Control of complex actions in humans and robots

Kleijn, R.E. de

CHAPTER 1

General introduction

Robots are claimed to take over our jobs soon, and after that the world. Of course, in order for that to happen, the robots that would do that are nothing like the robots we know traditionally. No, the robots that will take over our jobs will be *smart* robots. But what is it exactly that makes a robot smart? One definition could be that a smart robot is a robot that can do things that humans can, such as doing the dishes or cooking your dinner. At the moment, there are many examples of software that can do things even better than humans can, such as playing chess [26], recognizing faces [116], and even playing Texas Hold 'Em poker [119]. Impressive as that may be, all these applications of artificial intelligence are domain-specific, and the intelligence they seem to possess in particular tasks is not generalizable to other tasks.

1.1 Human everyday action

If we truly want artificial intelligence to power the robots we know from Hollywood—that is, the robots that we can talk to, can interact with, and that can perform all different kinds of tasks for us—we may need to look at humans for inspiration. Thankfully, the tasks we would want such robots to perform have been the subject of study, and are collectively

known as *everyday action*. Good examples of everyday action that are often used in the literature are tea making, eating breakfast, and driving to work. But although the use of "everyday" could be interpreted as meaning "trivial", everyday human action is far more complex than the phrase may imply.

While we perform these everyday actions often without effort, it is clear that there are dependencies between subactions and dependencies on world knowledge that make these actions far from trivial. We can subdivide the action (or goal, depending on your theoretical viewpoint) of tea making into the subactions (1) getting a kettle from the cupboard, (2) filling the kettle with water, (3) putting the kettle on the stove, (4) pouring the boiling water in a teapot, (5) adding a teabag to the teapot, (6) getting a teacup from the cupboard, (7) pouring the tea into a teacup, and (8) adding some milk to the teacup. Although this is a good description of a single episode, it is quite clear that the information contained in this action plan is not enough for a completely naive person (or robot, for that matter) to successfully complete the action.

First, completing some of the subactions requires specific world knowledge that may not be available to the agent. For example, filling the kettle with water requires knowing that water is generally drawn from the tap in the kitchen. Second, the action plan is a high-level description of the task, and is severely underspecified with regards to motor parameters. In other words, it is necessary to convert the symbolic information in the action plan to subsymbolic information needed to actually *perform* the action. Several models have actually been proposed to explain motor action by integrating both symbolic and subsymbolic information (e.g. [90, 131]). Third, not all subactions are equal. Some can, under some circumstances, be omitted while still completing the action somewhat successfully. For example, skipping the pouring of milk into the teacup may not be that big of a problem, depending on the taste of the drinker. However, refraining to get a teacup from the cupboard will cause a problem for anyone longing for tea.

It should now be clear that, although everyday action seems trivial, it in

fact relies on mechanisms that are quite complex. Creating a robot that could perform everyday action by instructing it symbolically (e.g. by providing it with a recipe) or by haptic or observational instruction was the goal of the FP7-funded *RoboHow* project [31], and the research described in this dissertation was conducted as part of this project.

1.2 Project background

RoboHow's scientific goal was to "[enable cognitive] robots to competently perform everyday human-scale manipulation activities—both in human working and living environments." Cognitive robots are robots that reason, plan, and act, similar to humans. Due to the scope of the project, a consortium consisting of roboticists, computer scientists, and cognitive psychologists spread over five universities, two research institutes and one industry partner was formed, with Prof. Michael Beetz at Technische Universität München as project PI. Similar to humans, the resulting robot would be able to take a high-level action plan (such as a recipe) as input and successfully execute it (see Figure 1.1). As described before, this requires much more effort that it seems at first sight, and in fact the complexity of everyday action has proven to be one of the biggest obstacles in the RoboHow project. The project was completed in the summer of 2016.

1.2.1 RoboHow work packages

In order to tackle the complex problem of executing everyday action, the project was divided into nine work packages, of which six focussed on actual research problems, with each work package solving part of the problem before integrating the different work packages into a single robot pipeline:

- **Representation**: How are activities, knowledge, and data represented, and how can we reason with them? How can they be transformed into executable robot programs?

