Control of complex actions in humans and robots

Kleijn, R.E. de

Control of complex actions in humans and robots

Roy de Kleijn

Control of complex actions in humans and robots

ISBN: 978-94-6332-258-4

Printed by GVO Drukkers & Vormgevers B.V., Ede, The Netherlands

Printed on 90 gr Biotop,
typeset in Calluna, Myriad Pro and Source Code Pro

Control of complex actions in humans and robots

Proefschrift

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden,
op gezag van Rector Magnificus
prof. mr. C.J.J.M. Stolker,
volgens besluit van het College voor Promoties
te verdedigen op donderdag 23 november 2017
klokke 16.15 uur

door

Roy Edward de Kleijn

geboren te 's-Gravenhage
in 1982

Promotor

prof. dr. Bernhard H. Hommel

Co-promotor

dr. George E. Kachergis (Radboud Universiteit)

Promotiecommissie

prof. dr. Birte U. Forstmann (Universiteit van Amsterdam)
prof. dr. Verena V. Hafner (Humboldt-Universität zu Berlin)
prof. dr. Sander T. Nieuwenhuis
prof. dr. Aske Plaat
prof. dr. Frank van der Velde (Universiteit Twente)

The preparation of this dissertation was supported by the European Commission (EU Cognitive Systems project ROBOHOW.COG; FP7-ICT-2011).

Contents

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?

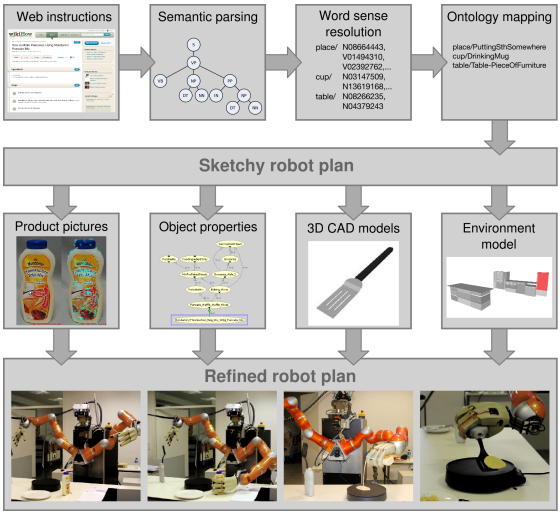


Figure 1.1 | The RoboHow processing pathway. A preliminary robot plan is created by parsing symbolic information (e.g. from a recipe). From this, a refined, executable motor plan is created.

- **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-

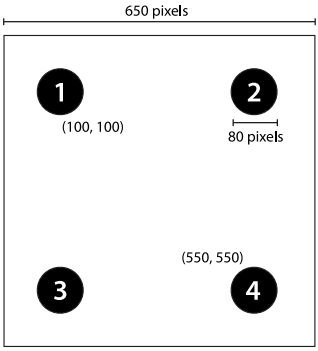


Figure 1.2 | A trajectory adaptation of the SRT task. Instead of discrete button presses, participants are required to move the mouse cursor to one of four targets. Adapted from [34].

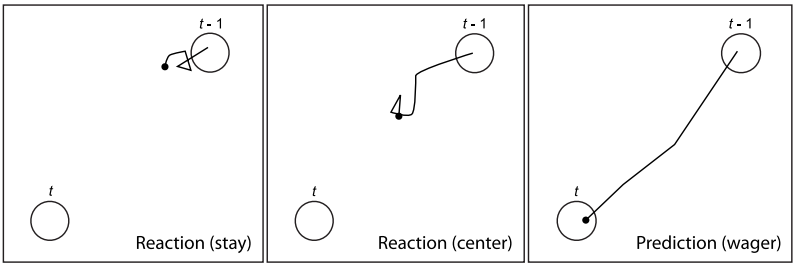


Figure 1.3 | Different types of movement during a trajectory SRT task. Participants with reactive movements tend to stay at the stimulus that was last touched, or move to the center when uncertain about the next stimulus. Participants with sequence knowledge make predictive movements toward the next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

Part A

Theoretical foundations

CHAPTER 2

What's so special about human action?

Over a relatively short time span, the discipline of robotics has advanced from producing industrial non-autonomous, repetitive machines to semi-autonomous agents that will be able to function in a dynamic, human-driven world. Simple examples include robotic vacuum cleaners such as Roombas, but more flexible and autonomous humanoid robots are currently under development (e.g. the RoboHow project [31]). As robots perform more and more everyday human activities such as household chores, interacting with humans, and thereby almost becoming citizens in our societies, we believe that psychologists can provide relevant knowledge about human behavior that is generalizable to robots.

Like early approaches to AI, traditional cognitive psychology considers behavior (of biological or artificial agents) to emerge from discrete series of cognitive operations that take information from the environment (registered by sensory organs or artificial sensors), process this informa-

This chapter is an adaptation of the article *de Kleijn, R., Kachergis, G., & Hommel, B. (2014). Everyday robotic action: Lessons from human action control.* Frontiers in Neurorobotics, 8:13.

tion in more or less complex ways, and eventually manipulate something in the environment as a result of this processing. In psychology, this *discrete, serial processing model* of cognition has been successful in explaining various psychological phenomena, but for some reason most research has focused on the early and middle stages of this process, leaving action and motor control far behind. Indeed, psychology as an autonomous science has historically shown an impressive neglect of the study of action and motor control, to the extent that it has even been called the "Cinderella of psychology" [127].

Fortunately, however, more recent approaches have emphasized the role of action not only as an output function but as a precondition and basic ingredient of human cognition (e.g. [28, 65, 110]). These recent approaches have criticized the traditional sequential-stage account of human behavior for analyzing action as a consequence of stimuli. They argue that action is more aptly characterized as people's means to produce stimuli (desired outcomes), rather than as a means to respond to stimuli [63]. Moreover, actions are more than mere ballistic outputs: they are events that unfold in time and that must be structured in such a way that their outcome satisfies current needs and goals. Consider, for example, the act of tea-making, which consists of a number of components: (1) boiling water, (2) putting a tea bag in a teapot, (3) pouring the boiling water in the teapot, and (4) pouring the tea in one or more cups. Executing these different components in such a way that the intended goal is eventually achieved requires planning.

In the following, we will provide a brief overview of available psychological insights into how this planning works in humans, and how these insights might inform the creation of robotic everyday action systems. At the moment, although robot actions mimic human action, the control systems are in fact quite different. We will confine our discussion to four principles that we think could be particularly beneficial for robot control: (1) the integration of symbolic and subsymbolic planning of action sequences, (2) the integration of feedforward and feedback control, (3) the clustering of complex actions into subcomponents, and (4) the

contextualization of action-control structures through goal representations.

2.2 Integrating symbolic and subsymbolic planning

In contrast to the ballistic, single-step actions that participants in laboratory experiments often carry out, everyday action commonly consists of multiple components, as in the tea-making example. In AI and robotics, multi-component actions are commonly planned at a symbolic level, with each action component being represented by an arbitrary symbol or function. The STRIPS (Stanford Research Institute Problem Solver) planner [47] is a famous example: it serves to translate an initial state into an intended goal state by determining the subset of actions (defined as a symbolically described relation between sets of pre- and post-conditions) needed to do so. The format of all representations involved is symbolic allowing all goals and actions to be represented in basically the same way, although they can be arbitrarily linked to subsymbolic trigger states. This uniformity allows for a very efficient planning process, as action components can be easily manipulated and exchanged until the entire plan is optimal.

Symbolic action planning of this sort is consistent with early models of human action planning, which typically connected underspecified symbolic action representations with subsymbolic trigger states that took care of timing. For instance, Washburn considered that later action components might be triggered by the perception of the execution of the previous one: "If the necessary stimulus for pronouncing the last syllable of a series were the muscular contractions produced in pronouncing the next to the last syllable, then the proper sequence of movements would be insured" [167, p. 9]. Along the same lines, James [69] suggested a *serial chaining model*, according to which each action component is triggered by the perception of the sensory feedback produced by the previous component. Accordingly, learners will create associations linking the motor patterns and their sensory consequences in a chain-like fashion.

As more studies were conducted, however, it was found that chaining accounts of sequential behavior cannot account for several empirical observations. In a seminal paper, the neurophysiologist Lashley [87] pointed out that the serial chaining models of the time were not adequate, because: (1) movements can still be executed if sensory feedback is impaired; (2) some movements are executed too quickly to have time to process feedback from preceding actions, and (3) errors in behavior suggest the presence of predetermined action plans [129]. Rosenbaum et al. [129] added further arguments against a chaining account of sequential action. For example, the time needed to initiate an action is a function of its complexity [59, 79, 125], suggesting that the agent anticipates later action components before beginning to execute the first.

Along the same lines, Cohen and Rosenbaum [30] (for another good example see [163]) had participants grasp a vertical cylinder placed on a platform and move it to another platform that was either higher or lower than the initial location. The researchers determined the vertical location of the grasp, and found that the grasp location was dependent on the expected end state. More specifically, subjects tended to choose a lower grasp location when bringing the cylinder to a higher position, and vice versa. Likewise, when subjects were asked to move the cylinder back to its starting position, they tended to grasp it in the location where they grasped it before. This *end-state comfort effect* suggests that people anticipate the position that they will assume after the action has been completed.

