made by the algorithm. Often, the oracle would be a human annotator with the ability to provide a correct label to any datapoint, but their use should be minimised for reasons of cost, time or annoyance.

Using an oracle to provide the label of specific points with high uncertainty should highlight missing features in the current classifier resulting in improvements both in term of accuracy and learning speed. However, this specific relation between the learner and the human teacher raises new 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 this relation between the learner and the oracle. This research has been especially active in HRI with robots directly asking questions to human participants and exploring how the robot's queries could inform the teacher about the knowledge of the learner (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. On the other hand, 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. However, 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 proactive 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 (Settles, 2009). However, when interacting in the world, the learner is not in control of which sample can be submitted to an oracle to obtain a label. Datapoints are provided by the interaction and are influenced by the learner's actions and the environment reaction. For agents learning during the interaction, the active learning approach working for classifiers is not applicable, so other methods have been applied such as RL, learning from human feedback or LfD.

2.3.3 Reinforcement Learning

The main framework of learning applied to learning from interaction is Reinforcement Learning (RL). RL aims to solve the problem of finding the best action policy by observing the environment reaction to the agent's action.

Concept

Young infants and adults learn by interacting with their environment, by producing actions and receiving a direct sensory motor feedback from their environment. By learning the impact of their actions, humans can learn how to achieve their goals. Similarly, the field of 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 agents interact in a discretised version of the time, considering life as a sequence of states and actions. The simplest version of RL is modelled as a finite

Markov Decision Process (MDP), a discrete environment defined by the five ensembles $(S, A, P_a(s,s'), R_a(s,s'), \gamma)$ (Howard, 1960), 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_* (assigning an action from A to each state in S) maximising the discounted sum of future rewards. The agent is not aware of all the parameters of the model, but only observes the transitions between states and the rewards provided by the environment and has to update its policy to maximise this cumulated reward. Different algorithms exist to reach this policy, but the main features present in all of them are the concepts of exploration and exploitation (Sutton & Barto, 1998).

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 rapidly collect knowledge on the environment and then decrease this probably to converge toward an efficient 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 (Sutton & Barto, 1998). 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.

Application to HRI

This approach presents many features relevant to HRI: it possesses the autonomy required for meaningful interactions with humans and provides the adaptivity desired for having a large impact. However, 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 iterations before reaching an efficient action policy. Generally, RL copes with these issues by having the agent interacting in a simulated world. This allows the agent to explore safely in an environment where its actions have limited impact on the real world (only time and energy) 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 of time and random behaviours would have direct impact.

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 robots are often stiff and strong or cause distress. This reliance on random exploration presents a clear violation of the first principle to interact with humans presented earlier ('Only execute appropriate actions').

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

Despite this real-world constraints, RL has been used in robotics (Kober et al., 2013), but mostly applied to manipulation, locomotion or navigation tasks. For the reasons stated above, as of early 2018, RL has never been used to autonomously learn social behaviours for HRI.

Opportunities

Despite the limitations presented in the previous section, changes can be made to RL to increase its applicability to HRI. Combining RL and IML ensures that the behaviour is appropriate to interactions with humans even in the learning phase.

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 achieves a minimal acceptable performance. The 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 are also applicable to other machine learning techniques to make them safer (for instance by using LfD Billard et al. 2008).

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 or by having a list of safe actions (Alshiekh et al., 2017). Using this method, the anticipated cases of errors can be prevented. However it seems unlikely that every case could be specified in advance. As such, an efficient way to prevent the execution of undesired actions could be to include a human expert in the action selection loop in the early learning phase, and giving them the capacity to preempt undesired actions before their execution.

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. For example, Abbeel & Ng (2004) proposed 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 a reward function are learned, 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, and these small variations ensure that policies leading to incorrect behaviours are 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 to learn social behaviours or to have robot interacting with humans.

2.3.4 Human as a source of feedback on actions

When an agent is learning in a RL fashion and improves its behaviour by receiving rewards from the environment, an intuitive way to steer the agent's 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 (Bouton, 2007). In ML, using rewards from a human to bias and improve the learning presents advantages: the interface is simple and generalisable to any type of problem, and the teacher only needs a way to provide a scalar or a binary evaluation of an action to steer the learning. However, this simplicity of interaction is associated with a limited efficiency and a complexity of interpretation: the issues of how to interpret human rewards and how to combine them with environmental ones if existent are an active research field today (Knox & Stone, 2010).

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 robotics as a clear reward function applicable to HRI or robotics in general can be complex to define. 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 relations and executed actions in the environment. Users of LambdaMOO could either reinforce positively or negatively Cobot's action by providing rewards. While the interaction between the agent and the users was limited, Isbell et al. presented the first agent to learn social interactions with humans in an online complex and social environment.

While the goal of Cobot was to create an entity interacting with humans, Knox & Stone (2009) explored how humans could actively teach an agent an action policy in the absence of environmental rewards using TAMER (Training an Agent Manually via Evaluative Reinforcement). With this approach, 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.

However, unlike environmental rewards, human rewards are subjective evaluations of an agent's behaviour. As such by knowing humans tendencies and intentions when providing rewards, an agent is able to obtain more information from these human rewards than by treating them the same way as environmental ones. Many researchers explored

how to obtain more information from human reward. For instance, Advice (Griffith et al., 2013) models how trustworthy the teacher is and as such how much importance the learner should give to their rewards. For example, rewards from inconsistent teachers will be reduced as the agent should only provide limited attention to them. Alternatively, Loftin et al. (2016) explored how to infer the strategy used by the teacher in the reward delivery. Similar behaviours from different teachers might have different meaning: not rewarding an action might reflect an implicit acknowledgement of the correctness of an action or the active refusal to provide a positive reward (indicating the incorrectness of an action). By modelling this intention, the real meaning of rewards can be inferred and used to further improve the learning. A last relevant feature explored by this community is the dependence in time of the human reward policy. While reward functions are generally constant in time with RL, with humans they might vary according to the current performance of the agent. 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, when the average agent's performance is better. MacGlashan et al. (2017) proposed COACH (Convergent Actor-Critic by Humans) to model how humans adapt their rewarding scheme in function of the agent's performance and deal with this non-stationary reward function. Similarly to other factors biasing human rewarding strategies, this dependence of the reward function to the current agent's policy should be taken into account to maximise the knowledge gained from human rewards.

Even when environmental rewards are present, human rewards still have opportunities to improve the learning: they can enrich a sparse reward function, guide the robot faster to an optimal policy or correct incomplete or incorrect environmental rewards. Knox & Stone (2010) explore the impact of nine different ways to combine these two types of rewards and the impacts on the learning of each methods. And, from these analysis, they explain how to select an approach according to the specificities of the environment and the reward function.

Teachers can also use rewards to communicate other information to the learner. For example, Thomaz & Breazeal (2008) aimed to explore how humans would use feedback to teach a robot how to solve a task in a virtual environment. They used Interactive Reinforcement Learning (IRL) as a way to directly combine environmental rewards and human ones. However, during early studies, Thomaz and Breazeal discovered that

participants tried to use rewards to convey intentions, 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 channel to provide information about the action the robot should execute. This guidance has been actively decided to be ambiguous; participants could not explicitly control the robot, but just bias the exploration, and adding this second channel improved the performance of participants. This study presented a first attempt to combine environmental rewards, human ones and human guidance to teach an agent an action policy and demonstrated the importance of giving additional ways for the teacher to impact the robot's behaviour.

While not being applied to robotics but mostly to learning non-social interactions, these implementations of IML provide important research describing how robots could be taught to interact with humans. These human rewards are especially interesting when the environmental reward function is sparsely defined or non-existent, providing a way to teach robots in any environments. However, humans do not simply evaluate an agent's actions, they adopt teaching strategies influencing their way of rewarding and want to provide guidance, hints or commands to help the agent to learn better and faster. In summary, human teachers desire to go beyond simply evaluating what the robot is doing by providing advices or commands about how it should behave.

2.3.5 Interactive Learning from Demonstration

As presented in Argall et al. (2009) and Billard et al. (2008), LfD is majoritarily used in an offline learning fashion to learn a defined task without extending the action policy once the task is considered mastered. However, tasks such as social interaction are complex even for humans and probably will never be fully mastered for robots; as such (and as argued before), robots would highly profit from learning throughout all their life, not only once before being deployed, but learning new tasks and improving their skills as often as required (Dautenhahn, 2004).

With Interactive Learning from Demonstration (ILfD), an agent receives demonstrations not only once, but as often as required after being deployed. ILfD is related to Mixed Initiative Control (Adams et al., 2004) where an agent and a human share control on the agent's actions. The robot acts mostly autonomously, but in some cases (at the initiative of the human or the robot), the human takes over the robot control and make a demonstration that will be used by the robot to refine its action policy for the future.

One approach giving teachers a total initiative on the interaction is Dogged Learning (DL) (Grollman & Jenkins, 2007). With DL, an agent is autonomously interacting and a teacher has the power to override the agent behaviour at any time by selecting desired actions or outputs. Facing a potential difference between the algorithm's outputs and the teacher's ones, the robot executes the commands with highest confidence (often the human's one) and the learning component aims at reproducing the executed output. If the teacher does not provide any commands, the ones from the algorithm are used. DL does not provide the robot with the opportunity to request a demonstration, but instead, the robot can communicate its uncertainty to the teacher, indirectly asking for demonstrations.

Chernova & Veloso (2009) propose a method with a more complex interaction between the learner and the teacher. The Confidence Based Algorithm (CBA) is composed of two components: the Confident Execution (CE) and the Corrective Demonstration (CD). The CE enables the agent to act autonomously when its confidence in its action policy is high and on the other hand to actively request a demonstration when the confidence is low. The CD allows the teacher to provide a corrective demonstration when the agent executes an incorrect action, which provide more information to the agent than a classic negative reward. These two components aim to leverage the complementary capabilities of the learner and the teacher. CBA has demonstrated efficient teaching in diverse scenario such as simple driving simulators or other classification tasks. But the effectiveness of this approach is bounded by the capacity of the learner to estimate this confidence to be able to request demonstrations and prevent incorrect behaviour to be executed. Another limit of such an approach is the impossibility for the teacher to correct undesired actions before they negatively impact world.

Both methods rely on the teacher being able to anticipate the robot's behaviour to provide demonstrations before an incorrect action is executed or before it impacts the agent and its environment. As such, the appropriateness of the robot controller is not at maximum as the teacher cannot ensure that no incorrect action will be executed during the learning, only that the robot would learn faster from its errors.

2.3.6 Importance of control

Results from active learning, research using human to provide feedback and LfD have shown that human teachers take an active stance during the training of an agent and

want multiple ways to influence the learner's behaviour (Amershi et al., 2014). Humans are not oracles, 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 improving the teacher's experience in the teaching process, providing the humans with more control improves the learning (Thomaz & Breazeal, 2008; Chernova & Veloso, 2009). By allowing the teacher to demonstrate an action policy online, bias the action selection and preempt or correct undesired actions, the learner interacts mostly in useful states of the environment and with a correct action policy. This lead to faster learning and would improving the robot's performance highly in early stages of the learning. Another fundamental feature added by this human control over the robot's actions is safety. If a domain expert can prevent a robot interacting with humans to make errors and can ensure that all its actions are efficient, the quality of the interaction for the humans involved is greatly increased. This will further improve the applicability and use of the robot and would satisfy the two first principles: appropriateness of actions and adaptivity of the robot.

However providing the teacher with this control presents challenges for designing the interaction. Unlike a simple scalar reward, being able to control the robot requires the teacher to be able to give commands or advice to the robot and to receive additional information about the learner beyond its observable behaviour. This enriched two-way communication might be complex to design, especially when the action space is bigger than a few actions or the learning mechanism not transparent. In addition to the communication interface, the time scales of the interaction are also key: to give the opportunity to the teacher to preempt undesired actions, the learner needs to communicate its intentions in a timely manner to the teacher which complexifies the relation between the learner and the teacher.

2.4 Summary

This chapter presented first an overview of fields of HRI where robots interact socially with humans. From these cases of application, we defined three principles 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 and learn.
3. Have a high level of autonomy.

Secondly, a review of current controllers 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 ML with potential to satisfy these principles. IML shows promises for enabling a robot to learn online how to interact with humans, especially when the teacher is given control over the robot's behaviour and can demonstrate a correct action policy. However while humans have been used to teach robots behaviours or concepts, teaching them to interact with human in an interactive, online fashion has not been demonstrated in the field so far and would satisfy all these requirements.

Chapter 3

Supervised Progressive Autonomous Robot Competencies

Key points:

- Proposition of a novel interaction framework to teach robots an action policy while interacting.
- A human teacher is in control of the robot's actions whilst the robot learns from this supervision.
- The robot proposes actions to be executed to the teacher.
- The teacher provides feedback on intentions rather than actions.
- The robot's behaviour (under supervision) can be assumed to be optimal.
- The 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:

- http://dx.doi.org/10.1007/978-3-319-25554-5_60.
- https://doi.org/10.1016/j.patrec.2017.03.015.

As presented in Chapter 2, robots would profit from being able to learn from human teachers how to interact with other humans. We propose to use Interactive Machine Learning (IML) to achieve this transfer of social and task knowledge from the human domain-expert to the robot. This would result in a faster and safer learning than slow iterative update of behaviours by engineering the action policy, learning from large quantities of data or learning by trials and errors as with Reinforcement Learning (RL).

However, as stated in this Section 2.3, IML has never been applied to teach robots to interact with humans. No current system provides the teacher with enough control over the robot's actions to validate the first principle presented in Section 2.1.2 ('Only execute appropriate actions'). Techniques relying solely on feedback from the teacher cannot prevent the robot to execute an incorrect action, but only reduce the chances of future errors by rewarding negatively incorrect actions after their execution (Senft et al., 2017a). And, with current techniques based on Learning from Demonstration (LfD) the teacher relinquishes its control over the robot when not demonstrating, only reacting in hindsight after the learner makes mistakes and its erroneous actions have impacted the real world (Chernova & Veloso, 2009).

The problem tackled in this research is to provide a robot with an appropriate action policy, adaptive to different contexts and partners' behaviours and requiring a low workload on the teacher. As such, in Senft et al. (2015), we introduced the Supervided Progressive Autonomous Robot Competenties (SPARC) framework of interaction. SPARC aims to allow end-users to safely and easily teach a robot an action policy applicable to social Human-Robot Interaction (HRI).

For this chapter, the terms 'agent', 'robot' and 'learner' will be used interchangeably. Whilst SPARC has been developed to allow robots to learn how to interact socially with humans, it could also be applied to any type of agent learning to interact in an environment. Similarly, the terms 'supervisor' and 'teacher' are exchangeable as the teacher teaches the robot how to behave while supervising its behaviour.

3.1 Frame

Similarly to other applications of IML, SPARC requires inputs from 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 (as shown in Figure 3.1). This results into two intertwined interactions: the application interaction (task the robot learns to achieve)

and the teaching interaction (relation with the teacher). In the generic case, the overall interaction is a triadic interaction (Teacher - Robot - Human target or Teacher - Robot - Environment); for instance, a teacher could teach a tutor robot to support child learning in an education task (as implemented in Chapter 6). But in specific cases, the overall interaction can be only dyadic (Teacher - Robot), such as a robot at home learning from its user how to support them better.

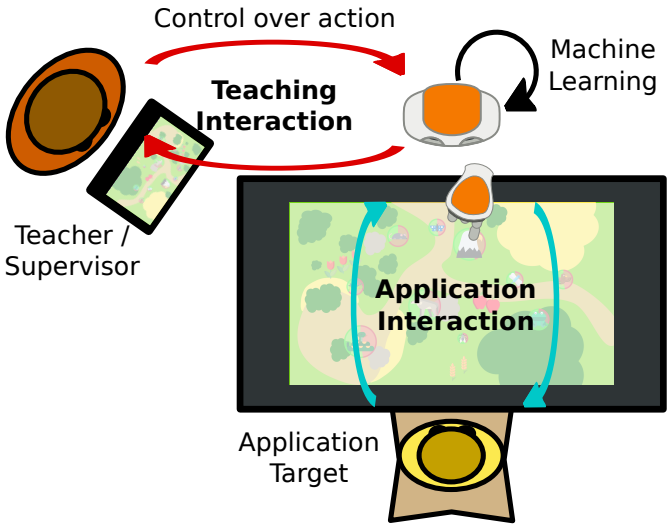


Figure 3.1: Frame of the interaction: a robot interacts with a target, suggests actions and receive commands and feedback from a teacher. Using machine learning, the robot improves its suggestions over time, to reach an appropriate action policy.