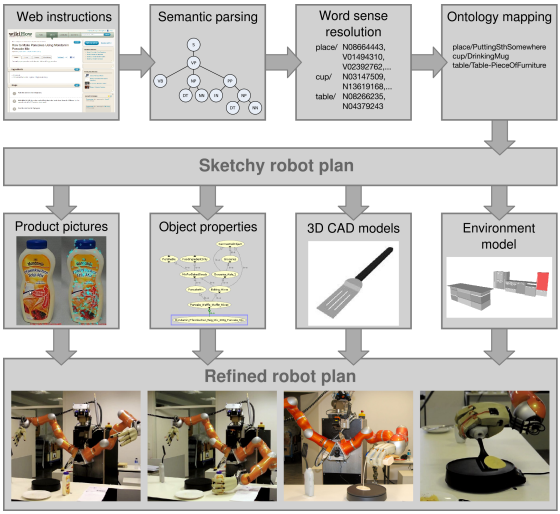

Figure 1.1 | The RoboHow processing pathway. A preliminary robot plan is created by parsing symbolic information (e.g. from a recipe). From this, a refined, executable motor plan is created.

- **Observation of human demonstrations**: How can a video image, as captured by the robot's sensors, be converted into a usable symbolic representation of scene objects and actors? How are these representations associated over time?

- **Constraint- and optimization-based control**: How can the robot generate fast and smooth movement that is constraint- or optimization-based?

- **Perception for robot action and manipulation**: How can the robot best use its sensors to extract useful information about objects in the environment? How can the robot learn task constraints?

- **Learning from interaction with a human**: Developing adaptive stiffness control to ensure grasp stability; learning of haptic interaction.

- **Plan-based control**: Developing a plan language to represent and specify robot behavior. How can the robot infer gaps in incomplete symbolic action specifications? How can the robot learn complex action and its subcomponents?

The work conducted at Leiden University, part of which is the result you are reading, concerned itself with representation and plan-based control. More specifically, the relationship between complex action in humans and robots, and the question of how the acquisition of sequential action could best be investigated and modeled were investigated. However, this was not the only focus of our research.

1.2.2 Cognitive work inside RoboHow

Over the course of the project, our group focused on several issues relevant for robot control. During the first year of the project, we developed a deep recurrent neural network model for the execution of sequential action [76]. Using an extension of the LEABRA framework [111, 112], we investigated the effect of layer size and architecture on network performance. We found that for relatively simple tasks, two-layer networks perform as well as deep networks, and that recurrence in a single layer is enough to learn simple sequential tasks.

Next, we investigated the flexibility of action plans generated by AI planners, and ways to improve this flexibility. Traditional planners such as STRIPS determine the set of actions required to reach a goal state from an initial state. However, should one of those actions fail, the goal state can no longer be reached. What would be the correct course of action to take? Standard planners would fail, and return control to a higher planning layer or human operator. Smarter control is needed for robots performing everyday action. For example, a cooking robot asked to make pancakes should not fail if the recipe calls for whole milk, but only skim milk can be found in the refrigerator. In other words, ingredient replacement is a necessary capability for planners in smart robots. To make this possible, we developed an open-source, ROS-based software component

that uses holographic reduced representations [117] to determine similarity between ingredients. By analyzing a large corpus of recipes, ingredients that are used in similar ways are considered to have a higher similarity coefficient. This component could readily be integrated in the robot architecture used in RoboHow.

Finally, we focused our attention on the acquisition of action sequences. In order to capture rich data, we adapted Nissen and Bullemer's serial response time (SRT) task [107] to a mouse cursor paradigm, allowing us to investigate predictive processes and context effects. Also, we manipulated the nature of the task by adapting it to a reinforcement learning task, in which participants were no longer cued by the stimuli. Instead, they were expected to explore all possible alternatives and learn the sequence by trial-and-error. The results of this research direction are described in the rest of this dissertation.