The same conclusion is suggested by studies on context effects in speech production. For example, people round their lips before pronouncing the *t* in the word *tulip*, in anticipation of pronouncing the *u* later in the sequence [14, 35, 49, 126]. This does not seem to be a purely epiphenomenal property of human action; one can easily see how this produces more efficient, smoother speech, and a more careful use of the human speech-production "hardware". An analogous action blending effect occurs when people reach for objects: people adaptively flex their fingers while moving the hand toward an object [71], and has been observed to

develop when sequentially moving a cursor through a learned series of stimuli [75]. Compared to typical step-wise robotic motion, this action blending seems to be more efficient, using predictive motion to minimize the time and energy required to achieve the goal.

Further insights into human sequential action planning come from Gentner et al. [51], who conducted a photographic study of a skilled typist. Using high-speed photography, they analyzed the hand movements of a 90-wpm typist, and found that the typist's hands were moving continuously, with fingers starting to move toward a destination before several preceding characters were to be typed. In fact, for 96% of all keystrokes, movement was initiated on average 137 ms before the preceding keystroke was completed, and for 21% the movement was initiated before the preceding keystroke was initiated. Larochelle [86] presents a similar but more extensive study, analyzing the typing of four professional typists while they typed either words or non-words, of which half were typed with one hand, and the other half with two hands. In more than half of the trials the movement was initiated before completion of the previous keystroke for two-handed trials.

These interactions between early and later sequence elements cast doubt on a simple chaining theory of sequential action. Rosenbaum et al. [129] interpreted these findings as evidence that sensory feedback is not a necessary component for action sequencing, in keeping with the conclusion of Lashley [87]. They argued that "the state of the nervous system can predispose the actor to behave in particular ways in the future," [129, p. 526], or, there are *action plans* for some behaviors. And yet, studies on spontaneous speech repair (e.g. [103]) also show that people are very fast in fixing errors in early components of a word or sentence, much too fast to assume that action outcomes are evaluated only after entire sequences are completed. This means that action planning cannot be exclusively feedforward, as Lashley [87] seemed to suggest, but must include several layers of processing, with lower levels continuously checking whether the current action component proceeds as expected. In other words, action planning must be a temporally extended process in which higher-

level representations to some extent provide abstract goal descriptions, which must be integrated with lower-level subsymbolic representations controlling sensorimotor loops. The existence of subsymbolic sensorimotor representations would account for context and anticipation effects, as described above. In the more general field of knowledge representation, some authors even take it one step further, positing that subsymbolic, sensorimotor representations are *necessary* for higher-level symbolic cognition. For example, Barsalou's [11, 12] *perceptual symbol systems theory* defines cognition as embedded in the world, stating that agents form grounded models via perception and interaction with their environments. With these models, the representation of abstract concepts can be implemented using grounded perceptual symbols. The empirical support for theories like these motivate the notion that both symbolic and subsymbolic representations can (and should) work together to account for human cognition.

A good example for an action planning model that includes one symbolic and one subsymbolic level is the typewriting model suggested by Rumelhart and Norman [131]. To control typing the word "WORD," say, the model would assume that the symbolic (or "semantic") representation WORD would activate motor units controlling the finger movements required to type "W," "O," "R," and "D" in parallel. This parallel activation allows for crosstalk between the different units, which would account for context effects and anticipations. At the same time, the activated units are prevented from firing prematurely by means of a forward-inhibition structure. That is, each unit is inhibiting all following units in the sequence (so that the "W" unit inhibits the "O," "R," and "D" units, the "O" unit the "R" and "D" units, and the "R" the "D" unit) and releases that inhibition only once they are executed. The dynamics of these inhibition and release processes automatically produce the necessary sequence. It is thought that such activation and inhibition processes play a role even in young infants [165]. Immediate feedback, though not explicitly addressed by Rumelhart and Norman [131], could serve to repair the actions controlled by particular units, but the feedback would not be needed to

produce the sequence—a major advantage over chaining models. For an overview of similar models and other action domains, see [90].

The main lesson for robotic everyday action control is that purely symbolic planning may be too crude and context-insensitive to allow for smooth and efficient multi-component actions. Introducing multiple levels of action planning and action control may complicate the engineering considerably, but it is also likely to make robot action more flexible and robust—and less "robotic" to the eye of the user.

2.3 Integrating feedforward and feedback mechanisms

In perfectly predictable environments such as industrial construction halls, there is hardly any need for feedback mechanisms. Indeed, early industrial robots, such as Unimate, could rely on fully preprogrammed feedforward control for repetitive multi-component actions such as picking up and manipulating objects [56]. However, real-life environments are much too unpredictable to allow for purely feedforward control. Considering that purely feedback-based control is often much too slow to allow for real-life human action, it is unsurprising that human action control seeks for an optimal integration of feedforward and feedback mechanisms.

One of the earliest studies into feedforward planning was conducted by Henry and Rogers [59], who compared reaction times of participants performing a simple finger movement to reaction times of a moderately complex arm movement (reaching and grasping) in response to a stimulus. The authors found that participants performing the more complex movement showed a 20% increase in reaction time, with as much as a 25% increase for even more complex movement. This suggests the existence of feedforward action planning prior to action execution.

Linguistic studies have shown a similar effect. Eriksen et al. [44] had participants read aloud two-digit numbers consisting of a varying number of syllables. Longer numbers were shown to have a longer onset delay. In order to account for the possibility that factors other than motor plan-

ning played a role, participants were given the same task with a delay between stimulus onset and vocalization. Here, the effect disappeared, again providing evidence for pre-execution action plan formation.

However, while it may be tempting to conclude that an action plan is formed completely before action onset, incremental approaches to sequential action posit that this is not the case. Palmer and Pfordresher [113] argued that it is unlikely for actors to have access to all elements in a long sequence, as this would place unnecessarily large demands on memory—just think of a conductor starting to conduct a 4-hour Wagner opera. Instead, planning and execution co-occur in time, limiting access to sequence elements that appeared much earlier or that lie far in the future. Evidence for this was indeed found by Sternberg et al. [151], in which six participants prepared and produced sequences of mono- or tri-syllabic words. In addition to the length effect discussed above, preparation times were found to increase with length of the word sequence until approaching asymptote (which was 10.3 ± 0.6 words for sequences of mono-syllabic words and 6.4 ± 0.9 words for tri-syllabic words). This suggests that plan formation and execution occur simultaneously, at least for longer sequences of actions, with a limited capacity.

However, feedforward mechanisms alone cannot account for such complex action as our tea-making example. A complete feedforward program would need to incorporate numerous unknown parameters, such as the exact location and physical properties (e.g. weight) of all necessary objects. The prior unavailability of such parameters is not the only reason feedback mechanisms might be helpful. Some parameters might be possible to include in a feedforward program, but would simply be more efficient or optimal if filled in online, such as grip strength. Even if all this information were available, an actor still needs to be able to correct possible—sometimes inevitable—perturbations in action execution.

Indeed, it seems that the presence of uncertainty (i.e. unavailability of necessary parameters) increases the importance of feedback mechanisms. Saunders and Vijayakumar [137] fitted participants with a prosthetic hand that could provide vibrotactile feedback. Using this prosthetic hand, they

were asked to manipulate objects of different weights. Manipulating both feedforward uncertainty by adding an unpredictable delay in the prosthetic hand and feedback information by manipulating vibrotactile feedback, they found that performance decreased when feedback was removed in situations with feedforward uncertainty. This illustrates that human action emerges from the interaction of feedforward and feedback mechanisms.

Integrating feedforward and feedback mechanisms holds the promise to get the best from both worlds. Feedforward mechanisms are likely to determine the necessary action components and pre-load at least some of them before initiating the action [59], and to selectively tune attention to stimuli and stimulus dimensions that are relevant for the task [64]. Feedback processes, in turn, provide excellent accuracy—often at the cost of speed [141]. These strengths and weaknesses have motivated hybrid models claiming that feedforward mechanisms provide the skeleton of action plans which leave open slots for parameters provided by feedback processes [53, 64, 140].

A particularly good example of this kind of interaction is provided by the observations of Goodale et al. [55]. In a clever experiment, participants were asked to rest their hand on a platform and point to a visual target presented at a random location on an imaginary line in their right visual field. The participants were not told that in half of the trials the target changed location during the first saccade. The authors found that participants would successfully point to the target on these trials without even being aware of the location change, and without additional delay. As feedforward programming is thought to take time, a fast and online feedback mechanism of which participants are unaware has to be responsible for this finding. After this study showing online adaptation of hand velocity, Prablanc and Martin [120] found that these results generalize to two dimensions. Using stimuli presented on a screen, it was found that both the velocity and trajectory of the hand were adjusted online. This demonstrates that action is the result of a preprogrammed action plan (the initial movement of the hand) combined with online adaptation to

reach goal requirements. Interestingly, such a division of labor fits well with the architecture of the human brain, which includes both a slow, cognitively penetrated ventral route from perception to action and a fast dorsal sensorimotor loop (for a broader overview, see [97]).

It is clear that both feedforward and feedback mechanisms are responsible for producing complex action, but a number of questions remain unanswered. Are feedforward processes always responsible for certain actions? How are these plans learned, and how do people know when to apply them? How does feedback on a lower level result in action replanning on a higher level, and does this require conscious intervention? What is the division of labor between feedback and feedforward mechanisms? How fluid is it—how hierarchical?