3.2 Principles

SPARC defines an interaction between a learner (virtual agent or robot) and a teacher following these principles:

- The learner has access to a representation of the state of the world and a set of actions.
- The teacher can select actions for the robot to execute.
- The learner can propose actions to the teacher before executing them (informing them about its intentions).
- The teacher can enforce or cancel actions proposed by the learner and actions non evaluated are implicitly validated and executed after a short delay.
- The learner uses Machine Learning (ML) to improve its action policy using the teacher's commands and feedback on propositions.

This type of interaction between the learner and the teacher is similar to the level 6 on the Sheridan scale of autonomy: "computer selects action, informs human in plenty of time to stop it" (Sheridan & Verplank, 1978); with the addition that the human has also the opportunity to select actions for the agent to execute. In this thesis, we will refer to this interaction as 'Supervised Autonomy': the robot interacts autonomously under the supervision of a human who can ensure that the robot's behaviour is constantly appropriate.

This way of keeping a human in the learning loop, with the opportunity to override the agent actions, and the robot learning from these demonstrations is similar to Dogged Learning (Grollman & Jenkins, 2007). However, with SPARC, this ability to provide demonstrations is combined with the Supervised Autonomy. This results in a mixed control system where the teacher can select actions and have the robot execute them while the robot only proposes actions to the teacher. In response to this suggestion, the teacher has the choice between pre-empting the action or let it be executed. A learning algorithm on the robot side uses the feedback and commands from the teacher to improve the correctness of the suggested actions until reaching an efficient action policy. This learning mechanism coupled with auto-execution of actions aims at decreasing the requirement of interventions from the teacher over time, thus reducing the workload on the teacher as the robot learns. Additionally, keeping the human in the loop also gives them the opportunity to provide additional information to the algorithm speeding up the learning. The diagram presented in Figure 3.2 and the flowchart in Figure 3.3 present in a graphical way this interaction between the learner and the teacher.

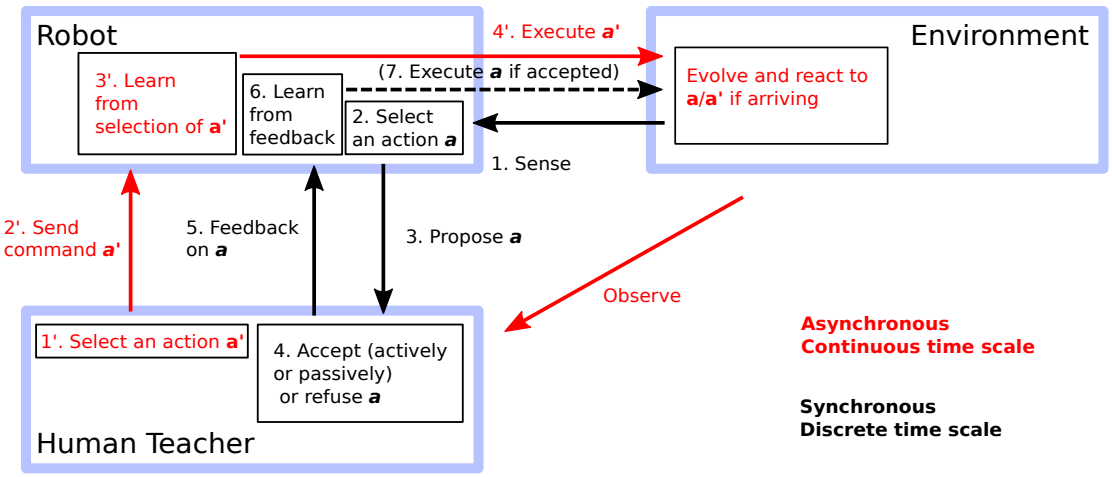


Figure 3.2: Diagram of interaction between the robot, the human teacher and the environment with SPARC: synchronously, the robot can propose actions to the teacher that evaluated (approved or cancelled). And asynchronously, the teacher can select actions to be executed.

Additionally, The main difference between SPARC and CBA (Chernova & Veloso, 2009) is that with SPARC, the robot communicates its intentions and the teacher has total control over the robot's action. With CBA and other classical IML, the teacher have to wait for an action to impact the world before correcting it or providing negative feedback (Thomaz & Breazeal, 2008; Knox & Stone, 2009). However, using SPARC, the teacher is informed beforehand of the robot future actions and can pre-empt them before they impact the world. The agent learns to avoid actions with expected negative impact without having to face the results of the execution of these undesired actions. This implies that the behaviour executed by the robot can be assumed to be optimal, making the interaction safer and potentially simplifying the learning mechanism.

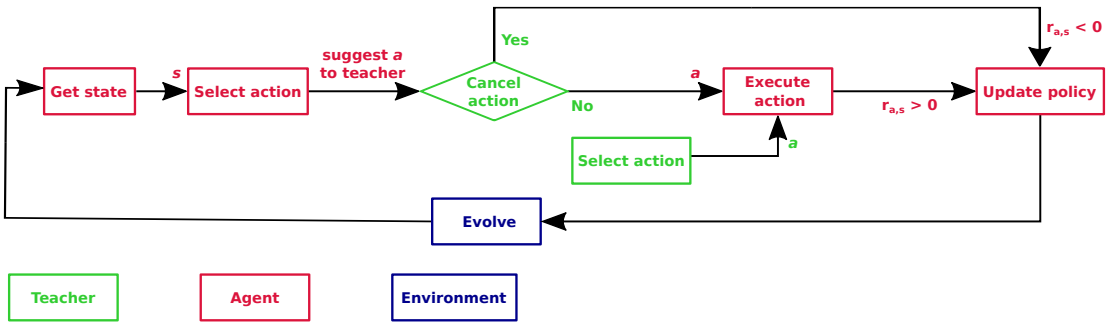


Figure 3.3: Flowchart of the action selection loop between the agent and the teacher.

This approach is comparable to predictive texting as seen on phone nowadays. The user can select the words proposed by the algorithms (or implicitly accept them by pressing the space bar) or write their own. The algorithm learns the user's preferences and habits and aims to suggest words more and more appropriate for the user. However, predictive texting aims mostly to correct users' errors and interact in static environments. On the other hand, SPARC aims to replicate a teacher's action policy in continuous time, in a dynamic and interactive environment evolving both dependently and independently of the agent's actions.

Alternatively, SPARC can be seen as a way to provide pro-activity to an agent. By observing interactions on longer time scales, such as an assistant robot at home, SPARC allows the robot to propose to help its user when the current state is similar to previous observation. This would compare to a passive case where each action executed by the robot has to be requested by the user or an autonomous robot interacting in the house without any transparency. By proposing actions to the teacher, the robot takes the initiative to support humans, while not imposing its presence. In human environment, executing actions (such as starting to play music, changing the lighting condition or

cleaning the kitchen) without informing the surrounding humans could be perceived as rude or annoying if the timing is not right. On the other hand, this pro-activity also needs to be kept in control as a robot proposing to help too often might be equally annoying.

3.3 Goal

SPARC aims to provide an interaction framework to teach robots an action policy possessing the following characteristics:

- Be usable by people without expertise in computer science.
- Allow fast policy learning from in-situ guidance.
- Require few or no human input for the robot to act in the world.
- Ensure a constantly appropriate robot behaviour.

Figure 3.4 presents an idealistic comparison of the expected workload, performance and autonomy of four methods: autonomous learning (e.g. RL; Sutton & Barto 1998), feedback based teaching (e.g. TAMER; Knox & Stone 2009), Wizard-of-Oz (WoZ) (Riek, 2012) and SPARC. Unlike other learning methods, by following the principles presented in Section 3.2, SPARC is expected to maintain a constant high performance even during early stages of learning. In later stages of the learning, the agent keeps improving its action policy, making its suggestions more accurate and allowing the auto-execution of actions to reduce the workload on the teacher.

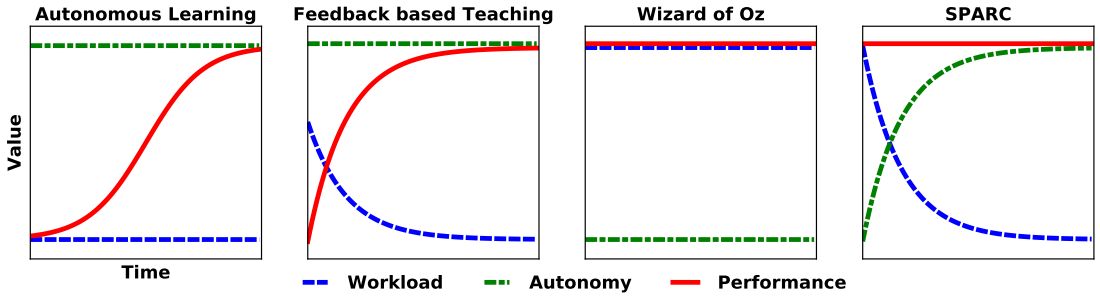


Figure 3.4: Idealistic evolution of workload, performance, and autonomy over time for autonomous learning, feedback-based teaching, WoZ and SPARC (arbitrary units).

Once the behaviour is deemed appropriate enough by the teacher, the agent is ready to be deployed to interact autonomously in the real world, if this outcome is desired. Alternatively, in contexts where a human expert still cannot be removed from the control loop, such as Robot Assisted Therapy (RAT), a supervisor stays in control of the robot's

actions in using the Supervised Autonomy. Similarly to the teaching phase, with this Supervised Autonomy, the robot informs the supervisor of its actions and the human only has to intervene in case of incorrect propositions. While still requiring attention from the supervisor, this reduction of human actions to control the robots aims at reducing the workload on the supervisor. Furthermore, as the supervisor accumulates a better knowledge of the agent's behaviour, they can be especially careful in cases where the agent is prone to making errors. And as the control of the agent requires less effort from the supervisor, they can focus more closely on the application interaction rather than the teaching one.

3.4 Implications

3.4.1 Relation with time

Similarly to other IML approaches, the requirement of a human in the action selection loop limits the time scale of interaction and this effect is an important consideration when applying SPARC to an interaction. With this type of IML, three time scales are coexisting: the robot's, the teacher's and the interaction's. The robot has an internal clock running probably multiple times per second, sensing the world and deciding if an action should be proposed. The human teacher has to be able to cancel proposed actions before their executions, so they need to be provided with a 'correction window' spanning more than one second to react to propositions. And finally, for some applications, actions are appropriate only during a short time, and if this amount of time is shorter than the correction window, action executed automatically would not be appropriate anymore, reducing the validity of SPARC.

As mentioned previously, depending of the application, actions might have to be executed in a time critical environment (such as driving) or less critical ones (such as an assistant robot at home). While being easily applicable to slow interactions, SPARC could still be applied to these time critical ones. For instance, if applied to autonomous driving, SPARC could display to the driver the planned trajectory with augmented reality and let the driver correct this trajectory if not appropriate. However, more critical elements, such as emergency breaking, would probably have to be done with a much shorter correction window so the car can break in time.

Additionally, the presence of this correction window reduces the rate of actions selection to 0.5 Hz or below, which might reduce the application of SPARC compared to other

IML methods. However, this limitation of application is the price to pay to ensure the appropriateness of actions, and this effect can be mitigated by using higher level actions or by focusing on applications less critical in term of time.

Finally, the rate of selection of high level actions will be much lower than the rate of the robot's action selection loop. At a human level, actions will be executed at a rate of few actions per minutes, while the robot's processing runs at multiple hertz. This indicates that unlike classical RL methods, in most of the steps, the robot should not select any action. When interacting with humans, the learning algorithm and the state representation needs to take into account these differences of time scales to ensure that the robot's behaviour is coherent and useful.

3.4.2 Relation with LfD

SPARC is an Interactive Learning from Demonstration (ILfD) method (cf Section 2.3.5) and as it uses human demonstrations of policies to learn, it presents many similarities with non-interactive LfD techniques (cf. Section 2.2.4). 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; Munzer et al., 2017) and never in an online fashion. Munzer et al. proposed an interactive planner that would learn offline the current user's preferences and desires, but two key differences exist between this approach and SPARC. The first one is the application domain. The learning planner is well suited to clearly defined environments (e.g. Human-Robot Collaboration (HRC)) where a similar task with clear steps has to be done multiple times. As such the learning can happen offline between the repetitions. SPARC is defined to be applicable to non-linear underspecified environments, with less constrained tasks, where the learning should happen online. Secondly, using a threshold, Munzer et al. define actions with low risk which are executed (and can be cancelled during the execution) and actions with higher risk which have to be validated first by the human. SPARC does not make this distinction, but proposes both types of actions to the teacher and will start executing them if no feedback is received. This removes the need for the human to explicitly approve correct proposed actions while ensuring that the human can cancel incorrect actions before their execution. This principle aims at reducing the number of required interventions from the human to teach and interact with the robot

compared to methods such as the one presented by Munzer et al.

3.4.3 Interaction with Machine Learning Algorithms

The principles of SPARC define it at the crossroads between Supervised Learning (SL) and RL. SPARC can be used in two ways: either to reproduce a teacher's policy in a supervised fashion or to discover a useful action policy using the teacher to bias the exploration, limiting errors and only interacting in desired parts of the environment.

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 SL or RL. This research presented in this thesis explored combinations with three types of algorithms: supervised learning using a feed-forward neural network (Chapter 4), reinforcement learning (Chapter 5), and supervised learning using an instance based algorithm (Chapter 6). However SPARC could be combined with a wide range of other algorithms or techniques such as planning.

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

3.5 Summary

This chapter introduced Supervided Progressive Autonomous Robot Competenties (SPARC), a novel interaction framework to teach agents an action policy. This approach is suited to teach a robot to interact with humans as it validates the principles defined in Section 2.1.2 (appropriateness of action, adaptivity and autonomy). SPARC starts in a similar fashion to WoZ: the teacher selects actions at any time to be executed by the robot. Then, using a learning algorithm, the agent starts to propose actions to the teacher who can let them be executed after a short time or cancel them if not appropriate. This mechanism combining selections, suggestions, and evaluations of actions ensure the appropriateness of actions as a human expert could have pre-empted any inappropriate action before their execution. This additionally provides the adaptivity as the teacher can extend the behaviour beyond the current action policy. Finally, the learning algorithm associated with the auto-executions of actions ensures that once the robot starts to learn, the human workload decreases; and, when an acceptable

action policy is reached, the robot is ready to be deployed to interact autonomously if this outcome is desired.

Chapter 4

Relation with Wizard of Oz

Key points:

- Design of an experiment 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 the target behaviour.
- Design of a robot model exhibiting probabilistic behaviour (simulating a child) with a non-trivial optimal interaction policy.
- Results from a within subject study involving 10 participants 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

The Supervided Progressive Autonomous Robot Competenties (SPARC) as been designed to enable end-users without expertise in computer science to teach a robot an action policy while interacting in a sensitive environment, such as Human-Robot Interaction (HRI). By using machine learning and Supervised Autonomy, SPARC intends to allow a field expert to progressively transfer their knowledge to an autonomous agent without having to enforce each action manually. Additionally, as the agent is interacting in the target environment, displaying an appropriate action policy, the time spent to teach it is not lost but used to deliver the desired interaction even during the learning phase. For example, in the context of Robot Assisted Therapy (RAT), a therapist would teach the robot during a therapy session. And, as the therapist is in total control of the robot's action, the behaviour expressed by the robot always fits the desired goals for the therapy. This ensures that even the sessions used to teach the robot have a therapeutic value for the patient involved in the therapy.

SPARC, as a principle, allows to start a robotic application in a Wizard-of-Oz (WoZ) fashion and then move away from it as the robot gains autonomy. The aim of SPARC is twofold: maintaining a high level of performance in the target application while reducing the workload of the teacher over time. As the robot learns, the action policy is refined until reaching a point where the robot is autonomous or only necessitates minimal supervision to interact successfully.

As explained in Chapter 3, SPARC involves two interactions: the teaching interaction and the application one. As such, when the goal of the teaching is interacting with humans, the robot interacts simultaneously with at least two humans (the target(s) and the teacher). These two dependent interactions add complexity to the evaluation of the approach, especially as both humans are impacting each other.

The first step to evaluate SPARC was to focus on the teaching interaction, the relation between the robot and its teacher. To evaluate this aspect of the interaction, we decided use the context of RAT for children with Autism Spetrum Disorder (ASD). However, as the presence of two humans decrease the repeatability of the test bench, we replaced the child involved in the therapy by a robot running a model of a child. The setup ends up with two robots interacting together: the *child-robot* completing a therapeutical task and the *wizarded-robot*, controlled by a participant, supporting the child-robot in its task

completion. Actions from the wizarded-robot impact the child-robot's behaviour and to achieve a high performance in the task, the child-robot needs to receive an efficient supporting policy from the wizarded-robot. As such, the child-robot's performance is used as a proxy to evaluate the performance of the participant in the supervision. This environment with a single human-robot interaction allows us to observe and evaluate the impact of SPARC on the teaching interaction.