1.3 Learning sequential action

1.3.1 History

The acquisition of action sequences has been studied as far back as the 19^{th} century, when the first theories of sequential action argued that subactions in a sequence are triggered by the sensory effects of the previous subaction [69, 167]. Early critics, such as Münsterberg [102] noted that these *chaining models* lack the directional element required to guarantee the correct order of execution due to the bidirectional nature of associations. That is, if stimulus A activates stimulus B, then the activation of stimulus B is likely to activate stimulus A, leading to an infinite loop. Instead, he argued that the execution of action sequences relies on the acquisition of a *motor program*. Since, ample evidence has been published to suggest that this is indeed the case [59, 79, 87, 125, 129, 160].

One popular paradigm to study the acquisition of action sequences is the serial reaction time (SRT) task [107]. In this task, a visual stimulus appears in one of four locations, horizontally distributed on a computer

screen. Four buttons are located below the four possible stimulus locations, and participants are asked to press the button below the visual stimulus that appears as quickly as possible. Unbeknownst to participants, the sequence of stimuli presented could either be a deterministic, repeating sequence of length 10, or a randomly generated sequence. Comparing the two conditions, Nissen and Bullemer [107] found that speed-up over time was larger for the deterministic group, indicating learning of the sequence, both explicitly and implicitly.

Although this paradigm may be effective for investigating sequence acquisition, it has several limitations. Improved performance on the SRT task is shown to be caused by general motor speed-up as participants get used to the task, but also the learning of the sequence, evidenced by the difference in response times between a random condition and a deterministic, repeating sequence. It is thought that *predictive* processes are responsible for this speed-up—that is, action onset precedes stimulus onset. However, due to its discrete nature of responding, the standard SRT paradigm does not allow researchers to differentiate between predictive movement and associative learning mechanisms. Indeed, decreased reaction times may just be a reflection of efficient responding due to the strength of local memory traces, similar in nature to Hommel's *prepared reflex* concept [34, 61] (see also section 1.4).

Adapting the discrete nature of responses in the SRT task to a more continuous measure of behavior *does* allow for such a distinction to be made. Earlier studies have shown that hand or mouse tracking allows for the investigation of continuous, dynamic internal processes that reflect cognitive states [48, 146–148]. One way to adapt the SRT task to a trajectory paradigm, is to map the four stimulus (and response) alternatives to four corners of a computer screen (see Figure 1.2).

Several studies using this paradigm have validated the adaptation by replicating Nissen and Bullemer's [107] original findings [75, Chapters 4 and 5]. And because this paradigm allows investigators to look at movements in the inter-trial interval (ITI) and right before and after targets have been reached, it has been successfully used to investigate reactive and predic-

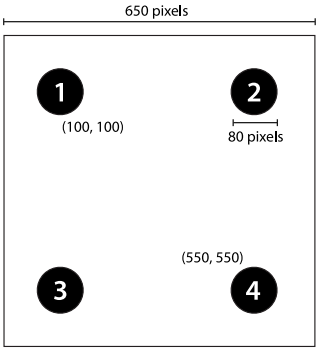

Figure 1.2 | A trajectory adaptation of the SRT task. Instead of discrete button presses, participants are required to move the mouse cursor to one of four targets. Adapted from [34].

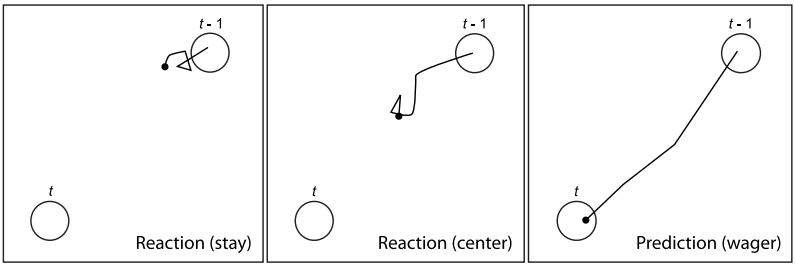

Figure 1.3 | Different types of movement during a trajectory SRT task. Participants with reactive movements tend to stay at the stimulus that was last touched, or move to the center when uncertain about the next stimulus. Participants with sequence knowledge make predictive movements toward the next stimulus during the ITI. Adapted from [34].

tive movements (illustrated in Figure 1.3), as well as context effects (e.g. [34, 75], but also Chapters 4 and 5).