We know that with practice, the roles of feedback and feedforward processes change. In a standard rapid aimed limb movement paradigm, participants are asked to perform a manual action in order to reach a target. During such tasks, the response can be regarded as having two elements: (1) a ballistic primary movement, thought to be controlled by a feedforward mechanism, and (2) a secondary, corrective movement, thought to be caused by a feedback mechanism. Pratt and Abrams [121] used such a paradigm to investigate the effect of practice on the weight of primary and secondary movements. Participants were asked to repeatedly move a visual cursor to a target location using wrist rotation. With more practice, the percentage of time spent in the first movement increased, while time spent in the second movement decreased. As the first movement is feedforward-controlled, this suggests that practice reduces the need of feedback control, as the feedforward process becomes more accurate. But will this learning generalize to new situations with similar action requirements, and is it long-lasting?

To investigate the relationship between practice and feedback control, Proteau et al. [123] had participants practice an aiming task on either 200 or 2000 trials and found that, when visual feedback was taken away, participants who had more practice were more impaired by the removal of feedback. This is not what one would expect if practice simply shifts con-

trol to feedforward processes. Subsequent research has shown that, with practice, higher peak velocities are reached in the early phase of movement, thereby leaving more time for corrective submovements based on feedback. Thus, instead of a shift from feedback control to feedforward control, feedback processes seem to be optimized as a result of practice [40, 78, 123].

While the first generation of robots and other intelligent systems had a strong preference for feedforward control, not in the least because of the rather predictable environments they were implemented in, some modern systems rely heavily on feedback control to perform actions—especially humanoid systems operating in real-world scenarios. This is likely to work as long as action production in such robots is slower than the feedback loops informing them [118], but progress in action mechanics is likely to make hybrid feedforward/feedback systems an attractive alternative in the near future.

2.4 Hierarchical action representation

Human actions can often be described in a hierarchical fashion: "Going on vacation" implies action such as "packing my bags," "getting the car," "loading it," "driving down to city X," et cetera. Many authors have taken that to imply that action control is hierarchical as well. According to Lashley [87], only a hierarchical organization of actions and action plans can provide the opportunity to have the same motor acts acquire different meanings, depending on the context in which the motor act is performed. In Miller's [96] seminal book, action plans are even hierarchical by definition: "A Plan is any hierarchical process in the organism that can control the order in which a sequence of operations is to be performed" [96, p. 16]. And yet, while it is certainly uncontroversial that it is possible to *describe* actions as hierarchical, this need not have any implication for the cognitive organization of actions. As Badre [10] argues, "the fact that a task can be represented hierarchically does not require that the action system itself consist of structurally distinct processing levels" [10, p. 193]

(see also [80]). Moreover, it is not always clear what authors mean if they say that actions are organized in a hierarchical fashion.

Uithol et al. [162] noted that there are at least two ways to look at hierarchical action. These two ways differ in what are considered to be the different levels in such a hierarchy. One way to look at action hierarchies is the view of part-whole relations. In this account, each level in the hierarchy exists solely as the sum of lower-level units. In other words, an action unit such as "get a pan for pancake making" consists of the subunits "open the cupboard," "take pan from cupboard," "place pan on counter," and "close the cupboard." It should be clear that when all subordinate units are present, the superordinate unit "get a pan" is also present, as it is identical to the sum of its parts. Uithol et al. [162] argue that this kind of hierarchy does not provide an explanation of the complex action; it merely provides a thorough description of the to-be-explained action, in which higher levels are more complex than lower levels. It also does not give information about the causal relationship between the different levels in the hierarchy, as you cannot consider an element to be the cause of its own parts. Another restriction of this type of hierarchy is that it can only accommodate levels that are of a similar nature. That is, actions can only be divided into sub-actions, not into objects or world states.

Another way to view hierarchies is to see the different levels as representing causal relations between the levels. In this approach, units on a higher level causally influence units on a lower level. In this type of hierarchy, lower-level units can be modulated by higher-level units. In contrast with the part-whole hierarchy, lower levels are not necessarily less complex than higher levels. Goals that are formulated as simple and propositional states can be the cause of more complex elements. Using this hierarchical approach also opens up the possibility of states or objects being the cause of an action, as it does not have the limitation of requiring action-type goals.

Uithol et al. [162] proposed a new model, in which the fundamental foundation for the hierarchical structure is not cause-and-effect (i.e. goals

cause motor acts), or complexity (i.e. complex motor acts such as grabbing a pan consist of simpler acts such as flexing fingers and grasping the handle), but temporal stability. In this view, stable representations can be considered goal-related, while more temporary representations reflect motor acts on different levels, not unlike the more enduring conceptual representations and the less enduring motor units of Rumelhart and Norman's [131] model discussed above. However, this representation proposal does not include a model of how the hierarchies within a task are abstracted and learned from experience, nor of how they may be shared across tasks despite requiring different parameterizations.

Botvinick and Plaut [18] tackled some of these issues, pointing out that not only is it unclear how existing hierarchical models learn hierarchies from experience, but also that most theoretical accounts lead to a circular reference: acquiring sequence knowledge relies on the ability to identify event boundaries, which in turn requires sequence knowledge. A further problem is sequencing in hierarchical structures; many models (e.g. [66, 131]) solve that by means of forward inhibition, but this only works on units at the lowest level of a hierarchy. Botvinick and Plaut [18] offered a recurrent connectionist network model that helps to avoid these problems. Using computer simulations they showed that such a network, which contains no inherent hierarchical structure, can learn a range of sequential actions that many consider hierarchical. The hierarchy, they argued, emerges from the system as a whole. The network they used is a three-layer recurrent network, with an input layer representing held objects and fixated objects, an output layer representing actions to be taken, and a hidden layer (with recurrent connections) for the internal representation. Having trained this network on a routine complex task (making coffee or tea), they showed that it can perform complex action that can be considered hierarchical in nature (e.g. varying orders of subactions leading to the same outcome) without relying on a hierarchical system architecture. The network also showed slips of action when the internal representation layer was degraded, as well as other action errors found in empirical studies, although Cooper and Shallice

[33] suggest that the relative frequency and types of errors shown by the recurrent model do not match human subjects.

We believe that architectures offering such hierarchical behavior, without necessarily being hierarchically structured, can provide robots with the needed flexibility to function in a dynamic, human-driven world. Botvinick and Plaut's [18] model seems to be able to account for some aspects of flexible behavior, but more complex and biologically inspired models such as LEABRA [76, III] promise to generalize to other tasks, as well as being able to learn relatively fast, two aspects of human behavior we consider essential to emulate in robot behavior.

2.5 Contextualizing action control

As pointed out above, one of the reasons why Lashley [87] considered action representations to be necessarily hierarchically organized was the fact that the meaning and purpose of action components vary with the goal that they serve to accomplish: while making a kicking movement with your right leg can easily be replaced by moving your head sideways when trying to score a goal in a soccer game, that would not be a particularly good idea when performing a group can-can on stage during a performance of *Orpheus in the Underworld*. In other words, goals are needed to *contextualize* action components. In AI, robotics, and some information-processing approaches in psychology, the main function of goal representation is to guide the selection of task components, including stimulus and response representations or perception-action rules. In traditional processing models, like ACT-R or Soar [3, 85], goal representations limit the number of production rules considered for a task, which reduces the search space and makes task preparation more efficient [33]. Moreover, goals commonly serve as a reference in evaluating an action, when comparing the current state of the environment with the desired state [96].

This practice was challenged by Botvinick and Plaut [18], who pointed out at least two problems with goal representations in cognitive models.

First, goals themselves may be context-dependent. The goal of cleaning the house may have rather different implications depending on whether it serves to satisfy the expectations of one's partner or to prepare for a visit of one's mother-in-law. Likewise, the goal of stirring will produce somewhat different behavior depending on whether one is stirring egg yolks or cement. Most models that postulate the existence of goals do not allow for such context dependence. Second, it is argued that many everyday activities do not seem to have definable, or at least not invariant goals; just think of playing a musical instrument or taking a walk. The authors demonstrated that goal-directed behavior can be achieved without the explicit representation of goals. In the previously mentioned simulation studies with recurrent neural networks, they were able to simulate goal-directed actions that operate very much like Miller et al.'s [96] TOTE units, without any need to represent the goal explicitly. Obviating the need for representing goals, such a model could be applied to behavior with non-obvious goals, such as taking a walk as a consequence of feeling restless or having the thought of fresh air [18].

Cooper and Shallice [33] took issue with this non-representationalist account of goals, giving at least two reasons why goals *should* be implemented in cognitive models. First, goals allow for the distinction between critical and supporting actions. When making pancakes, the subaction of adding egg to the mixture consists of picking up an egg, breaking it (above the bowl), and discarding the empty shell (not above the bowl). It should be clear that the breaking of the egg is the most important action in this sequence. Dissociating important actions from less important actions can account for skipping unnecessary steps. When applying butter to two slices of toast, it is not necessary to execute the supporting actions "discard knife" and "pick up knife" between the two executions of the "butter toast" action program. Second, the implementation of goals would allow for subactions that serve the same purpose to be interchanged. For example, flipping a pancake by flipping it in the air or flipping it using a spatula would both be perfectly good methods for pancake flipping, and the shared goal allows these actions to be in-

terchanged. Models without goal representation can only show this behavior if they are explicitly trained on all the alternative actions that can be taken. To make the realization that a set of actions are equivalent for achieving a goal, a model would in essence have to contain a representation of that goal.