4.2 Scope of the study

The study presented in this chapter intends to evaluate if the learning component of SPARC allows participants to teach an efficient action policy for a robot interacting with humans. For repeatability concerns, the human-robot interaction target of the learning has been modelled by two robots interacting together. The control condition is a variation of WoZ, where participant still control a robot but without the learning component. By combining learning and Supervised Autonomy, SPARC aims to allow the teacher to maintain a high performance during the interaction while reducing the workload on the teacher over time.

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

H1 The child-robot's performance is a good proxy for the teacher's performance.

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

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

H4 Using SPARC allows the teacher to achieve similar performance than WoZ but with a lower workload.

H1 represents a validation of the model, ensuring that the child-robot performance represents the efficiency of the action policy applied by the teacher. H2 tests that human teachers are not static entities, they adapt their teaching target and their interaction strategy. H3 tests one of the motivations behind SPARC: does reducing the number of physical actions from a human to control a robot while requiring the teacher to monitor the robot suggestions lead to a lower workload. Finally, H4 is the main hypothesis, does

SPARC enables a robot to learn a useful action policy: reducing the teacher's workload while maintaining a high performance.

4.3 Methodology

4.3.1 Participants

The study involved 10 participants (7M/3F, age *M*=29.3, 21 to 44, *SD*=4.8 years). While SPARC is expected to be usable by anyone, regardless of their knowledge of computer sciences, this first study involved members of a robotic research group assuming the role of the robot supervisor. This decision is supported by the fact that in RAT scenarios, the wizard is typically a technically competent person with significant training controlling this robot for this therapy. As such, as the participants come from a population expected to assume this type role, the results of the study maintain their applicability to HRI.

4.3.2 Task

This study is based on a real scenario for RAT for children with ASD based on the Applied Behaviour Analysis therapy framework (Cooper et al., 2007). The aim of the therapy is to help a child to develop/practice their social skills. The child has to complete an emotion recognition task by playing a categorisation game with a robot on a mediating touchscreen device (Baxter et al., 2012). The robot can provide feedback and prompts to encourage the child and help them to classify emotions. In the task, images of faces or drawings are shown to the child on the touchscreen, and the child has to categorise them by moving them to one side of the screen or the other depending on whether the picture shown denotes happiness or sadness. In real therapies, the robot would be generally remote-controlled by an operator using the WoZ paradigm Riek (2012).

This study explores if SPARC can be used to teach the robot a correct action policy to support the child in this therapy scenario. As timing in human-robot interactions is complex, for simplification reasons, the interaction has been made discrete to have clear steps when the robot has to select an action. During these action steps, the selection of an action is decided by following the principles defining the Supervised Autonomy:

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 after a short delay.

3. The robot executes the selected action.

4. Both the robot and the teacher observe the outcome of the action until the next action selection step.

This study compares two conditions: SPARC, where the robot learns from the human selections and the WoZ condition where the robot is simply controlled by the participant. As mentioned before, in both conditions, actions can only be executed in predefined time windows dictated by the dynamics of the interaction. Additionally, a 'wait' action is present in the SPARC condition and presents an active choice for the participants. In a real WoZ scenario, participants would have to enforce every single action made by the robot, however, a 'wait' action in that context does not make sense. As such, in the WoZ condition, the robot proposes random actions, thus increasing the probability of having the teacher correcting the suggestion leading up to a setup similar to a real WoZ.

As mentioned earlier, the focus of the study being on the teaching interaction (the relation between the teacher and the robot), the second interaction (the application) 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 supervised by the human teacher is the *wizarded-robot* (cf. Figure 4.1).

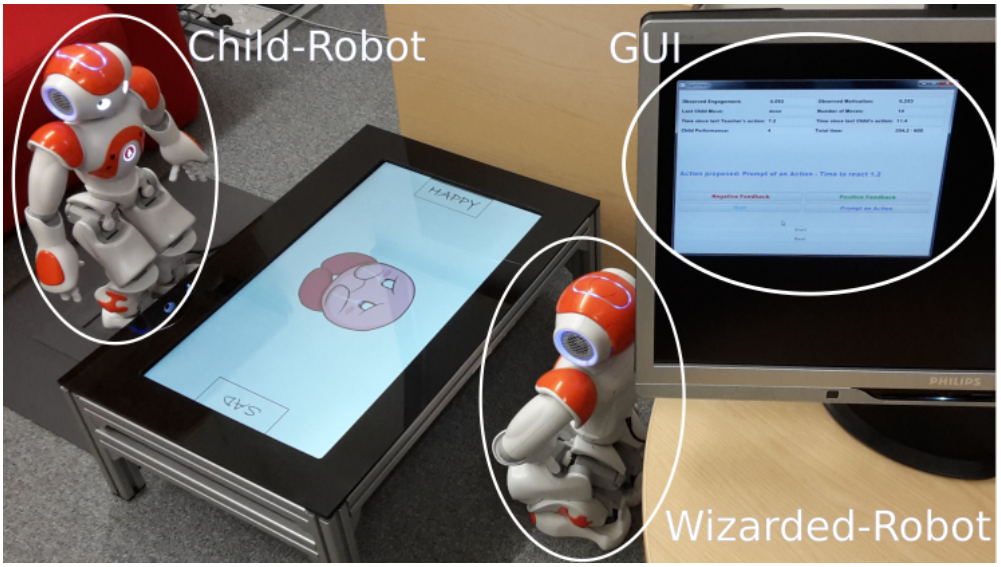


Figure 4.1: Setup used for the user study from the perspective of the human teacher. The child-robot (left) stands across the touchscreen (centre-left) from the wizarded-robot (centre-right). The teacher can oversee the actions of the wizarded-robot through the Graphical User Interface (GUI) and intervene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

4.3.3 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 these form the state of the child. The engagement represents the involvement of the child in the task, i.e. how often the child-robot will make categorisation moves. And the motivation relates to the seriousness of the child in solving task; in the model, the motivation gives the probability of success of each categorisation move.

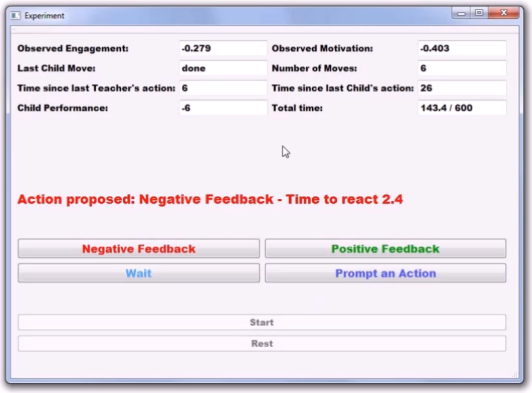
These states are bound to the range [-1, 1] and influenced by the behaviour of the wizarded-robot. Values of 1 indicate that the child-robot's behaviour is positive, it is involved in the task. Values of -1 show that the child-robot is actively refusing to participate. And a 0 represents a neutral state where the child-robot is neither especially involved nor actively disengaged. To represent a tendency to return to a neutral state of mild engagement, both states asymptotically decay to zero with no 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.

As explained in Section 4.3.4, the wizarded-robot's action impact the child-robot state: congruent action will tend to increase engagement and motivation. However, if repeated, actions can lead to frustration for the child-robot. If a state is already high and an action from the wizarded-robot should increase it further, there is a chance that this level will sharply decrease. 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 combines feedback actions and waiting ones to maintain the state values high but prevent them from overshooting. This non-trivial optimal action policy approximates better a real human-robot interaction scenario requiring a more complex strategy to be expressed by the robot.

4.3.4 Wizarded-robot control

The wizarded-robot is controlled through a GUI (shown in Figure 4.2) and has access to the variables defining the state of the interaction used by the learning algorithm:

- Observed engagement.
- Observed motivation.
- Type of last categorisation made by the child-robot (good/bad/done).



	+ive feedback	-ive feedback
Good move	E↑ M↑	E↓ M↓
Bad move	E↑ M↓	E↓ M↑
No move	E↓ M↓	E↓ M↓

Prompt an Action: E↑ M=

Wait: E= M=

Engagement is linked to the rate of *child robot* categorisations and its gaze behaviour

Motivation is linked to the success of *child robot* actions and its speed of movement on the touchscreen

Figure 4.2: Screenshot of the interface used by the participants, the GUI on the left allows to control the robot and a summary of the actions' impact is displayed on the right.

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

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

The wizarded-robot has a set of four actions it can execute, each represented by a button 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 4.3.3, if the engagement or the motivation exceeds a threshold, their value can decrease abruptly to simulate the child-robot being frustrated. This implies that the optimal action policy consist on providing congruous feedback and prompts, but also requires wait actions to prevent the child-robot from becoming frustrated and maintain its state-values close to the threshold without exceeding it. A 'good' strategy keeping the engagement and motivation high leads to an increase in performance of the child-robot in the categorisation task.

As introduced previously, to simplify the algorithm part, the interaction has been discretised, the teacher cannot select 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 categorisation or if nothing happened for five seconds since the last wizarded-robot's action. 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. The teacher can only select an action in reaction to a proposition from the wizarded-robot; alternatively, 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 intervene by selecting a different action and forcing the wizarded-robot to execute it.

4.3.5 Learning algorithm

In the SPARC condition, the robot learns to reproduce the action policy displayed by the teacher. For this study, the algorithm used for learning is 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 are set to -1. The network is fully retrained with all the previous state-action pairs and the new one between each selection step.

This learning algorithm, MLP, is not optimal for a real time interaction as the online learning should happen quickly between learning iterations. However, as the length of interaction (and so the number of datapoints) is limited, the network can be retrained between two consecutive uses. Finally, the desired learning behaviour being purely supervised learning, this type of algorithm has been deemed suitable for this study.

4.3.6 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, and the order of interaction have been balanced between participants to control for any ordering effects. In the order S-W the participants first interact with the learning wizarded-robot in the SPARC condition, and then with the non-learning one in the WoZ condition; and in the order W-S, this interaction order is inverted (starting with WoZ then SPARC). 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. The participant, 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. After the information was read, a 30s video presenting the GUI in use was shown to participants to familiarise them with the interface, 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. Similarly, in the second part of the experiment, the participants interacted with the other condition and completed a second post-interaction questionnaire. Finally, a post-experiment questionnaire asked participants to explicitly compare the two conditions. All questionnaires and information sheet are available online[2].

4.3.7 Metrics

Two types of metrics 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, auto-execution and selection of another action (intervention). The states of the child-robot (motivation, engagement and performance) were also recorded at this step.

The first metric is the performance achieved by participants in each interaction. As the policy applied by the participants cannot be evaluated directly, the performance of the child-robot in the task (number of correct categorisations minus number of incorrect categorisations) is used as a proxy for the participant performance. H1 evaluates if this approximation is valid by analysing the relation between the performance of the child-robot and the value of its inner states. If a correlation is found, it would demonstrate that a good supervision policy (managing to keep the engagement and the motivation of the child-robot high) leads to a high performance. As such, this child-robot performance

[2] https://emmanuel-senft.github.io/experiment-woz.html

represents how efficient the action policy executed by the wizarded-robot was 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 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 questionnaire where they had to compare the two conditions. All the rating questionnaires used seven item Likert scale. For clarity for participants, in the questionnaires the wizarded-robot is named 'teacher-robot'.

Post-Interaction questions:

- The child-robot learned during the interaction.
- The performance of the child-robot improved in response to the teacher-robot's 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 questions:

- 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 4.3 presents the aggregated results (collapsed between orders) for the performance and the final 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 interventions (intervention ratio: SPARC: 0.38 (95% CI [0.29,0.47]) - WoZ: 0.59 (95% CI [0.52,0.67])).

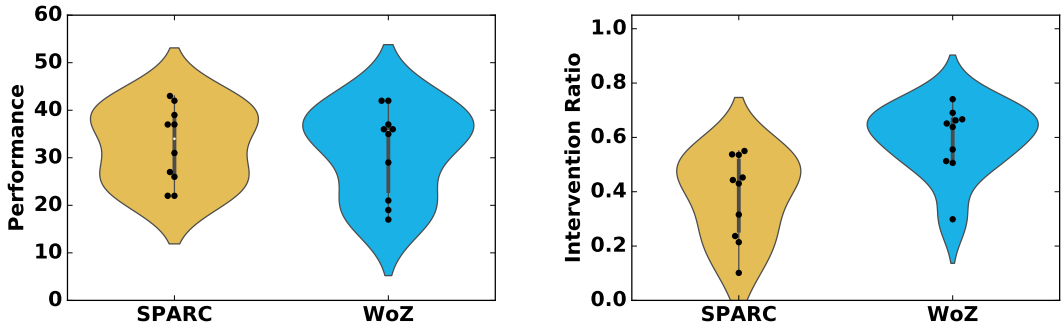


Figure 4.3: Aggregated comparison of performance and final intervention ratio for both conditions. Dots represent individual datapoint (N=10 per condition) and shaded area the probability distribution most likely to lead to these points.

Figure 4.4 presents the evolution of intervention ratio for each condition and orders. During the first interaction, participants discovered the interface and how to interact with it, which resulted 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 had developed their teaching policy, there is a tendency for SPARC to require 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. This would indicate that the two conditions differ in term of required interventions.

For both the performance and the intervention ratio, a strong ordering effect was observed. Figure 4.5 and Table 4.1 present the performance and final intervention ratio separated by condition and order. In both orders, the performance in the second interaction is higher performance as the participants were used to the system and developed an efficient interaction policy. On the other hand, the performance between

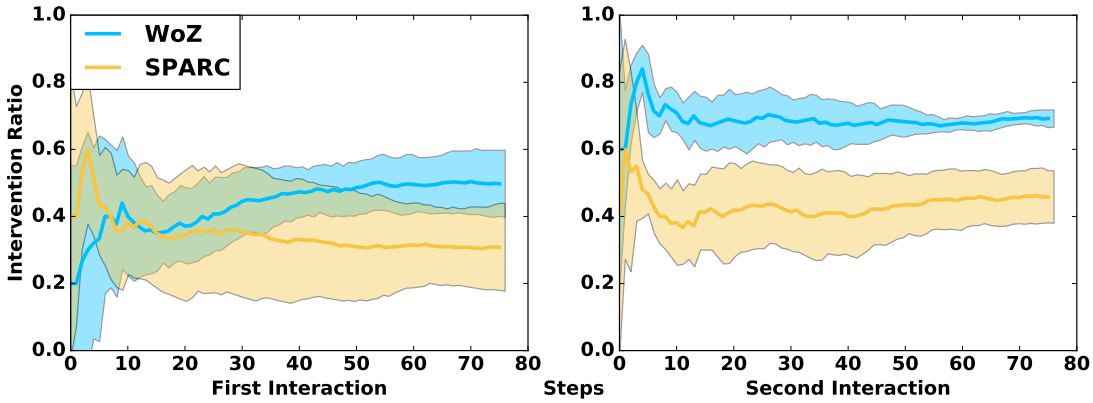


Figure 4.4: Evolution of intervention ratio over time for both conditions and both orders. Shaded area represents the 95% CI.

condition for the same interaction number is similar (in both their first and second interactions, the condition of interaction did not impact the performance). However, for both orders, when comparing between condition for the same interaction number, 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.

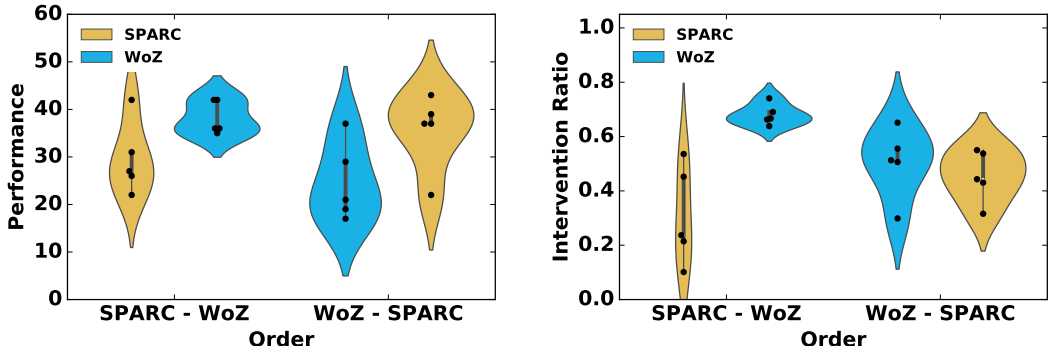


Figure 4.5: Performance achieved and final intervention ratio separated by order and condition. For each order, the left part presents the metric in the first interaction (with one condition) and the right part the performance in the second interaction (with the other condition).

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 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,

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

	Order S-W		Order W-S	
	SPARC (int 1)	WoZ (int 2)	WoZ (int 1)	SPARC (int 2)
Performance M	29.6	38.2	24.6	35.6
95% CI	[23.6,35.6]	[35.5,40.9]	[18.1,31.1]	[29.3,41.9]
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]

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 4.6.