But even with the rich data that can be extracted from mouse trajectories, what exactly is the mechanism by which we acquire these sequences remains unclear. One approach to tackling these questions is known as

computational modeling. In this technique, a formal model in the form of a computer program[1] is designed to reflect the cognitive processes and their interactions that are thought to be responsible for a phenomenon. By comparing the behavior of human participants with the behavior expressed by the model and by investigating which parameters and variables cause systematic changes in model behavior, it is possible to infer properties of underlying mechanisms in humans.

1.3.2 A reinforcement learning account of sequential action learning

One class of models that can give a computational account of the process of sequence acquisition are known as *reinforcement learning models*. Reinforcement learning is an area of machine learning dating back to the 1950s, inspired by Thorndike's *operant conditioning*. Unlike *supervised learning*, another popular approach in machine learning in which a predictive model is trained using an external knowledgable supervisor, reinforcement learning models learn by interacting with the environment and receiving feedback on their produced actions in the form of a positive or negative reward. So, instead of being told what to do by a supervisor, reinforcement learners have to discover that autonomously by trial-and-error [153]. A simple reinforcement learning model consists of five basic elements:

1. The *agent*, which represents a learning agent that can sense its surroundings to determine the state of itself and the environment, do something with that information based on its knowledge, and produce behavior that changes its state.

2. The *environment*, which represents the current state of the agent's observable world.

[1] Or in the form of mathematical concepts and language, in which case this is known as *mathematical modeling*.

3. A *reward function*, which maps each state–action pair to a reward, indicating the desirability of that state. In other words, it defines the reward r when action a is taken in state s.

4. The agent's *policy*, defining what action the agent should take when observing a certain state.

5. (optional) A model of the environment. Some RL models work using a model of the environment, e.g. to estimate the transition probability from state s to state s' when action a is taken. These RL models are called *model-based*, others are known as *model-free*.

The goal is to discover the policy that maximizes accumulated reward. At the start of the learning process, no information is known about the rewards associated with actions taken in certain states; it is the task of the learner to *explore* the environment in order to learn these state–action reward values, and update its policy accordingly. After a certain amount of interactions with the environment, the agent can somewhat reliably predict which actions to take in which states in order to maximize reward. In other words, it can *exploit* the knowledge it has gathered.

This trade-off between exploration and exploitation has a large influence on the speed and accuracy of learning. If an agent would mostly exploit the knowledge it already has, it runs the risk of consistently choosing actions that produce small rewards, simply by virtue of having encountered them before having had the chance to try other actions. On the other hand, an agent that only explores does not make good use of the information about state–action rewards it has gathered over time.

Human participants in reinforcement learning tasks (i.e. tasks in which the participant is rewarded or punished in a systematic or probabilistic way for performing an action in a given state) vary widely in how fast they learn, and how sensitive they are to reward (see Chapter 4 for a good example). By comparing the performance of reinforcement learning models with the behavior produced by human participants, these differences can be identified and quantified. But these individual differ-

ences in learning action sequences do not require computational modeling to be made visible.

1.4 Executive control modes in sequential action

There is evidence to suggest that implicit learning takes place in the SRT task, demonstrated by the fact that amnesic patients could learn the sequence without being able to verbalize or regenerate the sequence, while healthy participants nearly all gained explicit sequence knowledge [107]. Clearly, the sequence could be learned either with or without explicit knowledge. However, several studies have shown that explicit sequence knowledge strongly correlates with higher accuracy in manually reproducing the sequence [34], a higher proportion of predictive movements [34, Chapter 5], and a reduced stimulus–response compatibility effect [60, 159, 160].

The development of explicit sequence knowledge is considered to be a consequence of the shift from a reactive, stimulus-based control mode to a predictive, plan-based control mode that takes place during learning [160]. Under stimulus-based control, the cognitive system relies on "prepared reflexes" [61] to respond to highly response-compatible stimuli in an automatic fashion. Under this executive control mode, control is delegated to the stimulus, and little of the sequence is actually learned. Under plan-based control, plan-related representations are internally generated, thereby reducing the reliance on stimuli. As such, this control mode is less affected by stimulus-related properties such as frequency or stimulus–response compatibility [160]. In Chapter 5 of this dissertation, determinants of these executive control modes and their behavioral effects are investigated and discussed.