Interestingly, however, goal representations (whether explicit or implicit) can play an important role in contextualizing cognitive representations. Most representational accounts assume that representations of stimulus and action events are invariant. The need to contextualize representations (i.e. to tailor them to the particular situation and task at hand) thus seems to put the entire burden on the goal, so that the explicit representation of the goal seems to be a necessary precondition for adaptive behavior. But, from a grounded cognition perspective, it seems that alternative scenarios are possible. In a grounded cognition framework, the representation of objects and object categories takes an embodied form, using modal features from at least the visual, motor, and auditory modalities [122]. For example, the concept of apple would be represented by a network of visual codes representing `<green>` and `<round>`, but also the auditory `<crunchy sound>` of biting into it. The embodied cognition framework has already been successfully implemented in robot platforms such as iCub, and shows stimulus compatibility effects similar to those that can be observed in humans [93, 115].

Similarly, according to the Theory of Event Coding [65], events are represented—like objects—in a feature-based, distributed fashion. This will mean that the aforementioned apple would be represented by a network of codes representing not only its perceptual features such as `<greenish>` and `<round>`, but also other properties such as being `<edible>`, `<graspable>`, `<carryable>`, `<throwable>`, etc. In this view, one of the main roles of goals is to emphasize (i.e. increase the weight of) those features that in the present task are of particular importance. This means that when hungry, the feature of being `<edible>` will be primed in advance and become more activated when facing an apple, while the feature of `<throwability>` will become more important when being

in danger and trying to defend oneself. Several studies have provided evidence that goals are indeed biasing attentional settings toward action-relevant feature dimensions (e.g. [45, 83, 172]), suggesting that the impact of goals goes beyond the selection of production rules and outcome evaluation. Interestingly, this kind of "intentional weighting" function [95] can be considered to represent the current goal without requiring any explicit representation, very much along the lines of Botvinick and Plaut's connectionist model [18].

Another potential role of goals is related to temporal order. In chaining models, the dimension of time was unnecessary because the completion of each component automatically "ignites" the next component. The same holds for current planners in cognitive robotics, which commonly fix the order of action subcomponents (e.g. CRAM [13]). But action plans may follow a more abstract syntax instead, much like how syntactic constraints of natural languages allow for various possible sequences. For instance, again consider the process of making tea. With the possible exception of true connoisseurs, it doesn't make any difference for most tea drinkers whether one puts the tea or the water into the cup first; i.e. the order of these two subactions is interchangeable. A truly flexible system would thus allow for any of these orders, depending on whether water or tea is immediately at hand. While a chaining model would not allow for changing the original order, a more syntactic action plan would merely define possible slots for particular subcomponents (e.g. [128]), so that the actual order of execution would be an emerging property of the interaction of the syntactic plan and the situational availability of the necessary ingredients.

These considerations suggest that robotic systems need to incorporate at least some rudimentary aspects of time and temporal order to get on par with humans. Along these lines, Maniadakis and Trahanias [94] have propagated the idea that robotic systems should be equipped with some kind of temporal cognition, be it by incorporating temporal logic or event calculus. Indeed, recent robotic knowledge representation systems, such as KnowRob [156], do possess the ability to do spatiotempo-

ral reasoning about the changing locations of objects, such as predicting when and where objects can be found.

2.6 Conclusion

We have discussed how conceptions of robotic action planning can benefit from insights into human action planning. Indeed, we believe that constructing truly flexible and autonomous robots requires inspiration from human cognition. We focused on four basic principles that characterize human action planning, and we have argued that taking these principles on board will help to make artificial cognition more human-like.

First, we have discussed evidence that human action planning emerges from the integration of a rather abstract, perhaps symbolic representational level and concurrent planning at a lower, more concrete representational level. It is certainly true that multi-level planning can create difficult coordination problems. Using grounded cognition approaches in robotics is potentially a good method to ground such higher-level symbolic representation in lower-level sensorimotor representations, which may allow robot action to become more flexible and efficient.

Second, we have argued that human action planning emerges from the interplay of feedforward and feedback mechanisms. Again, purely feedforward or purely feedback architectures are likely to be more transparent and easier to control. However, fast, real-time robotic action in uncertain environments will require a hybrid approach that distributes labor much like the human brain does by combining slow and highly optimized feedforward control with fast sensorimotor loops that continuously update the available environmental information. A major challenge for the near future will be to combine such hybrid systems with error-monitoring and error-correcting mechanisms. When preparing pancake dough, accidentally pouring some milk outside the bowl would need to trigger a fast correction mechanism informed by low-level sensory feedback but not necessarily the re-planning of (or crying over) the

entire action. However, if for some reason the entire milk carton is emptied by this accident, leaving the agent without the necessary ingredient, feedback would have to propagate to higher, more abstract or more comprehensive planning levels to decide whether the plan needs to be aborted. How this works in detail and how decisions are made as to which level is to be informed is not well understood, but progress is being made. Research into feedback processes has yielded information about the optimal speed of sensorimotor loops [73], and we find it reasonable to expect that models using such fast feedback loops combined with accurate feedforward planning can ultimately produce human-like motor performance in robots.

Third, we have argued that while descriptions of human actions may refer to a hierarchy, it is not yet clear whether the cognitive—*in vivo* or *in silico*—representations of such actions need to be explicitly hierarchical as well. Equally unclear is whether representations that differ in hierarchical level would necessarily need to differ in format. However, it is clear that representations that are considered to be "higher in hierarchy" are more comprehensive. The concept of "making a pancake," say, is necessarily richer and more abstract than the associated lower-level actions of "reaching for egg" and "grabbing a pan," suggesting that the latter two are more directly grounded in sensorimotor activity [82]. Future research will need to investigate how representations at different planning levels (or different levels of description) interact or relate to each other.

The nature of goals and their role in action control is also a matter of ongoing research. The two different viewpoints—i.e. that goals require explicit representation or not—seem to reflect different preferences in conceptualization and modeling techniques, and it may well turn out that an explicit representation of goals in the preferred modeling language translates to a more implicit representation of goals in the actual functional or neural architecture. In robotics, most modern plan languages use a form of explicit goal-related action control that defines a goal as a required world state on which constraints can be imposed. Such a structure is flexible enough to allow equifinality, but it is unclear how

knowledge about the various means to produce a result is acquired. Ultimately, we believe that subsymbolic programming approaches may allow for more adaptive, human-like representational architectures—though likely more difficult to engineer and define provably safe operating conditions for.

To conclude, we believe that the construction of robots that are up to real-life, everyday actions in environments that are as uncertain as human environments requires the consideration of cognitive principles like the four principles we have discussed in this article. The benefit of doing so will be twofold. For one, it will strongly increase the flexibility of robots. For another, it will make robots more human-like in the eyes of the human user, which will help us understand and cooperate with our future robotic colleagues.

CHAPTER 3

Robotics and human action

The field of robotics is shifting from building industrial robots that can perform repetitive tasks accurately and predictably in constrained settings, to more autonomous robots that should be able to perform a wider range of tasks, including everyday household activities. To build systems that can handle the uncertainty of the real world, it is important for roboticists to look at how humans are able to perform in such a wide range of situations and contexts—a domain that is traditionally the purview of cognitive psychology. Cognitive scientists have been rather successful in bringing computational systems closer to human performance. Examples include image and speech recognition and general knowledge representation using parallel distributed processing (e.g. modern deep learning models).

Similarly, cognitive psychologists can use robotics to complement their research. Robotic implementations of cognitive systems can act as a "computational proving ground", allowing accurate and repeatable real-world testing of model predictions. All too often, theoretical predict-

This chapter is an adaptation of the book chapter *de Kleijn, R., Kachergis, G., & Hommel, B. (2015). Robotic action control: On the crossroads of cognitive psychology and robotics. In H. Samani (Ed.),* Cognitive robotics. *Taylor & Francis.*

ions—and even carefully conducted model simulations—do not scale up or even correspond well to the complexity of the real world. Psychology should always seek to push theory out of the nest of the laboratory and see if it can take flight. Finally, cognitive psychologists have an opportunity to conduct experiments that will both inform roboticists as they seek to make more capable cognitive robots, and increase our knowledge of how humans perform adaptively in a complex, dynamic world. In this chapter, we will give a broad but brief overview of the fields of cognitive psychology and robotics, with an eye to how they have come together to inform us about how (artificial and natural) actions are controlled.

3.2 Early history of the fields

3.2.1 History of cognitive psychology

Before cognitive psychology and robotics blended into the approach now known as cognitive robotics, both fields already had a rich history. Cognitive psychology as we now know it has had a rocky past (as have most psychological disciplines, for that matter). Breaking away from philosophy, after briefly attempting to use introspection to observe the workings of the mind, the field of psychology found it more reliable to rely on empirical evidence.

Although making rapid strides using this empirical evidence, for example in the form of Donders' now classic reaction time experiments which proposed stages of processing extending from perception to action, early cognitive psychology came to be dominated by a particular approach, *behaviorism*. This position, popularized by Watson [169] and pushed further by Skinner [143], held that the path for psychology to establish itself as a natural science on par with physics and chemistry would be to restrict itself to observable entities such as stimuli and responses. In this sense, behaviorists such as Skinner were strongly antirepresentational, i.e. against the assumption of internal knowledge and states in the explanation of behavioral observations. On the other hand, the focus on observable data brought further rigor into the field, and many interest-

ing effects were described and explained.