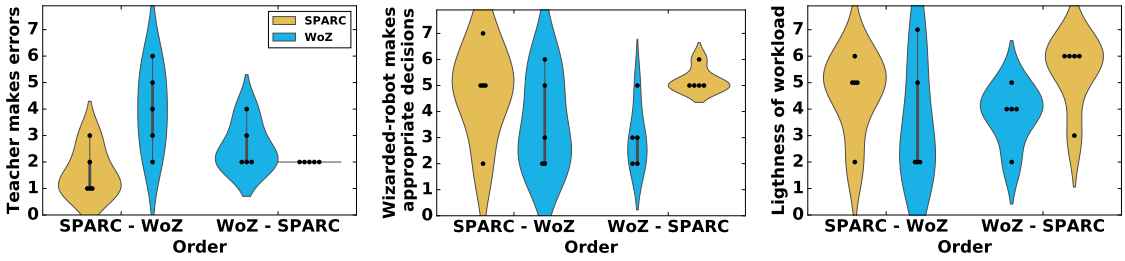


Figure 4.6: Questionnaires results on robot making errors, making appropriate decisions and on lightness of workload.

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 in the WoZ condition but the error margin is too high to conclude anything.

Participants rated the wizarded-robot as more suited to operate unsupervised with SPARC than with WoZ (95% Confidence Interval of the Difference of the Mean (CIDM) for S-W ordering [-0.2, 2.6], CIDM for the W-S ordering [1.6, 2.8]).

Similarly, a trend was found showing that the wizarded-robot with SPARC is perceived as making fewer errors than with WoZ (CIDM for S-W ordering [1.3, 3.4], CIDM for the W-S ordering [0.1, 1.1]).

The participants tended to rate the workload as lighter when interacting with SPARC, and this effect is much more prominent when the participants interacted with the WoZ

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

	Order S-W		Order W-S	
	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	[3.7,6.7]	[3.8,6.6]	[4.2,6.2]	4.7,6.1]
Child's performance M	4.6	5.0	5.0	4.4
95% CI	[3.4,5.8]	[3.3,6.8]	[4.0,6.0]	[3.7,5.1]
Wizarded-robot makes errors M	1.6	4.0	2.6	2.0
95% CI	[0.9,2.3]	[2.8,5.2]	[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]	[2.2,5.0]	[2.0,4.0]	[4.9,5.6]
Lightness of workload M	4.6	3.6	3.8	5.4
95% CI	[3.4,5.8]	[1.8,5.4]	[2.9,4.7]	[4.4,6.5]

first (CIDM for S-W ordering [-0.6, 2.6], CIDM for the W-S ordering [0.7, 2.5]).

Most of the difference of mean interval exclude 0 or include it marginally, which would indicate tendency of difference, but due to the low number of participants, no statistical tests are applicable and as such no significance can be demonstrated.

4.5 Discussion

Strong support for H1 (a good teacher leads to a better child performance) was found, a correlation between the average value of states (engagement and motivation) and the final performance for all of the 10 participants was observed (r=0.79). This validity check confirms that the performance of the child robot reflects the performance of the teacher in this task: supervising the wizarded-robot to execute 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. For instance, the ratio of waiting action compared to

other supportive actions varied between participants.

H3 (reducing the number of interventions reduces the perceived workload) is partially supported: the results show a trend for participants to rate the workload as lighter when interacting with the SPARC, and another trend between using SPARC 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 cause of the workload is probably the discovery of the system and how to interact with it rather than the requirement to manually select actions for the robot. Nevertheless, regardless of the order of the interactions, SPARC consistently received higher ratings for lightness of workload and required fewer interventions to be controlled which indicates that using SPARC could decrease workload on robot's supervisor compared to WoZ.

Finally, H4 (using learning maintains similar performance, but decreases the workload) is supported: interacting with a learning robot in the SPARC condition results in a similar performance than interacting with a non-learning robot in the WoZ condition, whilst requiring fewer active interventions from the supervisor and a lower workload to control. Reducing the workload on the robot operator has real world utility, for example, in the context of RAT, it might free time for the supervisor to allow them to focus on other aspects of the intervention, such as analysing the child's behaviour rather than solely controlling the robot.

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 algorithms and ways to handle time have been used in the following studies presented in Chapters 5 and 6.

4.6 Summary

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 the learning component of SPARC 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 both conditions allowed the participants to reach a good performance, with SPARC, the presence of learning

shifts part of the burden of selecting actions onto the wizarded-robot rather than on the human. Using SPARC, the robot partially learnt an interaction policy which decreases the requirement on the teacher to physically enforce each robot's actions. This indicates that a learning robot could reduce the workload on the operator freeing them to do more valuable tasks and that SPARC could be an efficient interaction framework to operate this learning. In addition to providing a robot with autonomy, this reduction of workload has real world implications, in the context of RAT, it 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:

- Design of an experiment comparing SPARC and another interactive teaching method: IRL.
- The application domain is a replication of the world used in early studies evaluating IRL.
- IRL uses partial guidance to the robot and explicit rewarding of the robot's action to teach it a policy.
- 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 from a mixed design study involving 40 naive participants show that SPARC achieves a better performance and an easier and faster teaching 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 using Qt.

5.1 Motivation

Previous work in Interactive Machine Learning (IML) showed that humans want to teach robots not only with feedback on its actions but also by communicating the robot what it should do (Thomaz & Breazeal, 2008). However, in most research where agents are taught policies using human guidance, the teacher is given little or no control over the agent's actions and has to observe the agent executing an action even when knowing that this action is incorrect (cf. Section 2.3.4). This chapter explores how these IML approaches could be improved by applying the principles of Supervided Progressive Autonomous Robot Competenties (SPARC) defined in Chapter 3. This chapter also presents experimental results demonstrating how these principles influence the learning process, the agent performance and the user experience and how these results compare to other traditional IML approaches.

The study presented in Chapter 4 explored how SPARC could be used with Supervised Learning, to replicate a teacher's action policy. However, some of the most promising features of IML arise when combined with Reinforcement Learning (RL) as it might allow an agent to learn beyond the demonstrations (Abbeel & Ng, 2004). 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. This chapter presents results from a study involving 40 participants comparing the teaching efficiency and user experience of SPARC to Interactive Reinforcement Learning (IRL), another IML approach offering less control but having been validated in previous studies (Thomaz & Breazeal, 2008).

5.2 Scope of the study

5.2.1 Interactive Reinforcement Learning

IRL implements the principles presented in Thomaz & Breazeal (2008): a human supervises and teaches an agent to interact autonomously in an environment. This teaching is achieved by providing guidance and positive or negative feedback on the last action executed by a robot. In this study, the algorithm controlling the robot combines this human feedback with environmental ones to form a reward used to update a Q-table. This Q-table assign a Q-value (interest of taking an action) to every state-action pair and is used to select the next action. Three additions to the standard interaction mechanism

have been proposed and implemented by Thomaz and Breazeal and are used in this study as well: guidance, communication by the robot and an undo mechanism (Thomaz & Breazeal, 2008).

The guidance channel emerged from the results of a pilot study where participants assigned rewards to objects to indicate that the robot should do something with them. With the guidance channel, teachers can direct the attention of the robot toward certain items in the environment, informing the robot that it should use them in its next action. This guidance behaviour offers partial control over the robot's actions restricting the executable options, but cannot be used to explicitly set the robot's behaviour.

Additionally, the robot communicates uncertainty by directing its gaze toward different parts of 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 the robot is unsure about its next action and that guidance should be provided.

Finally, the undo mechanism aims at providing a way for the teacher 'cancel' the robot's action, to bring it back to relevant part of the world, in order to speed up the teaching. After receiving a negative reward, the robot tries to cancel the effect of the previous action on the environment (if possible), resulting in an undo behaviour. As shown in Thomaz & Breazeal (2008) (hereafter the 'original study' or 'original paper'), these three additions improve the robot's performance on the task and the user experience.

In summary, Teachers have two ways to transmit information to the robot: a reward channel (providing a numerical evaluation of the last action) and a guidance channel (directing the robot's attention toward parts of the state to restrict the exploration).

5.2.2 SPARC

SPARC uses a single type of input from the human similar to the guidance in IRL but no reward channel. However with SPARC, the guidance channel directly controls the actions of the robot. The robot communicates all of its intentions (i.e the action it plans to execute next) to its teacher by looking at a part of the environment. Following the principles proposed in Section 3.2, 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 enforced or passively approved by the teacher.

5.2.3 Differences between IRL and SPARC

Unlike IRL, SPARC offers the user full control over the actions executed by the robot. SPARC changes the learning paradigm from learning from the human's evaluation of actions' impacts to learning from the human's knowledge and the *expected* impact of actions. An expert in the task domain evaluates the appropriateness of actions before their execution and can guide the robot to act in a safe and useful manner. This implies that the robot does not rely on observing the negative effects of an action to learn to avoid it (as in IRL), but rather it learns 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 should be able to stop the robot from ever executing them, preventing the robot from causing harm to itself or its social or physical environment.

Another noticeable difference is the type of information the robot communicates with the user: in IRL, the robot communicates its uncertainty about an action and with SPARC its unambiguous intention to execute an action. Similarly, the communication from the user to the robot differs between the two approaches. 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 and has to manually provide feedback to evaluate the robot's decisions. A result is that the quantity of information provided by the user to the robot is similar for both IRL and SPARC.

5.2.4 Hypotheses

Three hypotheses have been tested in the study:

H1 *Effectiveness and efficiency with non-experts.* IRL and SPARC will have differences in performance, speed, number of inputs used, mental effort on the teacher and the number of errors during the teaching phase when used by non-experts. We predict that for each metric, SPARC will lead to better results.

H2 *Safety with experts.* SPARC can be used by expert users (knowledgeable in

the interaction process) to teach an action policy safely, quickly and efficiently, achieving better results than 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

5.3.1 Participants

A total of 40 participants (age *M*=25.6, *SD*=10.09; 24F/16M) were recruited using a tool provided by the University of Plymouth to reach a mixed population of students and non-student members of the local community[2]. At the start of the experiment, all participants gave written informed consent, were told of the option to withdraw at any point and completed a demographic questionnaire. 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: not at all familiar to 5: extremely familiar). The study lasted around one hour and followed a 2x2x3 mixed design where participants interacted with both conditions three times. To avoid ordering effects, the order of interaction was counterbalanced between two groups: group 1 interacting with IRL then SPARC and the interaction order is inverted for group 2 (see also Figure 5.2). Participants were distributed randomly between the two groups whilst balancing gender and age. All participants received remuneration at the standard U.K. living wage rate, pro rata.

In addition to naive non-expert users, an expert user (the author) interacted five times with each system following a strictly optimal strategy for both conditions. 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 in robot interaction, such as therapists in the context of assistive robotics.

5.3.2 Task

The task used in this study is the same as Thomaz & Breazeal (2008): "Sophie's kitchen", a simulated environment presented 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

[2] `https://uopsop.sona-systems.com/Default.aspx?ReturnUrl=%2f`

online version of the task[3].

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) (see Figure 5.1a). The participant has to teach Sophie to bake a cake by guiding it through a sequence of steps while giving enough feedback so the robot learns a correct series of actions leading to the completion of the task. There are six crucial steps to achieve a successful result:

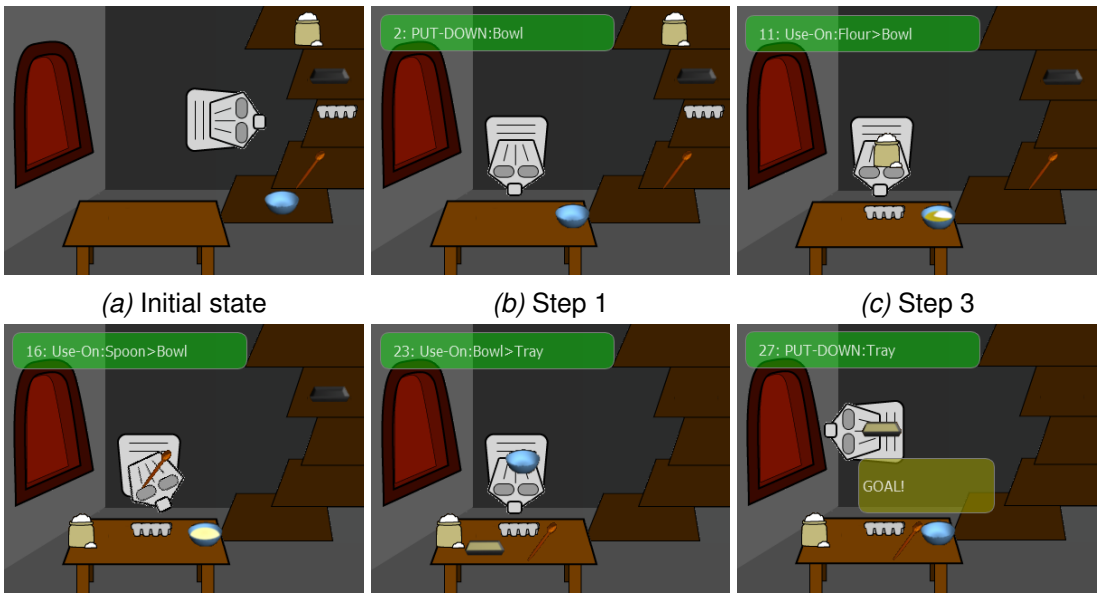
1. Put the bowl on the table (Figure 5.1b).
2. Add one ingredient to the bowl (flour or eggs).
3. Add the second ingredient (Figure 5.1c).
4. Mix the ingredients with the spoon to obtain batter (Figure 5.1d).
5. Pour the batter in the tray (Figure 5.1e).
6. Put the tray in the oven (Figure 5.1f).

The environment is a deterministic Markov Decision Process (MDP), defined by a state, a set of actions (move left, move right, pick up, drop and use), a deterministic transition function and an environmental reward function. The environment includes end states, corresponding to a success or a failure, which reset the simulation to the initial state and provide a reward (+1 for success and -1 for failure). All the other states corresponding to intermediate steps have a reward of -0.04 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. This environment includes a large number of possible states (more than 10,000), success and failure states and a sparse environmental reward function. These elements increase the value of having a teacher present to support the learning. As argued by Thomaz and Breazeal in the original paper, this environment provides a good setup to evaluate methods of teaching a robot.

5.3.3 Implementation

Two conditions were constructed to compare the IRL and SPARC approaches to this task. The underlying learning mechanism is identical in both conditions. The only

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



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

differences lie in the manner of interaction (inputs to and from the algorithm) and the amount of control over the robot's actions. With IRL teachers have to explicitly provide rewards and have a partial control over the action selection, while with SPARC rewards are implicit and the control over actions is total.

The learning algorithm (see Algorithm 1 and 2) is a variation on Q-learning, without reward propagation[4]. This guarantees that any learning by the robot is due to the human's teaching, and as such provides a lower bound for the robot's performance. By using Q-learning, the robot's testing performance would be higher.

As shown in Table 5.1, another difference between the conditions is that with SPARC, the algorithm learns immediately after executing an action (and only with positive rewards). On the other hand, IRL learns about an action just before executing the next one, based on a positive or negative evaluation received between the actions.

Interactive Reinforcement Learning

We have implemented IRL following the principles presented in Thomaz & Breazeal (2008). The user can use the left mouse-click to display a slider providing rewards. Guidance is implemented by right-clicking on objects to direct the robot's attention

[4] In Q learning the update function is $Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha(r_t + \gamma(\underset{a}{max}Q(s_{t+1},a)) - Q(s_t,a_t))$

Table 5.1: Simplified outline of algorithms used for both condition.

Algorithm 1: SPARC

while *learning* **do**
- a_t = $\underset{a}{argmax} Q[s_t, a]$
- look at object or location used in a_t
- **while** *waiting for command (2 seconds)* **do**
  - **if** *received command* **then**
    - a_t = received command
    - $r_t = 0.5$
  - **else**
    - $r_t = 0.25$
- Act in the world:
  - execute a_t, transition to s_{t+1}
- $r_t = r_t + r_{environment}$
- Learn:
  - $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_t + \gamma(\underset{a}{max} Q(s_t, a)) - Q(s_t, a_t))$

Algorithm 2: IRL

while *learning* **do**
- $A_{t+1} = [a^1 ... a^n]$, n actions with high $Q[s_{t+1}, a^i]$
- **while** *waiting for guidance and reward on a_t (2 seconds)* **do**
  - **if** $n > 1$ **then**
    - indicate confusion
  - **if** *received reward r'_t* **then**
    - $r_t = r_t + r'_t$
  - **if** *receiving guidance* **then**
    - **if** *guidance acceptable* **then**
      - a_{t+1} = guidance
- Learn:
  - $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_t + \gamma(\underset{a}{max} Q(s_t, a)) - Q(s_t, a_t))$
- Act in the world:
  - execute a_{t+1}, transition to s_{t+2}
- $r_{t+1} = r_{environment}$

toward a specific object. Guidance can only be provided for objects the robot is facing, otherwise right-clicking has no effect. Following a guidance message, 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 one location to another. This guidance gives a partial opportunity to the user to limit the exploration for the current step, without preventing the robot to explore in further steps.