1.5 Dissertation outline

The contents of this dissertation are divided into two parts. In the first part, an overview is presented of the theoretical similarities and differ-

ences between human action control and robotic action control. In the second part, empirical studies are presented that give an account of human sequence acquisition.

In **Chapter 2**, complexities surrounding everyday action are explained. First, in order for an agent to perform complex action, it is necessary to integrate symbolic and subsymbolic representations. Purely symbolic information, such as defined in a recipe for example, is not enough to actually execute the action. Such action instructions are usually underdefined, and it is necessary for the agent to fill in the necessary motor parameters in order to successfully complete the action. Second, after initiating a motor action by preparing the motor plan, feedback from the environment must have to be integrated in order to monitor successful execution and adapt the motor plan to changing circumstances. Also, while complex action is often hierarchical in nature, it is still unclear if the cognitive representations need to be hierarchical as well, and what the implications of that might be.

The histories of cognitive robotics and cognitive psychology, and how these fields have interacted historically are discussed in **Chapter 3**. Concepts like feedforward and feedback control systems are common to both fields, and learning mechanisms such as reinforcement learning and motor babbling are successful in explaining or producing learning behavior in both humans and robots. Other theories from cognitive psychology are increasingly used in robotics, such as Biederman's recognition-by-components theory, that tries to explain how humans infer object affordances based on an object's geometric properties.

Human everyday action is characterized by its sequential nature. As action sequences are learned, people get faster at performing them, which is easily shown by comparing people following a deterministic, repeating sequence with people following a random, unpredictable sequence [107]. However, most studies using this so-called SRT paradigm use discrete button presses, response times and response accuracy, leading to several limitations [34, 146]. In **Chapter 4**, a more informative trajectory task was used to reveal dynamic internal processes. By frequently and ac-

curately capturing participants' mouse position, predictive movements and context effects can be analyzed. However, even with trajectory analysis, the SRT paradigm used might not be a valid analogy for real-world sequence acquisition. Instead of responding to visual stimuli that are flashed on a screen, everyday sequence learning might be better regarded as exploratory—that is, people try things and receive positive or negative feedback regarding the outcome. To investigate this, the SRT paradigm was adapted to a reinforcement learning paradigm, in which participants could explore possible alternatives and receive feedback in the form of points. Interestingly, resulting scores were non-normally distributed, with a distinct low-performing group and a high-performing group with almost perfect performance. Several model-free reinforcement learning models were fit to participants' data to see if any of them could accurately model their performance, but the high-performing group easily outperformed all of them. Apparently, humans use other algorithms than simple model-free reinforcement learning to acquire action sequences.

Due to power limitations caused by the low number of participants, as well as the exploratory nature of this study, we could not say anything about the cause of the large difference in performance. However, in **Chapter 5** a larger group of participants was recruited, and several additional measures were collected. It was hypothesized that differences in performance could be attributed to cognitive limitations such as IQ or visuospatial working memory, or strategies or preferences for action plan formation. In a reinforcement learning paradigm, action plan formation was found to be strongly associated with explicit knowledge sequence, and predicted by both IQ and visuospatial working memory, but not by personal preferences. It seems that sequential action in exploratory paradigms is limited by cognitive capacity.

Whereas reinforcement learning models can account for the learning of action sequences (although not with the same performance as humans), they do not account for observed motor behavior during the trajectory tasks discussed earlier. More specifically, participants in the studies discussed in Chapters 4 and 5 were observed to move their mouse to the

center of the screen under conditions of uncertainty. This has also been observed in other studies (e.g. [34, 38]) and seems to be an efficient strategy as it minimizes the distance to possible targets. To investigate the nature of this behavior, in **Chapter 6** an artificial neural network embedded in a virtual robot was evolved using different levels of prediction quality. It was hypothesized that lower prediction quality would cause centering behavior in a cursor controlled by the neural network. Indeed, this seemed to be the case, confirming that this strategy is an efficient one under conditions of uncertainty.