The behaviorist approach dominated the field of psychology during the first half of the 20th century. In the 1950s, seeming limitations of behaviorism fueled what some scholars would call the *neocognitive revolution*. Starting with Chomsky's scathing 1959 review of Skinner's book [27] that tried to explain how infants learn language by simple association, many researchers were convinced that behaviorism could not explain fundamental cognitive processes such as learning (especially language) and memory. The foundations of the field of artificial intelligence were also nascent, and pursuing explanations of high-level, uniquely human aptitudes—e.g. analytical thought, reasoning, logic, strategic decision-making—grew in popularity.

3.2.2 The computer analogy

Another factor contributing to the neocognitive revolution was the emergence of a new way to describe human cognition as similar to electronic computer systems. The basic mechanism operating computers was (and still is, in a fundamental way) gathering input, processing it, and outputting the processed information, not unlike the basic cognitive model of stimulus detection, storage and transformation of stimuli, and response production.

Clearly, this processing of information requires some representational states which are unaccounted for (and dismissed as unnecessary) by behaviorists. This new way to look at human cognition as an information processing system not only excited psychologists as a way of understanding the brain, but the analogy also raised hopes for building intelligent machines. The idea was that if computer systems could use the same rules and mechanisms as the human brain, they could also *act* like humans. Perhaps the most well-known proponent of this optimistic vision was Turing [161], who suggested that it wouldn't be long before machine communication would be indistinguishable from human communication. Maybe the secret of cognition lies in the way the brain gathers,

stores, and subsequently manipulates data, it was thought.

Alas, the optimists would be disappointed. It soon became clear that computers and humans have very different strengths and weaknesses. Computers can calculate half a million decimals of π within a second. Humans can read terrible handwriting. Clearly, humans are not so comparable to basic input–output systems after all. It would take another 25 years for cognitive psychology and artificial intelligence to begin their romance once again, in the form of the *parallel distributed processing* (PDP) approach [130].

3.2.3 Early cognitive robots

With this idea of smart computer systems in mind, it seemed almost straightforward to add embodiment to build intelligent agents. The first cognitive robots were quite simple machines. The *Machina Speculatrix* [166] consisted of a mobile platform, two sensors, actuators and "nerve cells". Understandably, these robots were designed to mimic behavior of simple animals, and could move safely around a room and recharge themselves using relatively simple approach and avoidance rules. Due to their simplicity, it was questionable exactly how *cognitive* these robots were—they are more related to cybernetics and control theory (e.g. [8])—but soon enough complexity made its way into cognitive robotics.

From the 1960s, robots would be able to represent knowledge and plan sequences of operations using algorithms such as STRIPS [47], that would now be considered essential knowledge for every AI student. The STRIPS planner, which represents goal states and preconditions and attempts to derive the action sequences that would achieve them before carrying them out, is quite slow to execute. Moreover, this type of planning suffers from its closed world assumption (i.e. that the environment and all relevant states are known—by programming—and will not change), and the massive complexity of the real world, leading to intractable computations. Yet the general approach taken by STRIPS—of modeling the environment, possible actions and state transformations, and goal states

via predicate logic, and operating robots via a sense-plan-act loop—has dominated cognitive robotics for quite some time, and is still a strong thread today.

Various behavior-based robotics architectures and algorithms—taking some inspiration from biological organisms—have been developed in the past few decades. An early, influential example is Rodney Brooks' *subsumption architecture* [21], which eschews planning entirely; "planning is just a way of avoiding to figure out what to do next", using a defined library of basic behaviors arranged hierarchically to generate behavior based on incoming stimuli. Although fast and often generating surprisingly complex behavior from simple rules (see also [20]), the subsumption architecture and many other behavior-based robotics algorithms do not yet incorporate much from the lessons to be learned from psychological studies in humans.

3.3 Action control

3.3.1 Introduction

One of the other areas that shows considerable overlap between robots and humans is motor or action control. Two types of control systems govern motor action: *feedforward* and *feedback* control systems.

A feedforward motor control system sends a signal from the (human or robotic) motor planning component to the relevant motor component using predetermined parameters, executing said action. Information from the environment can be considered only before execution begins, which makes feedforward control suitable for predictable environments. In contrast, a feedback motor control system incorporates information from itself or the environment (feedback) more or less continuously to modulate the control signal. In this way, the system can dynamically alter its behavior in response to a changing environment.

3.3.2 Feedforward and feedback control in humans

For many years, psychology and related disciplines have approached action control from rather isolated perspectives. As the probably first systematic study on movement control by Woodworth [171] had provided strong evidence for the contribution of environmental information, many authors have tried to develop closed-loop models of action control that rely on a continuous feedback loop (e.g. [1]). At the same time, there was strong evidence from animal and lesion studies [81, 155] and from theoretical considerations [87] that various movements can be considered in the absence of sensorimotor feedback loops, which has motivated the development of feedforward models (e.g. [59]).

Schmidt [140] was one of the first who argued that human action control consists of both feedforward and feedback components. According to his reasoning, human agents prepare a movement schema that specifies the relevant attributes of the intended movement but leave open parameter slots that are specified by using online environmental information. In particular, feedforward mechanisms seem to determine the necessary action components offline and pre-load at least some of them before initiating the action [59], and to selectively tune attention to stimuli and stimulus dimensions that are relevant to the task [64]. Feedback processes, in turn, provide excellent accuracy—often at the cost of speed [141]. These strengths and weaknesses have motivated hybrid models claiming that feedforward mechanisms provide the skeleton of action plans which leave open slots for parameters provided by feedback processes. Neuroscientific evidence has provided strong support for such a hybrid control model, suggesting that offline action planning along a ventral cortical route is integrated with online sensorimotor specification along a dorsal route [53, 54, 64, 140].

A particularly good example of this kind of interaction is provided by the observations of Goodale et al. [55]. In a clever experiment, participants were asked to rest their hand on a platform and point to a visual target presented at a random location on an imaginary line in their right visual

field. The participants were not told that in half of the trials the target would change location during the first saccade. The authors found that participants would successfully point to the target on these trials without even being aware of the location change, and without additional delay. As feedforward programming is assumed to take time, a fast and online feedback mechanism of which participants are unaware has to be responsible for this finding.

On a higher level, interaction between feedforward and feedback systems must exist for goal-directed action to be carried out. Higher-level, goal-directed action planning, such as planning to make pancakes would be impossible to plan in a completely feedforward fashion: it would require all motor parameters to be specified *a priori*, and thus would require exact knowledge of the position and properties of all necessary equipment and ingredients, such as weight, friction coefficients, et cetera.

Instead, many of these parameters can be filled in online by using information from the environment. It is not necessary to know the exact weight of a pan, because you can determine that easily by picking it up: you increase the exerted force until the pan leaves the surface of the kitchen counter. This does not rule out a complementary role for feedforward parameter estimation: you likely also learn a distribution of probable pan weights (e.g. more than 50 g and less than 10 kg) from your experience of other pans—or even just similarly-sized objects.

Interaction between feedforward and feedback becomes even more apparent on a higher level when a planned action fails to be executed. When a necessary ingredient is missing, replanning (or cancellation) of a preprogrammed action sequence may be necessary: if there is no butter, can I use oil to grease up the pan? Somehow, this information gathered by feedback processes must be communicated to the higher level action planner.

3.3.3 Feedforward and feedback control in robots

The theorizing on action control in robotic systems must be considered rather ideological, sometimes driven by the specifics of particular robots or tasks considered and sometimes by broadly generalized antirepresentationalist attitudes. Many early robots only had a handful of sensors and responded in a fixed pattern of behavior given a particular set of stimuli. Some robots were even purely feedforward, performing the same action or action sequence, with no sensory input whatsoever [106]. Feedforward or simple reactive control architectures make for very brittle behavior: even complex, carefully-crafted sequences of actions and reactions will appear clumsy if the environment suddenly presents an even slightly novel situation.

More complex architectures have been proposed, often with some analogy to biology or human or animal behavior, giving birth to the field of *behavior-based robotics.* The *subsumption architecture* [21] was a response to the traditional GOFAI, and posited that complex behavior need not necessarily require a complex control system. Different behaviors are represented as layers that can be inhibited by other layers. For example, a simple robot could be provided with the behaviors *wandering, avoiding, pickup,* and *homing.* These behaviors are hierarchically structured, with each behavior inhibiting its preceding behavior [7].

This hierarchy of inhibition between behavior is (although somewhat more complex) also visible in humans. For example, if your pants are (accidentally) set on fire while doing the dishes, few people would finish the dishes before stopping, dropping, and rolling. In other words, some behaviors take precedence over others. An approach similar to the subsumption architecture has been proposed by Arkin [6]. The *motor schema* approach also uses different, parallel layers of behavior, but does not have the hierarchical coordination that the subsumption approach does. Instead, each behavior contributes to the robot's overall response.

On a higher level, as noted in the previous section, other problems arise. When a planned action fails to succeed, for example because a robot can't

find a pan to make pancakes in, replanning is necessary. The earliest AI planners such as GPS would simply backtrack to the previous choice point and try an alternative subaction. However, this does not guarantee the eventual successful completion of the action. Other planners, such as ABSTRIPS [134], use a hierarchy of representational levels. When it fails to complete a subaction, it could return to a more abstract level.