Some modifications to the original study were required due to the lack of implementation details in the original paper, one of them being the use of a purely greedy policy instead of using softmax. As, the presence of human rewards and guidance limits the importance of autonomous exploration, the greediness of the algorithm should assist the learning by preventing the robot from exploring outside of the guided policy.

It should be noted that the presence of the human in the learning process alters deeply the concept of convergence. By providing rewards, the teacher can manually force the robot's policy to converge or diverge.

SPARC

SPARC uses the gaze of the robot toward objects or locations to indicate to the teacher which action the robot is suggesting. Similarly to the guidance in IRL, the teacher can

use the right click of the mouse on objects to send a 'command' to the robot and have it execute the action associated to this object in the current state. However, in this condition, this communication has been extended to also cover locations. With SPARC, the command covers the whole action space: at every time step, the teacher can specify, if desired, the next action to be executed by the robot. Similarly to the guidance, this command can be used on objects only if the robot is facing them. If a robot's suggested action is not corrected, a positive reward of 0.25 is automatically received (as it has the implicit approval from the teacher). If the teacher selects another action, a reward of 0.5 is given to the selected action (the corrected action is not rewarded). That way, actions actively selected are more reinforced than the ones accepted passively and participants still have access to a wider range of rewards with IRL. This system allows for the use of reinforcement learning with implicit reward assignation, aiming to simplify the teaching interaction.

5.3.4 Interaction protocol

Participants were divided into two groups and interacted with both IRL and SPARC, with the order of presentation being counterbalanced between groups (see Figure 5.2). Participants in group 1 interacted with IRL first for three sessions and then with SPARC for the three remaining sessions; and the interaction order was inverted for participants in group 2.

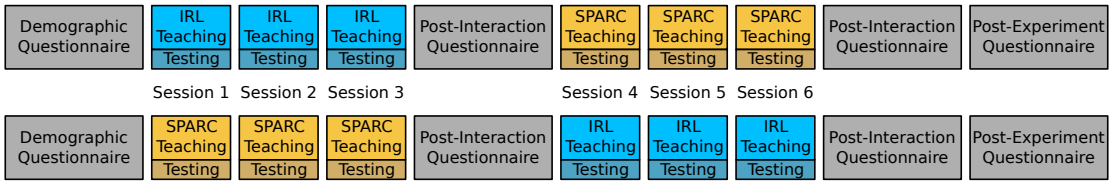


Figure 5.2: A representation of the timeline experienced by participants according to the order they were in. The top row corresponds to group 1 and bottom row to group 2.

After welcoming participants and before interacting with a system, participants completed a demographic questionnaire and received two information sheets. The first one explained the task (describing the environment and how to bake the cake) and the second one described the system they would interact with (IRL or SPARC).

After reading the sheets, participants interacted for three sessions with the system they were assigned to. Each session started with a teaching phase where the participants could interact with the robot to teach it to complete the task. The teaching phase was

composed of a number of episodes, corresponding each to a trajectory from the initial state to an end state (success or failure) after which the environment was returned to the initial state. Similarly to the experiment by Thomaz and Breazeal, participants could decide to terminate the teaching phase whenever they desired by clicking on a button labelled 'Sophie is ready'. But the teaching phase was automatically terminated after 25 minutes to impose an upper time-limit on the study.

After the teaching phase, the robot ran a testing phase where the participant's inputs, other than a force stop, were disabled. The test stopped as soon as an ending state was reached or the participant forced a stop (e.g. if an infinite loop occurs). This testing phase's aim is to evaluate the participants' performance in the teaching task. The interaction with each system involved three repeated independent sessions with their own teaching and testing phases. This aims to observe how the interactions evolved as participants get used to interact with a system.

After participants completed their three sessions with the first system, they were asked to complete a first post-interaction questionnaire. Then they received the information sheet for the second system, interacted with it for three sessions and completed a second post-interaction questionnaire.

At the end of the experiment, participants completed a last questionnaire, the post-experiment questionnaire, received the financial compensation and were explained the goal of the study. All information sheets and questionnaires can be found online [5] and the questionnaires are described in Section 5.3.5.

5.3.5 Metrics

Interaction Metrics

We collected four metrics during the teaching phase (teaching performance, teaching time, number of failures and number of inputs provided) and one during the testing phase (the testing performance). All interaction metrics were collected three times per conditions, once for each session. As not all participants reached a success during the testing phases, we used the six key steps defined in Section 5.3.2 as a way to evaluate the performance ranging from 0 (no step has been completed) to 6 (the task was successfully completed) during the testing phase. For example a testing where the robot puts both ingredients in the bowl but reaches a failure state before mixing them

[5] https://emmanuel-senft.github.io/experiment-irl.html

would have a performance of 3.

The testing performance represents the success of participants in teaching the robot to complete the task. On the other hand, the teaching performance corresponds to the highest step reached by participants in the teaching phase and represents a teaching method's ease of guiding the robot. The teaching time is the duration of the teaching phase, ranging from 0 to 25 minutes. The number of failures is the number of times a participant reached a failure state during the teaching phase. It can be related to the risks involved by the teaching; a safe teaching process should lead to a low number of failures, while a risky one would have a high number of failure. The number of inputs corresponds to the number of commands, guidances or feedback inputs used in a teaching session. Similarly to the teaching time, the number of inputs can be seen as the quantity of efforts invested in the teaching process.

Questionnaires

The post-interaction and post-experiment questionnaires provide additional introspective information to compare with the quantitative data from the interaction. 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 minimise the workload for the robot's user and to make the interaction as smooth and efficient as possible. Multiple definitions for workload exist and various measures can be found in the literature (e.g. Wierwille & Connor 1983; Moray 2013). Due to its widespread use in human factors research (Hart, 2006) and clear definition and evaluation criteria, we used the NASA-Task Load Index (TLX) (Hart & Staveland, 1988). Following the methodology proposed to administer NASA-TLX, we averaged the values from all 6 scales (mental, physical and temporal demand, performance, effort and frustration) ranging from 0 (low workload) to 20 (high workload) to obtain a single workload value per participant for each interaction. This assessment is made during the post-interaction questionnaires. This results in two measures of workload per participant, one for each condition.

Finally, the participants' perception of the robot was also evaluated in the post-interaction and post-experiment questionnaires using rating questions (measured on a 5 item Likert scale), binary questions (where participants had to select one of the two system), and open questions on the preference of system and the naturalness of the interaction.

5.4 Results

Most of the results collected in the study were non-normally distributed. Both ceiling and floor effects can be observed depending on the conditions and the metrics. For instance, for the teaching time, some participants preferred to interact much longer than others, resulting in skewed data. Likewise for the testing 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 and the absence of possible transformation making them normal, Bayesian statistics were conducted using the JASP software (JASP Team, 2018). Three types of test have been used: mixed ANOVA for omnibus comparisons between conditions for the first and the second interaction (between participants), independent t-test for post-hoc comparisons between participants and paired samples t-test for post-hoc comparisons within participants. All tests have been performed using their Bayesian counterpart, which also removed the need for doing a correction on post-hoc tests such as Bonferroni. As such, no p-value is reported, but a B factor representing how much of the variance on the metric is explained by a parameter (if $B < 1/3$ there is no impact, if $B > 3$ the impact is strong, and if $1/3 < B < 3$ the results are inconclusive; Jeffreys 1998; Dienes 2011).

For each interaction metric, two mixed ANOVA between participants were calculated to explore the impact of the conditions on the metric. The first ANOVA is applied to the first interaction (session 1,2 and 3) and compares participants in group 1 interacting with IRL and participants in group 2 interacting with SPARC. The second ANOVA is related to the second interaction (sessions 4, 5 and 6) and compares participants in group 1 interacting with SPARC and those in group 2 interacting with IRL. If required, additional post test were made within participants for the three sessions corresponding to each interaction to measure if successive interactions with a same system impact the metric.

5.4.1 Interaction data

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

Teaching Performance

Figure 5.3 presents the maximum performance reached by participants during the teaching phase, i.e how far in the steps they brought the robot during the teaching phase. It relates to the ease of guiding the robot through the task using a method. If a method does not allow a teacher to direct the robot's behaviour, the robot will have issues reaching useful states which will lead to a poor teaching performance. On the other hand, methods allowing the teacher to steer the robot to the desired parts of the environment should achieve a high teaching performance. The teaching performance 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 mixed ANOVA shows difference between conditions (and between participants) when interacting with the first system (participant in group 1 interacting with IRL and those in group 2 with IRL) and this effect is also present in the second interaction (first interaction:$B_1 = 2881$ - second interaction $B_2 = 76.2$). According to the medians shown in Table 5.2 and the graphs in Figure 5.3, for both interaction, participants using SPARC achieved a higher teaching performance than the ones using IRL. The session number (the repetition of additional sessions with the same system - within participants) had no impact on the teaching performance (first interaction: $B_1 = 0.089$ - second interaction $B_2 = 0.105$) which means that with additional interaction with a system participants did not reach a higher or lower teaching performance.

Table 5.2: Median performance in the teaching phase. Noted that between session 3 and 4 participants change system.

	$\widetilde{X}_1$	$\widetilde{X}_2$	$\widetilde{X}_3$	$\widetilde{X}_4$	$\widetilde{X}_5$	$\widetilde{X}_6$
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

This higher teaching performance for SPARC provides partial support for H1 and its prediction: 'SPARC will be more effective and efficient than IRL when used by non-experts'.

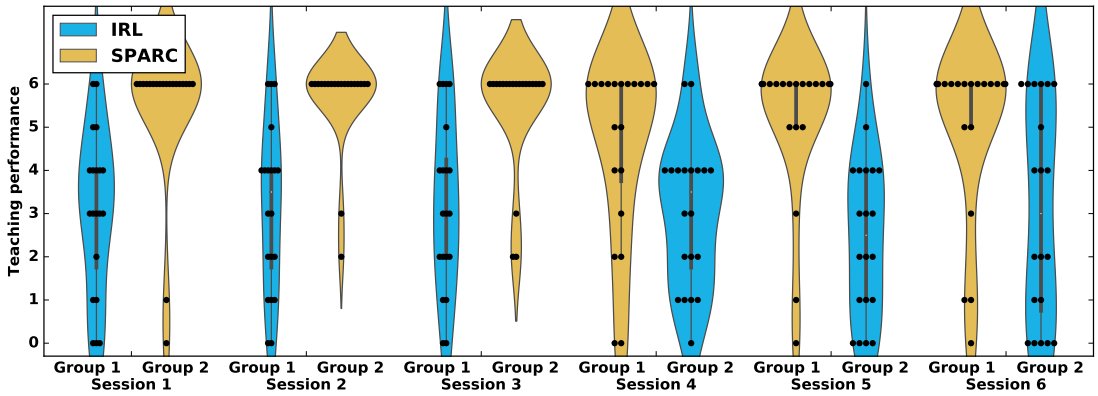


Figure 5.3: Comparison of the teaching performance for the six sessions (the left columns presents the data of participants in group 1 and the right ones those in group 2). The colours are swapped between session 3 and 4 to represent swapping of conditions. A 6 in teaching performance shows that the participant reached at least one success during the teaching phase. The vertical grey lines represent minimal barplots of the data and the shaded areas the probability distribution most likely to produce these results.

Testing Performance

Figure 5.4 presents the performance of the system during the testing phase, and represents how successful was the participants' teaching. The bayesian mixed ANOVA shows an effect of condition on the performance for both interactions ($B_1 = 8.8\text{x}10^5$ and $B_2 = 7340$). The median performance scores in Table 5.3 show that a higher test performance was achieved when participants used SPARC compared to when they used 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 $B_2 = 0.80$).

As shown in Table 5.3 and Figure 5.4, only a limited number of participants succeeded in teaching the robot to complete the task using IRL, this finding will be discussed in more details in section 5.6.

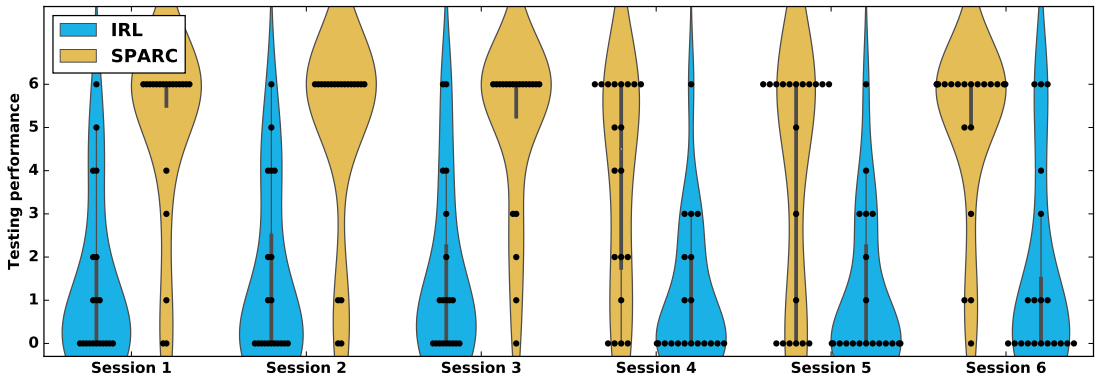


Figure 5.4: Comparison of the testing performance for the six sessions. A 6 in performance shows that the taught policy led to a success.

Table 5.3: 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

This higher testing performance for SPARC provides partial support for H1 and its prediction.

Teaching time

Figure 5.5 presents the time participants spent teaching. They could stop whenever they decided or the session would stop automatically after 25 minutes. The bayesian mixed ANOVA shows the important role of condition ($B_1 = 31.4$ and $B_2 = 679$) and session number on the time spent teaching ($B_1 = 8.3\text{x}10^9$ and $B_2 = 3188$). Table 5.4 and additional post-hoc comparisons between the sessions in each interaction 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 $B_{23} = 0.435$). A similar pattern occurs in the second interaction ($B_{45} = 850$, $B_{46} = 382$ and $B_{56} = 0.172$) with more support for a stabilisation of teaching time between session 5 and 6.

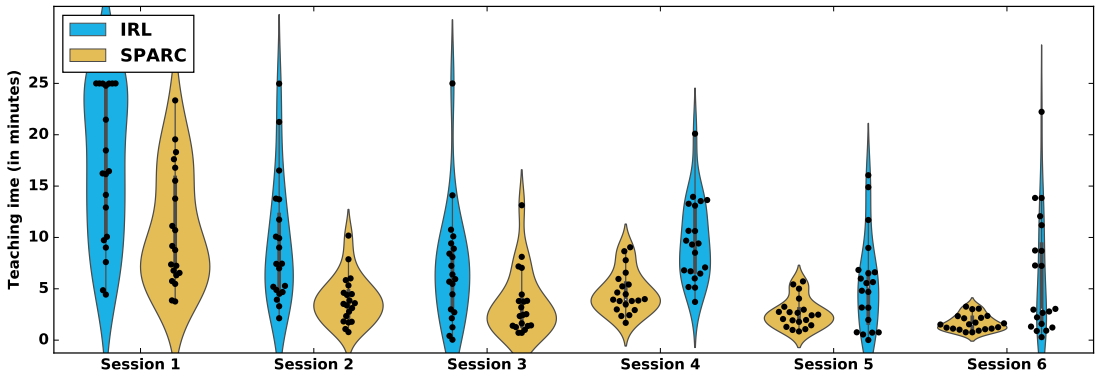


Figure 5.5: Comparison of the teaching time for the six sessions. At 25 minutes, the session stopped regardless of the participant stage in the teaching.

Table 5.4: 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

Combined with the consistent high performance of SPARC, this decrease of teaching time indicates that participants managed to learn an efficient way to use SPARC to

teach the robot a successful action policy. On the other hand, this similar decrease of teaching time and the lower performance with IRL could indicate that participants lost motivation to interact with IRL. As they did not find an efficient way to teach the robot with IRL, they dedicate less efforts to try in successive session. These interpretations provide partial support to H1 and its prediction.