However, truly intelligent systems should be more flexible in handling such unforeseen events. If a robot cannot make me a pizza with ham, maybe it should make me one with bacon? Generalization and substitution remain an elusive ability for robots, although vector space models of semantics (e.g. BEAGLE [72]) offer a step in the right direction. Like neural networks, these models represent items (e.g. words) in a distributed fashion, using many-featured vectors with initially low similarity between random items. As the model learns—say, by reading documents—item representations are updated to make them more similar (on a continuous scale) to contextually similar items. These continually-updated representations can be used to extract semantic as well as syntagmatic (e.g. part-of-speech) relationships between items. Beyond text learning, vector space models may ultimately be used to learn generalizable representations for physical properties and manipulations of objects and environments.

3.3.4 Robotic action planning

It is understood that reaching movements in humans have an initial ballistic feedforward component, followed by a slower feedback-driven component that corrects for error in the initial movement. As people become more adept at reaching to targets at particular distances, a greater portion of their movement is devoted to the initial feedforward component and less time is spent in the feedback component, thus speeding response times. Understanding how this happens should enable roboticists to make more adaptive, human-like motor planning systems for robots.

In this line of research, Kachergis et al. [75] studied sequence learning using mouse movements. Inspired by earlier work of Nissen and Bullemer [107], subsequences of longer sequences were acquired by human participants during a learning phase. The participants seem to implicitly extract the subsequences from longer sequences by showing faster response times and context effects.

These findings cast doubt on a simple chaining theory of sequential action. Rosenbaum et al. [129] interpreted these findings as evidence that sensory feedback is not a necessary component for action sequencing, in keeping with the conclusion of Lashley [87]. They argued that "the state of the nervous system can predispose the actor to behave in particular ways in the future," (p. 526), or, there are action plans for some behaviors. And yet, studies on spontaneous speech repair (e.g. [103]) also show that people are very fast in fixing errors in early components of a word or sentence, much too fast to assume that action outcomes are evaluated only after entire sequences are completed. This means that action planning cannot be exclusively feedforward, as Lashley [87] seemed to suggest, but must include several layers of processing, with lower levels continuously checking whether the current action component proceeds as expected. In other words, action planning must be a temporally extended process in which abstract representations to some extent provide abstract goal descriptions, which must be integrated with lower-level subsymbolic representations controlling sensorimotor loops. The existence of subsymbolic sensorimotor representations would account for context and anticipation effects, as described above.

The main lesson for robotic motor planning is that purely symbolic planning may be too crude and context-insensitive to allow for smooth and efficient multi-component actions. Introducing multiple levels of action planning and action control may complicate the engineering considerably, but it is also likely to make robot action more flexible and robust—and less "robotic" to the eye of the user.

3.4 Acquisition of action control

3.4.1 Introduction

In order for humans or robots to be able to achieve their goals, it is necessary for them to know what effect an action would have on their environment. Or, reasoning back to the inverse model, what actions are required to produce a certain effect in the environment. Learning relevant action–effect bindings as an infant is a fundamental part of development and likely bootstraps later acquisition of general knowledge.

In humans, learned action–effects seem to be stored bidirectionally. Following Lotze [91] and Harless [57], James [69] noted that intentionally creating a desired effect requires knowledge about, and thus the previous acquisition of action–effect contingencies. The *Theory of Event Coding* (TEC) is a comprehensive empirically well-supported theoretical framework explaining the acquisition and use of such action–effect bindings for goal-directed action ([65], for recent reviews see [63, 142]). TEC states that actions and their expected effects share a common neural representation. Therefore, performing an action activates the expectation of relevant effects and thinking of (i.e. intending or anticipating) an action's effects activates motor neurons responsible for achieving those effects.

3.4.2 Human action–effect learning

In traditional cognitive psychology experiments, action–effect bindings are acquired by having humans repetitively perform an action (such as pressing a specific button on a keyboard), after which an effect (such as a sound or a visual stimulus) is presented. After a certain amount of exposure to this combination of action and effect, evidence suggests that a bidirectional binding has been formed. When primed with a previously learned effect, people respond faster with the associated action [42]. This action–effect learning is quite robust but sensitive to action–effect contingency and contiguity [43].

Of course, action–effect learning does not only happen in artificial en-

vironments such as psychology labs. In fact, action–effect learning in humans starts almost instantly after birth [164] and some would argue even before. Young infants perform uncoordinated movements known as *body* or *motor babbling*. Most of these movements will turn out to be useless. However, some of them will have an effect that provides the infant with positive feedback. For example, a baby could accidentally push down with its right arm while lying on its belly, resulting in rolling on its back and seeing all sorts of interesting things. Over time, the infant will build up action–effect associations for actions it deems useful, and can perform motor acts by imagining their intended effects.

Having mastered the intricacies of controlling the own body, higher level action–effects can be learned in a manner similar to motor babbling. Eenshuistra et al. [39] give the example of piloting a spacecraft that you are trying to slow down. If nobody ever instructed you on how to do that, your best option would probably be pressing random buttons until the desired effect is reached (be careful with that self-destruct button!). Once you have learned this action–effect binding, performance in a similar situation in the future will be much better.

3.4.3 Robotic action–effect learning

The possibility that cognition can be grounded in sensorimotor experience and represented by automatically created action–effect bindings has attracted some interest of cognitive roboticists already. For instance, Kraft et al. [82] have suggested a three-level cognitive architecture that relies on object-action complexes, that is, sensorimotor units on which higher-level cognition is based. Indeed, action–effect learning might provide the cognitive machinery to generate action-guiding predictions and the offline, feedforward component of action control. This component might specify the invariant aspects of an action, that is, those characteristics that need to be given for an action to reach its goal, to create its intended effect while an online component might provide fresh environmental information to specify the less goal-relevant parameters, such as the speed of a reaching movement when taking a sip of water from

a bottle [64]. Arguably, such a system would have the benefit of allowing for more interesting cognitive achievements than the purely online, feedback-driven systems that are motivated by the situated-cognition approach [22]. At the same time, it would be more flexible than systems that rely entirely on the use of internal forward models [36]. Thus, instead of programmers trying to imagine all possible scenarios and enumerate reasonable responses, it might be easier to create robots that can learn action–effect associations appropriate to their environment and combine them with online information.

In robots as well as in humans, knowledge about one's own body is required to acquire knowledge about the external world. Learning how to control your limbs—first separately and then jointly (e.g. walking)—clearly takes more than even the first few years of life: after learning to roll over, crawl, and then walk, we are still clumsy at running and sport for several years (if, indeed, we ever become very proficient). Motor babbling helps develop tactile perception and proprioception—as well as visual and even auditory cues—of what our body in motion feels like. Knowing these basic actions and their effects on ourselves (e.g. what hurts) lays the foundation for learning how our actions can affect our environments.

In perhaps the first ever study of motor babbling in a (virtual) robot, Kuperstein [84] showed how random movement execution can form associations between a perceived object-in-hand position and the corresponding arm posture. This association is bidirectional, and as such is in line with ideomotor (or TEC) theory. We (and others, e.g. [25]) believe that such bidirectional bindings can help robots overcome traditional problems, such as inverse model inference from a forward model.

More recent investigations in robotic motor babbling have extended and optimized the method to include behavior that we would consider *curiosity* in humans. For example, Saegusa et al. [135] robotically implemented a sensorimotor learning algorithm that organized learning in two phases: *exploration* and *learning*. In the exploration stage, random movements are produced, while in the learning stage the action–effect bindings (or,

more specifically, mapping functions) are optimized. The robot can then direct more effort to learning bindings that have not yet been learned well.

3.5 Directions for the future

3.5.1 What's next?

Many questions remain with respect to the acquisition and skillful performance of not only well-specified, simple actions (e.g. reaching to a target) but of complex actions consisting of various components and involving various effectors. Indeed, how can we create a learning algorithm that can go from basic motor babbling to both successful goal-directed reaching, grasping, and manipulation of objects? To accomplish this obviously difficult goal, it will likely be beneficial for psychologists to study infants' development of these abilities and beneficial for cognitive roboticists to learn more from human capabilities.

3.5.2 Affordance learning

Object manipulation and use is an indispensable activity for robots working in human environments. Perceiving object affordances—i.e. what a tool can do for you or how you can use an object—seems to be a quick, effortless judgment for humans, in many cases. For example, when walking around and seeing a door, you automatically pull the handle to open it.

One of the ways robots can perform object affordance learning is by motor babbling using simple objects as manipulators (e.g. [152]). In a so-called *behavioral babbling stage* a robot applies randomly chosen behaviors to a tool and observes their effects on an object in the environment. Over time, knowledge about the functionality of a tool is acquired, and can be used to manipulate a novel object with the tool.

As impressive as this may sound, this approach does not allow for easy generalization, and the robot cannot use this knowledge to manipulate

objects using another, similar, tool. More recent approaches, such as demonstrated by Jain and Inamura [68] infer functional features from objects to generalize affordances to unknown objects. These functional features are supposed to be object invariant within a tool category.

In humans, an approach that seems successful in explaining affordance inference is based on Biederman's *recognition-by-components theory* [15]. This theory allows for object recognition by segmenting an encountered object in elementary geometric parts called *geons*. These are simple geometric shapes such as cones, cylinders and blocks. By reducing objects to a combination of more elementary units invariance is increased, simplifying object classification. Biederman recognized 36 independent geons, having a (restricted) generative power of 154 million three-geon objects.