Number of inputs

Figure 5.6 presents the number of inputs the participants provided while teaching. The bayesian mixed ANOVA shows that in both interactions, the condition had an impact on the number of inputs provided ($B_1 = 27.4$ and $B_2 = 34.1$). On the other hand, the session number only had a clear impact for the first interaction, the results are inconclusive for the second interaction ($B_1 = 4.1\text{x}10^5$ and $B_2 = 1.5$). Table 5.5 and additional post-hoc comparisons between sessions indicate that in the first interaction, the number of inputs used decreases between the first and second sessions and then tends to stabilise between the second and the third sessions($B_{12} = 2707$, $B_{13} = 4.7\text{x}10^4$ and $B_{23} = 0.410$). Similarly, for the second interaction, a difference tends to be observed between session 4 and 5 and session 4 and 6 while the number of inputs is similar between session 5 and 6 ($B_{45} = 2.6$, $B_{46} = 2.7$ and $B_{56} = 0.17$).

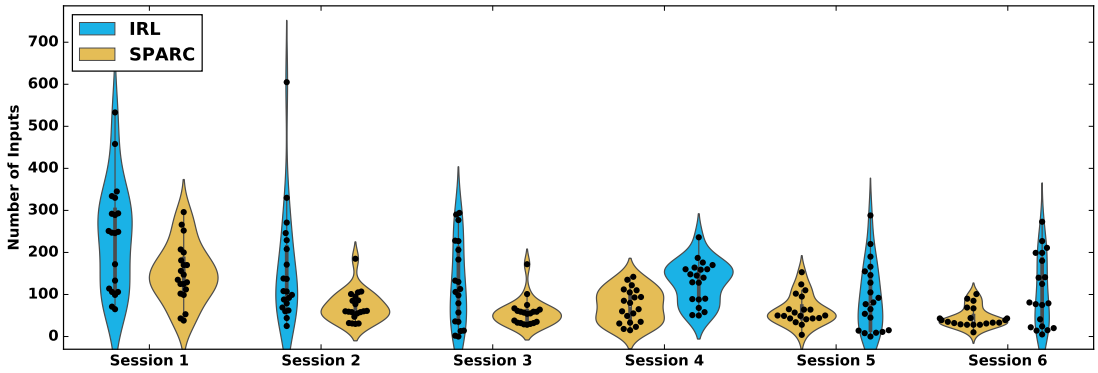


Figure 5.6: Comparison of the number of inputs provided by the participants for the six sessions.

Table 5.5: 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

Similarly to the teaching time, this reduction of inputs provided during the teaching, while maintaining a high performance for SPARC offers partial support for H1 and its prediction.

Number of failures

Figure 5.7 presents the number of failure states participants encountered during the teaching phase. The bayesian mixed ANOVA shows that for both interactions, both the condition ($B_1 = 6.2\text{x}10^4$ and $B_2 = 2.6\text{x}10^4$) and session number ($B_1 = 1.5\text{x}10^4$ and $B_2 = 11$) play an important role on the number of failures. Table 5.6 and additional post-hoc comparisons between 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 ($B_{12} = 619$, $B_{13} = 1.7\text{x}10^3$ and $B_{23} = 0.25$). Similar results can be observed in the second interaction ($B_{45} = 3.3$, $B_{46} = 7.5$ and $B_{56} = 0.2$).

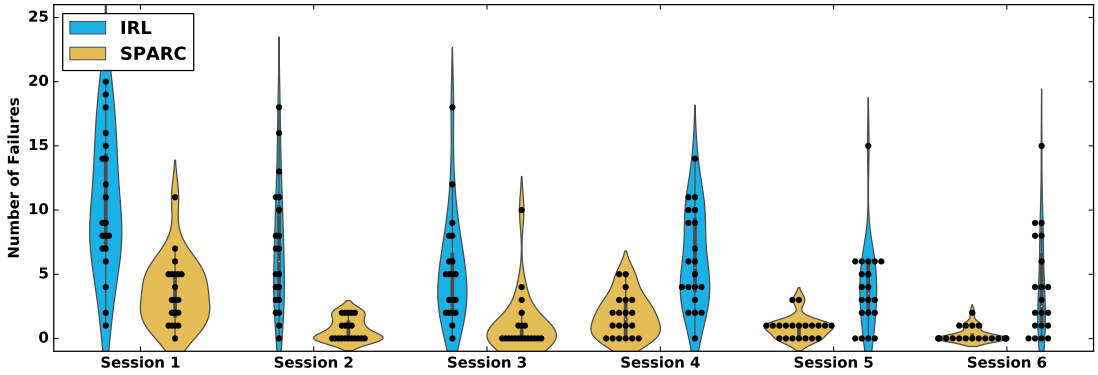


Figure 5.7: Comparison of the number of failures for the six sessions.

Table 5.6: 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

The fewer failures faced when using SPARC compared to IRL offers partial support to H1 and its prediction. Additionally, the low number of failures when using SPARC in the last sessions of both interaction (sessions 3 and 6) shows that participants became more efficient with SPARC, reaching successes without facing failures, which partially support H2: 'SPARC can be used by expert users (knowledgeable in the interaction process) to teach an action policy safely, quickly and efficiently'.

5.4.2 Questionnaire data

The main task of the post-interaction questionnaires was to assess the workload on participants when interacting with a condition using the NASA-TLX questionnaire. Figure 5.8 presents the workload for participants for each condition for both interactions (the average of the six ratings from 0 to 20 for each category). In the first interaction,

participants using IRL reported an average 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 IRL had an average workload of 13.87 ($SD = 2.84$) and the ones using SPARC reported 7.44 ($SD = 3.41$). Bayesian independent t-test show a strong effect on the condition for both interactions ($B_1 = 462$ and $B_2 = 8.1\text{x}10^4$) between participants. And bayesian paired t-test show a similar effect of the condition within participants for both orders (order 1: $B_{IRL-SPARC} = 1.7\text{x}10^6$ - order 2: $B_{SPARC-IRL} = 1.1\text{x}10^4$). Regardless of the comparison criteria (between or within subjects), participants reported a lower workload when interacting with SPARC than when interacting with IRL.

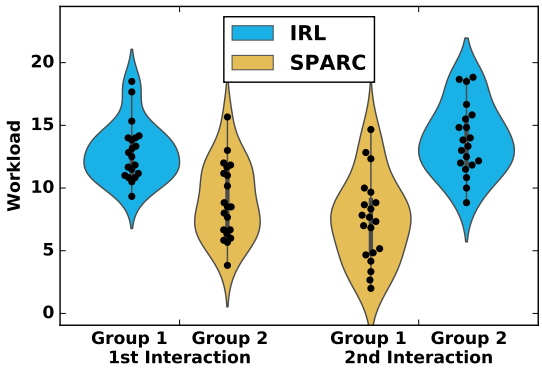


Figure 5.8: Average workload for each participants as measured by the NASA-TLX for each conditions in both interaction order.

This lower workload when using SPARC compared to IRL offers partial support for H1 and its predictions.

5.4.3 Expert

To evaluate the best case potential offered by SPARC and IRL, an expert in Human-Robot Interaction (HRI) knowing the detail of the algorithm and the interactions (the author) interacted five times with each system. For both systems, the expert followed a strictly optimal strategy. In the case of IRL the optimal strategy consisted on providing as much guidance as possible, rewarding positively correct actions and negatively incorrect ones. For SPARC the optimal strategy consisted on providing commands for every single action to demonstrate an optimal trajectory to the robot. This shows the expected behaviours in optimal conditions, the best metrics achievable. Results of the interactions are presented in Table 5.7. In both cases, the expert successfully taught the robot (as indicated by a performance of 6 during the teaching and the test), which indicates that both systems can be used to teach a robot an action policy. However as demonstrated by a bayesian independent t=test, the time required to teach the robot with IRL is higher

than with SPARC ($B = 7102$).

Table 5.7: Results of an expert interacting 5 times with each system following an optimal strategy. When the variance is 0, 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

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. Conversely, 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 of the interaction possibilities, an expert can teach the robot to behave safely without having to explore and reach undesired states using SPARC. This has real world applications in HRI, as random exploration is often impossible or undesirable when interacting with humans. SPARC offers a way for the teacher to stop the robot from executing actions with negative consequences whilst still guiding the robot toward useful parts of the environment.

Similar results to these were observed with our non-expert participants: in their last session with SPARC, both groups had a median of 0 failures and a performance of 6. This indicates that more than half of the participants successfully taught the robot the task without ever hitting a failure state after gaining understanding of SPARC in their first and second interaction with it.

The absence of failures, the lower number of inputs and the shorter time required to teach with SPARC compared to IRL when used by an expert user provide support for H2.

5.5 Validation of the hypotheses

5.5.1 Effectiveness and efficiency with non-experts

The objective metrics show that despite spending a shorter time interacting with SPARC and using fewer inputs, participants reached a higher performance than with IRL and faced fewer failures during teaching. Additionally, when interacting with SPARC, the time participants took to teach the robot decreased to reach a plateau in the second and third sessions, without negatively affecting the performance. This indicates that after the first

session, participants understood the interaction mechanism of 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 remained low over the sessions, and their teaching time decreases between session 1 and 2 but not further between session 2 and 3. This decrease of teaching combined with low performance 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. This conclusion is supported by subjective measures: the workload on the teacher is lower when using SPARC than when using IRL.

For these reasons, H1 and its prediction is ('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, an expert can reach a success easily and safely (requiring a low number of inputs and a short time and without facing a single failure). 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 session with SPARC.

However, when interacting with IRL, even the expert applying a strictly optimal policy cannot prevent the robot reaching failures states. This effect is due to the lack of control of feedback-based IML methods. As teachers only rate the actions of the agent, they cannot prevent the learners from making errors. They can only negatively reward these errors to reduce their chance of being selected in the future. While the guidance allows to partially mitigate this effect, the presence of actions not covered by this guidance limits the its efficiency during the teaching.

This difference shows support for H2 ('SPARC can be used by expert users to teach an action policy safely, quickly and efficiently, achieving better results other IML methods lacking control'). This also demonstrates how the principles presented in Chapter

3 provide control to the teacher over the robot's actions and by extend improve the teaching. Consequently, the principles underlying SPARC ensure that even in the early stages of teaching (when the robot's action policy is not mature to correctly select actions without supervision), the action policy of the robot is appropriate, which is not the case of most other IML methods (as demonstrated by the number of failures when teaching with IRL).

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 the freedom to explore, whilst receiving feedback on its actions and limited guidance about what action to pursue next from a teacher. This social aspect of the teaching: with hints and guidance, partial control over the robot actions and bidirectional communication is inspired by the way humans teach. While not every member of the population is knowledgeable about Machine Learning (ML), they are experienced with social learning (Thomaz & Breazeal, 2008). This similarity between how humans teach robots and other humans has also been supported by behaviours displayed by participants in the original study. Participants gave motivational rewards to the robot, just as one would do to keep a child motivated during learning, despite the absence of effect or use in classical reinforcement learning (Thomaz & Breazeal, 2008).

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 the robot in a non-social way how to interact socially in a non-social way. This approach is more task oriented, and we argue that it better fits many applications of HRI when the interaction with the teacher does not have to be social. For example, in Socially Assistive Robotics (SAR), the task (such as interaction with a child with Autism Spetrum Disorder (ASD)) is more important than the social relationship between the robot and its supervisor (a therapist for example) and as such the relevance of the social side of the interaction between the teacher and the robot is reduced.

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's 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 not to have total control over the robot's actions with IRL, they would have preferred being able to control each of the robot's actions.

To the question 'which interaction was more natural?', 10 participants rated IRL as being more natural, using justifications such as: 'The robot 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 these 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 outcome of the teaching, on the learner's proficiency in the task. As mentioned previously, this might relate to the role of robots, they often interact in human-centred scenarios 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 for use 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 an efficient teaching interaction. 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 (designer or trained participants), 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 to interact in human environments. As robots interact in task oriented, human-centred environments, human teachers need approaches with direct control and more focused on commands rather than letting the

robot explore on its own and only evaluate its actions.

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: when asked to rate their expertise in ML software and system (1=none, 7=very experienced), they reported an above average score (M=3.7, SD=2.3), 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 - on a 5-item Likert scale). 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 even during the teaching phase in the IRL condition. We identify the lack of control over the robot's actions as a limiting factor for the teaching, as participants did not manage to steer the robot to do correct actions, they could not reward them and teach the robot an efficient action policy. Additionally, the requirement of explicit feedback made the learning task more complex. Participants often did not reward an action after a guidance, assuming that informing the robot what it should do did not require an additional explicit reward. Teachers of robots need to have control over the robot's action and robots should also use implicit rewarding to ease the task for the teacher. For example, SPARC uses implicit rewarding by automatically providing a positive reward to actions selected by the teacher and also every action not corrected by the teacher as it has been implicitly validated. 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 implementation of SPARC for this study, the algorithm mostly reproduces actions selected by the teacher. One could 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. As a state visited multiple times in an episode is considered as a single identical state for the learning algorithm, loops in the demonstrations can be removed for future executions of the action policy. Additionally, this provide the algorithm with a way to deal with variations in teaching. It allows the robot to reach a success from the initial state but also to continue the action policy from any state in the trajectory. And finally, due to the suggestion/correction mechanism, the teacher can let the robot act on its own only intervening 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 returning to a previous state (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. This is a limitation as a lack of supervision can lead to undesired reinforcement of incorrect behaviours.

In this study, SPARC has been applied to a scenario where a clear strategy with optimal actions was present. The interaction also took place in a virtual environment with a discrete time and a limited number of states. Real human-robot interactions are stochastic, happen in real time and often there is no clear strategy known in advance, and as such present limited similarity with this study. However, in these real HRI, human experts in the application domain know what type of actions should be executed when, and which features of the environment they used for their decision. And, as this knowledge might not be available to the robot's designers or could be complex to formalise in a set of rules a robot should follow, robots should be able to learn from a domain user in an interactive fashion. And this study presented a simple environment allowing us learn more about how humans could teach a robot using SPARC.

The limitations of this study (simulated deterministic world, limited number of state, discretisation of time and absence of interaction with humans) have been addressed in Chapter 6. In the study presented in that chapter, SPARC has been applied to a real-world social interaction with humans, possessing all the challenges typical to these

interactions: complex non-deterministic world, with continuous time, real impact of actions and importance of the social factors in the interaction.

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

From observing 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 and transparency

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 interpretable, 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 a more classical Graphical User Interface (GUI) with buttons. This design decision had important consequences: as the interface is not explicit, participants first had to familiarise themselves with the interface, discover how to interpret the robot's behaviour, which actions are available for each state and learn the exact impact of the robot's actions. This lack of clarity led 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), such as including the users in the design process or finding intuitive ways 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 held 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 and undo behaviour IRL or 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 by a naive operator, they need mechanisms to detect these 'incorrect' uses, and either adapt to these suboptimal human inputs or at least inform the user that this type of input is not efficient 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 process is key. This human has the opportunity to provide important knowledge about the environment and allow the machine learning to deal with sensor errors or imperfect action policies. As in real world a robot's behaviour can hardly be perfect, keeping a human in the learning loop allows to continue improving the robot's behaviour even after an acceptable policy is reached. This is different to most of the Learning from Demonstration (LfD) approaches where the robot is left unsupervised to interact once an action policy is learned (Argall et al., 2009; Sequeira et al., 2016). This was one of the important points we considered when proposing SPARC: there is no distinction between a teaching and a testing phase, they are merged into a single phase, moving away smoothly from Wizard-of-Oz (WoZ) to Supervised Autonomy. The teacher can correct the robot when needed and let it act when it behaves correctly. In this study, participants used this feature of SPARC: many participants corrected SPARC only when required rather than forcing every action. For example, 37.5% of the participants even let the robot complete the task once without giving a single command before starting the test to be sure that the robot was ready. This demonstrate that keeping the human in the learning loop is important to ensure that the robot's behaviour stays appropriate; and this study demonstrated that SPARC allows human teachers to be actively involved in the teaching process without requiring important workload for them.

Keeping teachers in control

Most of the scenarios 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 SAR). In these scenarios, the goal of learning is to ensure that the robot can complete the task it has been assigned, not to provide the robot with tools to learn more efficiently in further interactions. Accordingly, 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 (see Section 5.5.3). In addition to reducing the effectiveness of the learning, a lack of control over the robot's actions can lead to frustration and loss of motivation for the teacher as shown with the results of IRL in this study. This human control is especially critical when the robot is designed to interact with other people because undesired actions can have a dramatic impact, such as causing physical or mental harm for the interaction partners or bystanders. For these reasons, we argue that when designing an interactively learning robot for HRI in human-centred scenarios, 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. With SPARC the human is in control during all the interaction, but especially when the robot is prone to making exploratory mistakes, so that the teacher can prevent them before they occur. But once the action policy is appropriate enough, the teacher can leave the robot to interact mostly on its own, providing only limited supervision to refine the action policy.