In addition to being useful for object classification, geons can also be used to infer affordances. For example, a spoon is suitable for scooping because its truncated hollow sphere at the end of its long cylinder allows for containing things, and an elongated cylinder attached to an object can be used to pick it up. One very promising example of the use of geons in affordance inference is demonstrated by Tenorth and Beetz [157]. This technique matches perceived objects to three-dimensional CAD models from a public database such as Google Warehouse. These models are then segmented into geons, which makes affordance inference possible.

However, the affordances that geons give us need to be learned in some way. Teaching robots how to infer what a tool can be capable of remains difficult. Ultimately, we want affordances to develop naturally during learning: be it from watching others, from verbal instruction, or from embodied experimentation. Task context is also an important aspect of affordance learning: depending on the situation, a hammer can be used as a lever, a paperweight, a missile, or well, a hammer. To understand how context affects action planning, studying naturalistic scenes and human activities jointly seems essential (cf. [2]).

Learning geon affordances that can be generalized to object affordances seems a fruitful approach to automating affordance learning in robots,

although it is early to say whether this or other recent approaches will fare better. For example, deep neural networks use their multiple hidden layers along with techniques to avoid overfitting to learn high-level perceptual features for discriminating objects. The representations learned by such networks are somewhat more biologically plausible than geon decompositions, and thus may be more suitable for generalization (although cf. [154] for recently discovered generalization problems with deep neural networks).

3.5.3 Everyday action planning

A major obstacle in the way of robots performing everyday actions is the translation of high-level, symbolic task descriptions into sensorimotor action plans. In order to make such translations, one method would be to learn the other way around: by observing sensorimotor actions, segment and classify the input.

Everyday action is characterized by sequential, hierarchical action subsequences. Coffee and tea-making tasks, for example, have shared subsequences such as adding milk or sugar. Moreover, the goal of adding sugar might be accomplished in one of several ways: e.g. tearing open and adding from a packet, or spooning from a bowl or box. Also, these subsequences do not necessarily have to be performed in the same order every time (with some constraints, of course). It is this flexibility and ability to improvise that makes everyday action so natural for humans, yet so hard for robots.

Cognitive models that represent hierarchical information have been proposed (e.g. [18, 33]), but differ in the way they represent these hierarchies. One approach explicitly represents action hierarchies by hard-coding them into the model—hardly something we can do for a general autonomous robot—whereas the latter models hierarchy as an emergent property of the recurrent neural network. More recently, the model put forth by Kachergis et al. [76], uses a recurrent neural network with biologically plausible learning rules to extract hierarchies from observed

sequences, needing far fewer exemplars than previous models.

3.6 Conclusion

In this chapter, we have discussed several concepts that are shared between cognitive robotics and cognitive psychology in order to argue that the creation of flexible, truly autonomous robots depends on the implementation of algorithms that are designed to mimic human learning and planning. Thus, there are many relevant lessons from cognitive psychology for aspiring cognitive roboticists.

Ideomotor theory and its implementations such as TEC provide elegant solutions to action–effect learning. Robotic motor learning algorithms that use motor babbling to bootstrap higher-order learning seem to be promising, and require little *a priori* knowledge given by the programmer, ultimately leading to more flexible robots.

Generalization of action plans is still a very difficult problem. Inferring hierarchical structure of observed or learned action sequences seems to be a promising approach, although the structure of everyday action appears to be nearly as nuanced and intricate to untangle as the structure of human natural language—and less well-studied, at this point. Again, we believe that biologically inspired learning models such as LeabraTI can play a role in making robotic action more human-like.

The overlapping interests of cognitive robotics and cognitive psychology have proven fruitful so far. Mechanisms like motor babbling and affordance inference, which are extensively studied in humans, can provide robots with techniques to make their behavior more flexible and human-like. We believe human inspiration for robots can be found at an even lower level by incorporating biologically-inspired neural models for learning in robots.

Part B

Empirical and model observations

CHAPTER 4

Predictive movements and human reinforcement learning of sequential action

Most daily human behaviors can be seen as learned sequential actions: from walking, cooking, and cleaning to speaking and writing. Consequently, sequence learning has been studied in different contexts ranging from implicit sequence learning [19, 29, 107, 149] to language acquisition [41, 136], typing [46, 52], and manual everyday actions [18, 32]. In implicit learning research, an important paradigm has been the *serial reaction time* (SRT) task, which requires participants to press one of four buttons when cued by a corresponding light, in a sequence that repeats—unbeknownst to learners—every 10 presses [107]. Subjects trained on this repeating sequence developed faster reaction times (RTs) over the course of training, as compared to a control group responding to a random sequence of stimuli. The SRT paradigm has been cited as evidence for implicit learning, as subjects experiencing the re-

This chapter is an adaptation of the article *de Kleijn, R., Kachergis, G., & Hommel, B. (under revision). Predictive movements and human reinforcement learning of sequential action.*

peating sequence, despite showing faster RTs over time, report no explicit knowledge of the sequence when debriefed afterwards. However, performance does suffer somewhat when participants must simultaneously perform a second task [107], suggesting that learning in the SRT task does require some attentional resources or effort. The role of attention in the SRT task was further studied by Fu et al. [50], who demonstrated that reward motivation can improve the development of awareness of the sequence. They reasoned that reward motivation regulates the amount of attention paid towards the stimuli, which in turn facilitates sequence learning. Additionally, Willingham et al. [170] found that some participants achieved a degree of declarative knowledge after a fixed training period in the SRT task, and that additional training resulted in more explicit knowledge for many subjects, if not all. On balance, it seems that the SRT task is neither wholly implicit nor wholly explicit.

The dissociation of implicit and explicit processes facilitating sequence learning remains a topic of debate, yet learning remains robust under high degrees of noise and complex structure in the sequences [29]. Complex action sequences are not mere stimulus–response chains, but rather require representing sequential context in order to learn [87]. Moreover, human behavior is often thought of as *predictive*—indeed, many models of sequential learning operate on a prediction-based error signal [18, 76]. As such, it is problematic that the discrete button presses in the SRT paradigm cannot distinguish an *anticipatory* response due to correctly predicting the stimulus (or a slow response due to an incorrect prediction) from *reactive* (though perhaps pre-potentiated) responses based on the cue. Truly predictive responses—that is, those made in the interstimulus interval before the next response is cued—are not valid responses in the SRT paradigm.

In this paper we introduce two modifications of the SRT paradigm that allow us to naturally investigate both predictive and reactive responding in human sequence learning. In Experiment 1, recognizing that actions are continuous movements that can reveal the underlying dynamics of

the cognitive processes driving them [147], we used a mouse-tracking adaptation of the SRT task in which spatial locations are both stimuli and response options [74, 75]. By tracking their movement before and after the next target is cued, we investigated changes in predictive versus cued responding over the course of the experiment [160]. Using this trajectory SRT paradigm, we replicated the overall Nissen and Bullemer [107] RT results, and moreover show sequential context effects—predictive bends in response trajectories—along with different movement dynamics pre- and post-cue.

In many implicit learning tasks such as artificial language learning and the SRT paradigm, learning is dependent on recognizing some statistically reliable sequential structure in stimuli not under the learner's control. However, everyday human action learning is often not characterized by processing a steady stream of stimuli, but by exploring the environment (i.e. choosing actions) and receiving positive and negative feedback. Prediction is thus an essential element of reinforcement learning (RL), which is a well-established paradigm in the field of machine learning [153] that was originally motivated by much earlier behaviorist stimulus–response learning studies [144]. RL paradigms allow learning agents to interact with a task solely through observations, actions, and rewards. The rewards validate the actions, without the need for explicit cueing or other forms of instruction. Thus, learning is exploratory, and accomplished via trial-and-error. In Experiment 2, we further modified the trajectory SRT paradigm by not cueing responses at all: participants had to explore response alternatives until the correct one was found, receiving feedback (negative or positive points) at each response. We investigated sequence learning in this RL SRT paradigm that required prediction rather than reaction, and found correspondences between successful learners in this paradigm and in the reactive SRT paradigm in Experiment 1. Using the RL paradigm allowed us to study the effect of rewards on sequence acquisition in more detail, yielding not only response times but also errors over time. Thus, the current study adapted the trajectory SRT task to allow for free movement and limited instruction, allowing

learners to explore and learn from trial-and-error.

In addition, we attempted to capture human performance and error patterns using reinforcement learning models. Due to the relatively simple nature of the task, we investigated if simple (i.e. model-free) RL models were sufficient to learn the repeating sequence by trial-and-error. We assessed the RL data both in terms of earlier SRT data and in comparison to three standard RL models. Overall, this study provides insights into prediction error-driven learning of sequential action learning.

4.2 Experiment 1

The purpose of the first experiment was to replicate earlier findings by Nissen and Bullemer [107] using the trajectory SRT paradigm. This study used four stimuli in a recurring sequence of length 10, horizontally displayed on a screen. Designating the stimulus positions from left to right as numbers, the original sequence read 4–2–3–1–3–2–4–3–2–1. To fit the trajectory paradigm the sequence was mapped to a square, left-to-right and top-to-bottom (i.e. 1 = top left, 2 = top right, 3 = bottom left, and 4 = bottom right). Participants moved the mouse from one stimulus position to the next, corresponding to the sequence. We tested two groups of participants, one trained on the recurring sequence and the other trained on a random sequence. After ten blocks of training participants completed a generating task. This task consisted of the same basic test conditions, except participants were asked to predict the sequence instead of following it.