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 an appropriate behaviour. This chapter presented a study where SPARC was combined with RL to teach a simulated robot to complete a baking task. SPARC used intentions communicated by the robot, full control over the robot's behaviour and an implicit rewarding mechanism to allows participants teach the robot an action policy. A study involving 40 participants compared this approach with IRL, another IML approach using communication of uncertainty, partial control and explicit rewarding to teach the robot. When interacting with SPARC,

participants took less time and fewer inputs to reach more successes, whilst facing fewer failures. Participants also reported a lighter workload when using SPARC than when interacting with IRL. This study, demonstrated that SPARC is usable by naive participants to successfully teach a robot an action policy quickly and safely.

Based on these results and our observations of the participants, we propose four guidelines to designing interactive learning robots: (1) the interface to control the robot should be intuitive, (2) the limits of human adaptability have to be taken into account (robots should detect deadlocks in human behaviours and adapt how they are controlled or inform the human about these incorrect behaviours), (3) the operator should be kept in the learning loop and (4) teachers should stay in control of the robot's behaviour when interacting in a sensitive environment (such as Robot Assisted Therapy (RAT)). 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.

In summary, this chapter extended SPARC and compared it to other methods from the IML field. SPARC succeeded in its goal, allowing participants to teach easily and safely an action policy to a robot. Finally, insights from this study have been used to guide the design of study presented in Chapter 6, the final study of this research, which involves teaching a robot to interact with humans in real-world HRI.

Chapter 6

Teaching a robot to support child learning

Key points:

- An experiment was designed to test Supervided Progressive Autonomous Robot Competenties (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).

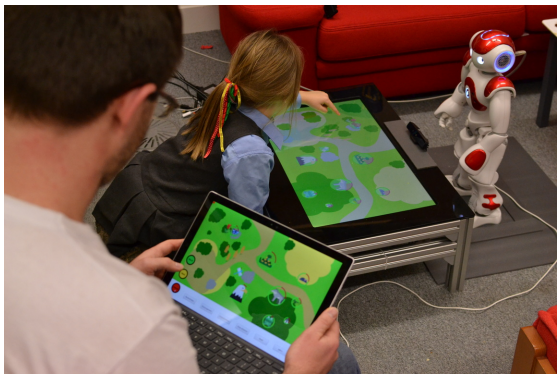


Figure 6.1: Setup used in the study: a child interacts with the robot tutor, with a large touchscreen sitting between them displaying the learning activity; a human teacher provides guidance to the robot through a tablet and monitors the robot learning.

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 Graphical User Interface (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.

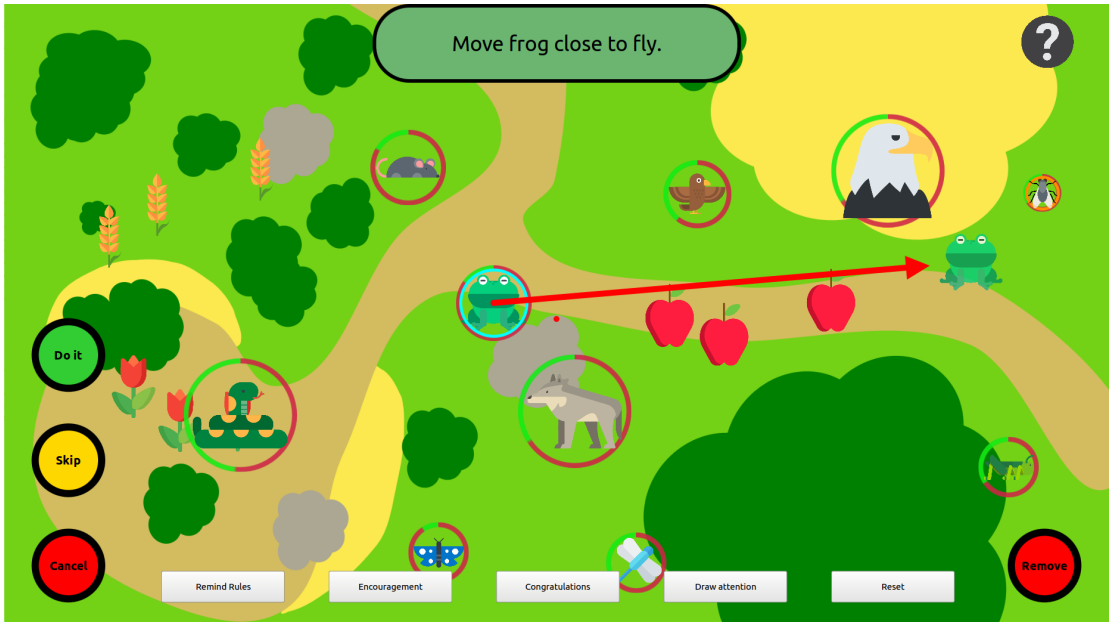


Figure 6.3: GUI used by the teacher to control the robot and respond to its suggestions. The game presents the same state as in Figure 6.2, and he robot proposes to move the frog close to the fly (text bubble, arrow, moving the *shadow* of the frog and highlight of the frog and the fly).

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.

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.

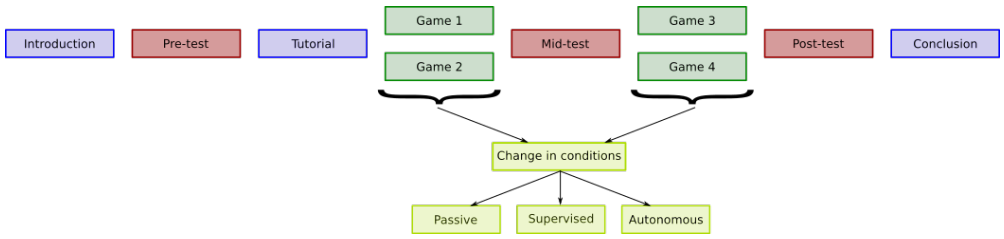


Figure 6.4: Methodology used for the study.

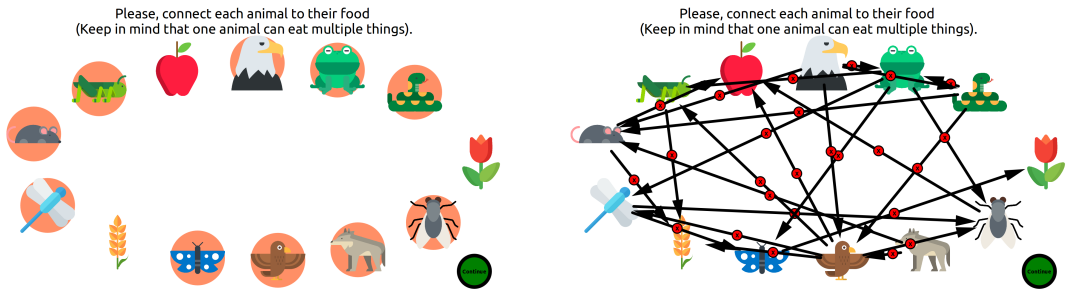
In all the conditions, the robot's behaviour during the introduction, tests, tutorial and conclusion is identical. The only change of behaviour happens during the games sessions. Figure 6.2 shows an example of the game screen. The child can move 10 animals across the game field and can have them interact 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.



(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

To demonstrate the presence or the absence of effects, we used bayesian statistics on the data. As such the Bayes factor B is reported and represents how much of the variance on the metric is explained by a parameter (if $B < 1/3$ there is no impact, if $B > 3$ the impact is strong, and if $1/3 < B < 3$ the results are inconclusive - Jeffreys (1998); Dienes (2011)). And the analysis is performed using the Jasp software JASP Team (2018).

Test performance Figure 6.6 shows the evolution of children's performance across the three tests. A Bayesian mixed-ANOVA show that in all conditions, children's performance increased across the tests ($B = 1.5\text{x}10^{12}$), however the impact of the condition on the learning is inconclusive with a tendency to show no impact ($B = 0.539$). This indicates that by being involved in the task, every children learned and improve their performances on the test (by gaining in average 13% of the missing knowledge), but the robot behaviour during the game did not have an important impact on the children's learning gain (see Figure 6.7).

Game metrics Figure 6.8 shows the evolution of the number of different eating behaviour exhibited by the children across the four game sessions. A Bayesian mixed-ANOVA shows an impact of the condition on the number of different eating behaviour produced by the children in the game ($B = 6.1$). Post-hoc tests show that there is no difference between the Supervised and the Autonomous conditions ($B = 0.154$), whilst differences are observed between the Supervised and the Passive condition ($B = 512$) and between the Autonomous and the Passive conditions ($B = 246$). This indicates that

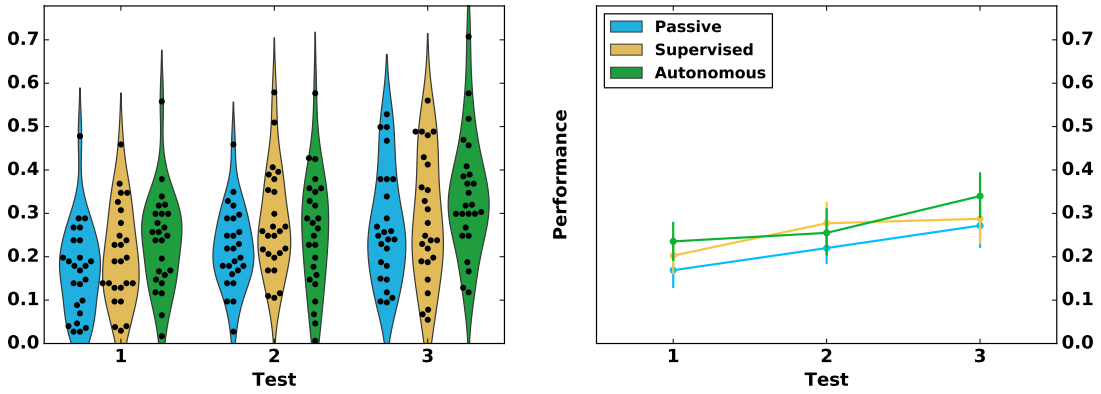


Figure 6.6: Children's performance for the three tests: pretest, midtest and posttest for the three conditions.

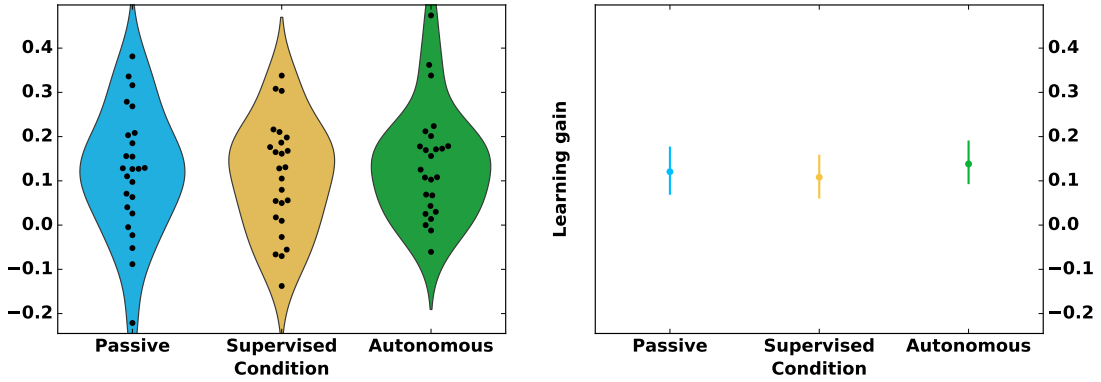


Figure 6.7: Children's normalised learning gain after interacting with the robot for the three conditions.

the Supervised robot provided additional knowledge to the child during the game, allowing them to create more useful interactions between animals and their food, receiving more information from the game potentially helping them to learn. The Autonomous robot managed to recreate autonomously this effect without the presence of a human providing input.

Figure 6.9 shows the evolution of the number of points achieved by the children across the four game sessions. A Bayesian mixed-ANOVA shows an impact of the condition on the number of points achieved by the children in the game ($B = 10.2$). Post-hoc tests show a strong difference between the Passive and the Supervised conditions ($B = 5.1\text{x}10^4$) and differences between the Supervised and the Autonomous conditions ($B = 5.2$) and the Autonomous and the Passive condition ($B = 5.9$). This indicates that when the robot was supervised, it allowed children to achieved more points than a passive robot. And a similar effect is observed when the robot is autonomous, however the autonomous robot did not manage to reach the same efficiency as the supervised robot in helping the children to achieve a high score in the game.

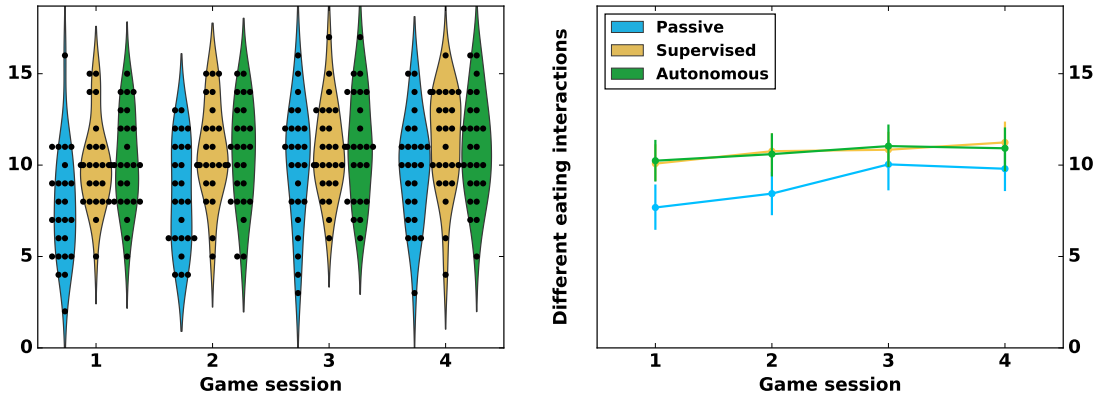


Figure 6.8: Number of different eating behaviour for the four games for the three conditions.

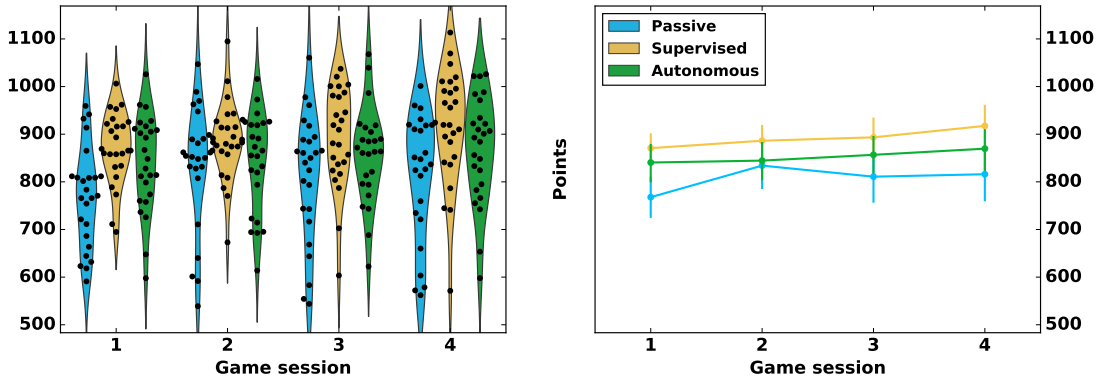


Figure 6.9: Points achieved by the children in each game session for the three conditions.

Figure 6.10 shows the evolution of interaction time across the four game sessions. A Bayesian mixed-ANOVA shows is inconclusive on the impact of the condition on the interaction time in the game ($B = 1.1$). However, post-hoc tests show that there is no difference between the Supervised and the Autonomous conditions ($B = 0.287$), whilst differences are observed between the Supervised and the Passive condition ($B = 118$) and a tendency of difference between the Autonomous and the Passive conditions ($B = 2.9$). This indicates that the Supervised robot allowed children to be better at the game, allowing them to maintain animal alive longer than a Passive robot. And the autonomous robot learn a policy tending to replicate this effect and without exhibiting differences with the supervised one.

These game metrics show that the action policy executed by the autonomous robot allows children to achieve similar results in the game than when the robot is supervised, and better results than when interacting with a passive robot. This provide support for H2 ("The robot reproduce the action policy demonstrated by the teacher enabling the child to achieve similar game metrics in autonomous and supervised condition but better

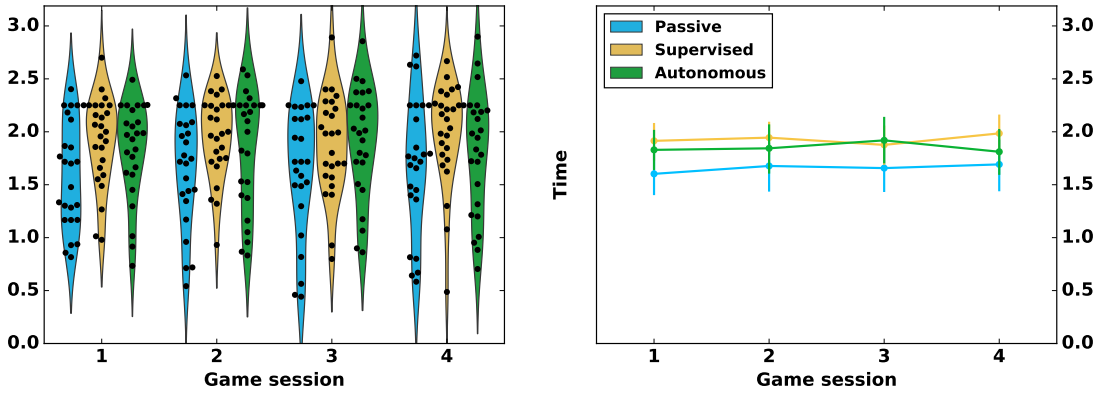


Figure 6.10: Interaction time for the four games for the three conditions.

than the passive condition"). However, children learned similarly in the three conditions, so these improvements in the game did not transfer to improvements in the test neither for the Supervised robot nor the Autonomous one. This does not support H1 ("The robot support child learning: learning gain in passive condition $<$ learning gain in autonomous condition $<$ learning gain in supervised condition")

Supervision Figure 6.13 presents the reaction of the teacher to the robot's suggestions across all the supervised sessions. In average the teacher accepted 22.3% of all the proposition of the robot (by enforcing the action, let it be executed or using the 'Do it' button), which represents 35% of the actions executed by the robot. This effect tends to be stable across the sessions. The teacher interaction pattern evolved overtime, such as by using mostly the 'Cancel' button in the start then the 'Skip' one, but in the end, the teacher used this two buttons mostly interchangeably even if the algorithm underlying reaction is different. Another observation is the evolution from using the auto-execution function to the 'Do it' button once the teacher felt more comfortable in the supervision. The teacher reported three phases in her teaching:

- First sessions: she was not paying much attention to the suggestions, mostly trying to have the robot executing a correct action policy.
- Session 20 to around 65: she was paying more attention to the suggestions without giving them much credit.
- Last sessions: she started to trust the robot more but without ever trusting it totally.

Additional comments:

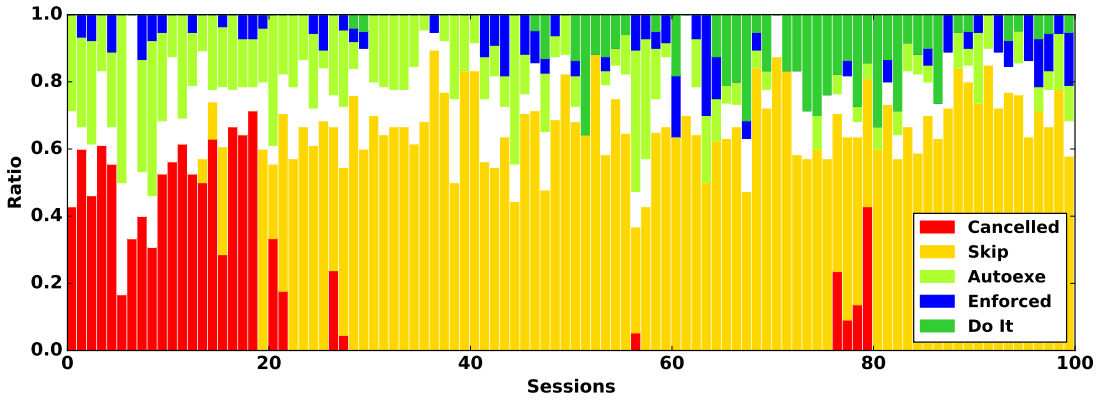


Figure 6.11: Teacher's reaction to the robot's propositions along the sessions.

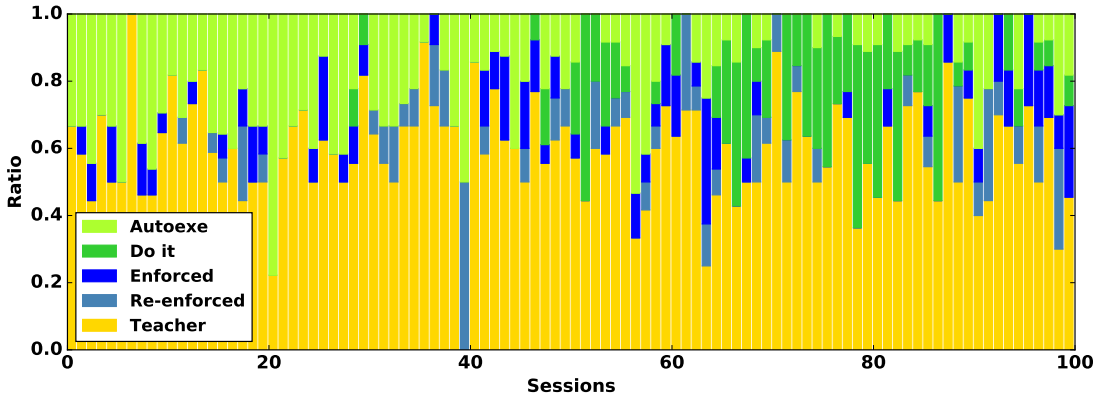


Figure 6.12: Origin of the actions executed by the robot. Re-enforced actions indicates that an action has been selected after having been cancelled or skipped by the teacher.

- The robot proposed in average 58% more actions than the number of actions selected by the teacher.
- As the time is continue and not discrete, the concept of correct actions is shifted, and actions selected by the teacher might have been proposed by the robot a fraction of second later without being counted as good proposition.
- The teacher often cancelled/skip actions directly as they arrived without taking time to analyse them (limitation of this implementation of SPARC).
- Evolution of teaching policy limits the potential of learning (not one single policy applied by the teacher, but an evolving one).
- Children are different, so the teacher tried to apply different action policy for each child.

Comparison of policy Figures 6.14 and 6.15 present the policies used by the teacher and the autonomous robot across their participants. It should be noted that participants

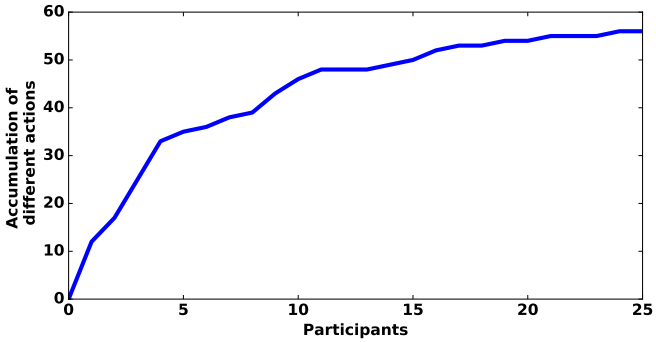


Figure 6.13: Accumulation of the number of different actions used by the teacher across the participants.

in both conditions are different.

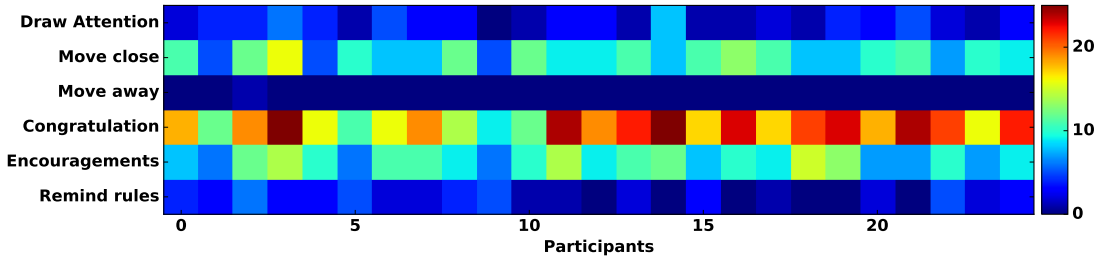


Figure 6.14: Repartition of actions across the participants in the supervised condition.

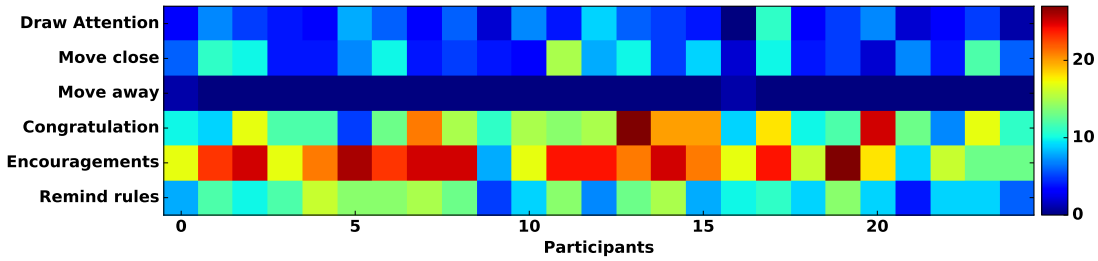


Figure 6.15: Repartition of actions across the participants in the autonomous condition.

6.5 Discussion

6.6 Summary

Table 6.1: Repartition of action in the policy for both conditions (in %).

	Draw Attention	Move close	Move away	Congratulation	Encouragements	Remind rules
Supervised	6.6	22.2	0.1	43.1	22.7	5.3
Autonomous	8.5	11.8	0.1	25.4	35.2	18.9

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 Human-Robot Interaction (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

Glossary

Supervised Autonomy An agent acts autonomously in the world while being monitored by a human. The agent proposes actions which will be executed to the human who has the power to pre-empt them and to select actions to be executed by the agent at any time. 50, 53, 58–60, 100

Acronyms

ASD Autism Spetrum Disorder. 12, 20, 58, 60, 95

CIDM 95% Confidence Interval of the Difference of the Mean. 70, 71

DREAM Development of Robot-Enhanced Therapy for Children with Autism Spectrum Disorders (European FP7 project). 57

GUI Graphical User Interface. 5, 6, 61, 63, 65–67, 99, 106, 107

HCI Human-Computer Interaction. 26, 29

HHI Human-Human Interaction. 14

HRC Human-Robot Collaboration. 17, 31, 33, 54

HRI Human-Robot Interaction. 8, 9, 11, 14, 19, 21, 23–27, 31–33, 36–38, 40, 44, 45, 48, 58, 60, 92, 93, 95, 100, 101, 119

ILfD Interactive Learning from Demonstration. 42, 54

IML Interactive Machine Learning. 9, 11, 32, 34, 35, 38, 42, 45, 48, 51, 53, 54, 76, 79, 94, 101

IRL Interactive Reinforcement Learning. 41, 75–83, 86–97, 99–101

LfD Learning from Demonstration. 2, 28, 30–32, 35, 36, 39, 42, 43, 48, 54, 55, 99, 100

LfW Learning from the Wizard. 28, 29

MDP Markov Decision Process. 37, 80

ML Machine Learning. 9, 17, 32, 40, 45, 49, 95, 96

NLP Natural Language Processing. 27

RAT Robot Assisted Therapy. 12, 22, 25, 29, 52, 58, 60, 72, 73, 101

RL Reinforcement Learning. 8, 34–41, 48, 52, 54, 55, 75, 76, 96, 101

SAR Socially Assistive Robotics. 11, 12, 95, 100

SL Supervised Learning. 35, 55, 76

SPARC Supervided Progressive Autonomous Robot Competenties. 5, 9, 48–55, 57–61, 65, 68–70, 72, 73, 75–83, 86–101, 103, 104

WoZ Wizard-of-Oz. 5, 9, 26–29, 52, 55, 57–61, 65, 68–70, 72, 100

XAI Explainable Artificial Intelligence. 17

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).

Adams, J. A., Rani, P., & Sarkar, N. (2004). Mixed initiative interaction and robotic systems. In *AAAI Workshop on Supervisory Control of Learning and Adaptive Systems*, (p. 613).

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.

Alshiekh, M., Bloem, R., Ehlers, R., Könighofer, B., Niekum, S., & Topcu, U. (2017). Safe reinforcement learning via shielding. *arXiv preprint arXiv:1708.08611*.

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.

Asada, H., Slotine, J.-J., & Slotine, J.-J. (1986). *Robot analysis and control*. John Wiley & Sons.

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 'sandtray' 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.

Bennewitz, M., Faber, F., Joho, D., Schreiber, M., & Behnke, S. (2005). Towards a humanoid museum guide robot that interacts with multiple persons. In *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, (pp. 418–423). IEEE.

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.

Bouton, M. E. (2007). *Learning and behavior: A contemporary synthesis.*. Sinauer Associates.

Bray, T. (2009). *Confronting the shadow education system: What government policies for what private tutoring?*. United Nations Educational, Scientific and Cultural Organization; International Institute for Educational Planning.

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.

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *Robotics and Automation, IEEE Journal of*, *2*(1), 14–23.

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).

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.

Cohen, P. A., Kulik, J. A., & Kulik, C.-L. C. (1982). Educational outcomes of tutoring: A meta-analysis of findings. *American educational research journal*, *19*(2), 237–248.

Cooper, J. O., Heron, T. E., Heward, W. L., et al. (2007). Applied behavior analysis.

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).

Dautenhahn, K. (2004). Robots we like to live with?!-a developmental perspective on a personalized, life-long robot companion. In *Robot and human interactive communication, 2004. ROMAN 2004. 13th IEEE International Workshop on*, (pp. 17–22). IEEE.

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.

Gordon, G., Spaulding, S., Westlund, J. K., Lee, J. J., Plummer, L., Martinez, M., Das, M., & Breazeal, C. (2016). Affective personalization of a social robot tutor for children's second language skills. In *AAAI*, (pp. 3951–3957).

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).

Grollman, D. H., & Jenkins, O. C. (2007). Dogged learning for robots. In *Robotics and Automation, 2007 IEEE International Conference on*, (pp. 2483–2488). IEEE.

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. (2006). Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, vol. 50, (pp. 904–908). Sage Publications Sage CA: Los Angeles, CA.

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.

Hayes, B., & Shah, J. A. (2017). Improving robot controller transparency through autonomous policy explanation. In *Proceedings of the 2017 acm/ieee international conference on human-robot interaction*, (pp. 303–312). ACM.

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.

Howard, R. A. (1960). *Dynamic programming and Markov processes*. MIT Press.

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.

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.

Kennedy, J., Baxter, P., Senft, E., & Belpaeme, T. (2016). Social robot tutoring for child second language learning. In *The Eleventh ACM/IEEE International Conference on Human Robot Interation*, (pp. 231–238). IEEE Press.

Kennedy, J., Lemaignan, S., Montassier, C., Lavalade, P., Irfan, B., Papadopoulos, F., Senft, E., & Belpaeme, T. (2017). Child speech recognition in human-robot interaction: evaluations and recommendations. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, (pp. 82–90). ACM.

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*.

Leite, I., Martinho, C., & Paiva, A. (2013). Social robots for long-term interaction: a survey. *International Journal of Social Robotics*, *5*(2), 291–308.

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.

Leyzberg, D., Spaulding, S., Toneva, M., & Scassellati, B. (2012). The physical presence of a robot tutor increases cognitive learning gains. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 34.

Linder, S. P., Nestrick, B. E., Mulders, S., & Lavelle, C. L. (2001). Facilitating active learning with inexpensive mobile robots. In *Journal of Computing Sciences in Colleges*, vol. 16, (pp. 21–33). Consortium for Computing Sciences in Colleges.

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*.

Matarić, M. J., Eriksson, J., Feil-Seifer, D. J., & Winstein, C. J. (2007). Socially assistive robotics for post-stroke rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, *4*(1), 5.

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.

Moray, N. (2013). *Mental workload: Its theory and measurement*, vol. 8. Springer Science & Business Media.

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).

Salter, T., Dautenhahn, K., & Bockhorst, R. (2004). Robots moving out of the laboratory-detecting interaction levels and human contact in noisy school environments. In *Robot and Human Interactive Communication, 2004. ROMAN 2004. 13th IEEE International Workshop on*, (pp. 563–568). IEEE.

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. (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., MIT 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.

Tanaka, F., & Matsuzoe, S. (2012). Children teach a care-receiving robot to promote their learning: Field experiments in a classroom for vocabulary learning. *Journal of Human-Robot Interaction*, *1*(1).

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.

Topping, K. J. (2005). Trends in peer learning. *Educational psychology*, *25*(6), 631–645.

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.

Wachter, S., Mittelstadt, B., & Floridi, L. (2017). Transparent, explainable, and accountable ai for robotics. *Science Robotics*, *2*(6).
URL `http://robotics.sciencemag.org/content/2/6/eaan6080`

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.

Wierwille, W. W., & Connor, S. A. (1983). Evaluation of 20 workload measures using a psychomotor task in a moving-base aircraft simulator. *Human factors*, *25*(1), 1–16.

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.