Nissen and Bullemer [107] originally found that participants showed improved performance within the first block of training. Performance suffered under dual-task conditions and varied as a function of serial position in a pattern suggesting that learners were chunking the sequence into two pieces. In total, the study's results suggest that attention to the sequence is crucial for both implicit and explicit sequence learning, but that improved performance is not critically dependent on awareness of the sequence. For the purpose of Experiment 1 only the initial experi-

ment was replicated. We expected to replicate the basic improvement of performance, as well as the chunking pattern that was observed. Like Willingham et al. [170], we included a final generation task, in which participants were asked to reproduce any action sequence they felt they had learned during training.

4.2.1 Methods

Participants

Participants in this experiment were 22 Leiden University undergraduate students who participated in exchange for 3.50 euros or course credit.

Apparatus and materials

The experiment was performed on a computer with a 21-inch monitor with 60 Hz refresh rate and a resolution of 1024x768 pixels. Participants used a mouse to move the cursor. The experiment was programmed in Python with the PyGame library, and cursor position was sampled at every screen refresh.

Procedure

Participants were alternately assigned to one of the two between-subjects conditions according to the order they signed up. In the NB87 sequence condition, participants were given a repeating sequence of 10 locations corresponding to the Nissen and Bullemer [107] sequence (4–2–3–1–3–2–4–3–2–1). In the random sequence condition, participants followed a randomly generated movement sequence without repetitions (i.e. staying at the same location).

Participants were told to quickly and accurately move the mouse cursor to whichever square turned green. After arriving at the highlighted stimulus, another stimulus was highlighted after a 500 ms ISI. Participants completed 80 training trials, each of which contained a series of 10 locations. Participants were given a rest break every 20 training trials. Fol-

lowing the training phase, participants were asked to try to reproduce any sequence they had learned.

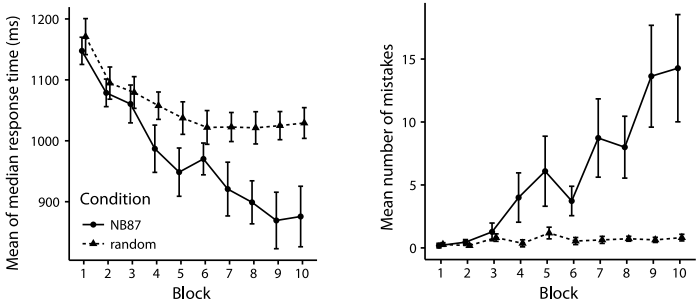
Each block contained a series of 80 location stimuli (i.e. 10 repetitions of the NB87 sequence) which participants had to track with the cursor. The stimulus display consisted of four red squares (location 1 = upper left, 2 = upper right, 3 = lower left, 4 = lower right), displayed continuously. Each stimulus was an 80 × 80 pixel square, separated by 440 pixels of white space. As a participant's cursor arrived at the green square, the square's color would change to red, like the other stimuli. The next target stimulus in the sequence would change color after a 500 ms ISI.

After training, participants were given a generating task similar to the training task. In the generating task, participants were asked to predict where they thought the stimulus would appear and move the mouse to that square. In other words, they were asked to complete the sequence without being cued. A correct prediction would cause no color change while an error would cause the correct continuation of the sequence to appear in green, and participants were to move to the next location.

4.2.2 Results

Response times

Data were analyzed from the 22 participants (11 per condition) that completed the experiment. Median movement time to a target was 1,040 ms (*SD*: 1,776). Of 17,578 target arrival times, 84 were removed for being slower than 2,816 ms (median + *SD*). Each subject's median RT for correct movements on each block was computed. Figure 4.1a shows the mean of median RTs by block for the two conditions. Participants in both conditions got faster over the course of the experiment, but participants in the NB87 sequence condition improved more than those in the random condition, replicating the Nissen and Bullemer [107] speedup. There was a 25% reduction in reaction time over the course of training. These data were analyzed by a two-way analysis of variance, which indicated significant main effects of condition ($F(1, 20) = 31.3$, $p < .001$) and block ($F(7,$



(a) Mean of median RTs by block show that both conditions sped up over the course of Experiment 1, but that NB87 improved more.

(b) Mean number of errors by block shows only the NB87 participants made an increasing number of errors.

Figure 4.1 | Experiment 1 RTs and error rates by block. Error bars show +/–1 SE.

168) = 6.3, $p < .05$), and a significant interaction effect ($F(7, 210) = 14.7$, $p < .01$) between the two.

The accuracy data is shown in Figure 4.1b. Accuracy was high across training blocks although it dropped over time in the NB87 group, particularly after the first three blocks of training. A two-way analysis of variance confirmed a significant main effect of group ($F(1, 20) = 36.7$, $p < .001$) and a significant interaction effect ($F(9, 210) = 14.1$, $p < .001$). These results are evidence of sequence learning, replicating the Nissen and Bullemer [107] keypress-based results. However, there was a speed-accuracy tradeoff in the NB87 condition: both accuracy and RT dropped over time. This was not present in the Nissen and Bullemer [107] results, but can be explained through the difference in response execution. Key-presses are intermittent and can only be made in response to a stimulus (prestimulus responses were not recorded), while mouse movements are continuous and made constantly. Indeed, in the NB87 condition faster median hit RTs on a training block had a significant negative correlation with the number of errors in that block (for the 67 of 110 blocks containing errors; $r = -.56$, $t(65) = -5.48$, $p < .001$), showing a speed-accuracy

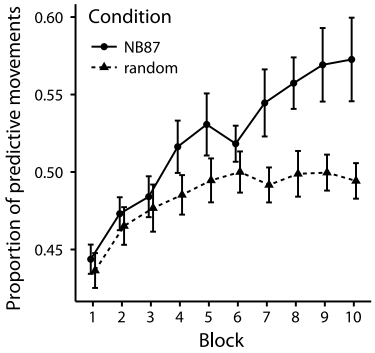


Figure 4.2 | Proportion of predictive movements (i.e. movements made during the ITI) by block in each condition. Random condition participants re-center, whereas NB87 participants move towards other stimuli. By block 4, NB87 participants were making more than half of their movement predictively, and continued to move more predictively: up to 57% by the end of the experiment. Error bars show +/–1 SE.

tradeoff. This is likely due to the trajectory SRT paradigm encouraging prediction, allowing participants to move freely while performing the experiment.

Indeed, an analysis of the proportion of distance traveled before arriving at the next target during the 500 ms interval before the cue appeared (i.e. predictive movement), shown in Figure 4.2, shows that participants in the random condition level off at making half of their movement, on average, during the pre-cue interval, whereas by block 10, participants in the NB87 condition predictively completed over 57% of their movement in the 500 ms interval before the next location is highlighted. This shows that participants in the NB87 are predicting the next target location and already moving towards–getting over halfway there–before the next cue appears.

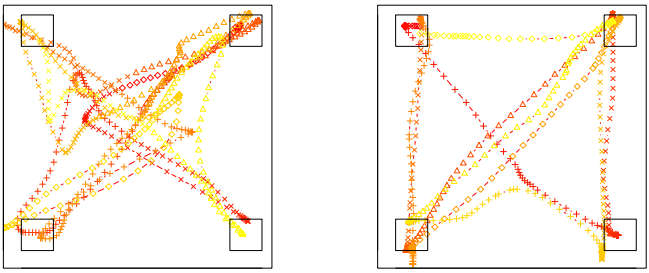
A two-way ANOVA with block as between-subject and serial position as within-subject factors showed significant main effects for block ($F(9, 210) = 32.3$, $p < .001$ and serial position ($F(9, 100) = 10.2$, $p < .01$). To de-

termine whether participants became faster at the entire sequence or rather learned some chunks better than others, mean RT was calculated for each serial position. Similar to the Nissen and Bullemer [107] results, RTs on the second, fifth and eighth serial positions are slow, which may indicate that participants chunk the full sequence into two small, well-learned pieces.

Performance on the generating task was poor, as participants on average did not manage to reproduce the sequence without making many errors ($M = 5.77$ errors). This indicates that, although training performance showed evidence of sequence learning, participants were not explicitly aware of the sequence. It is possible that participants would eventually be able to reproduce the sequence if training were extended, as in Willingham et al. [170]. Nissen and Bullemer [107] originally found that participants were able to score around 80% correct on the generating task after two blocks of ten trials. Although the current study only required participants to complete one block of ten trials during the generating task, participants did not show any improvement during the task.

Trajectory results

Figure 4.3 shows an example of mouse movements during a characteristic trial from each condition. Participants in the random condition (e.g. Figure 4.3a) tended to re-center the cursor after hitting a target, during the 500 ms ISI. This strategy is not unreasonable under conditions of uncertainty, as it minimizes the distance to potential targets, and the next target cannot be predicted in the random condition. Centering behavior is shown in Figure 4.4a. Centering behavior is defined as the proportion of time spent in the center 100×100 pixels of the screen between reaching the previous target and current target reached. We deemed the distinction between reactive and predictive movements (as made by Dale et al. [34]) unsuitable for the current analyses due to the random condition used to compare. As the experiment progressed, participants in the random condition adopted a centering strategy that minimized distance to potential targets, while participants in the predictable NB87 condition

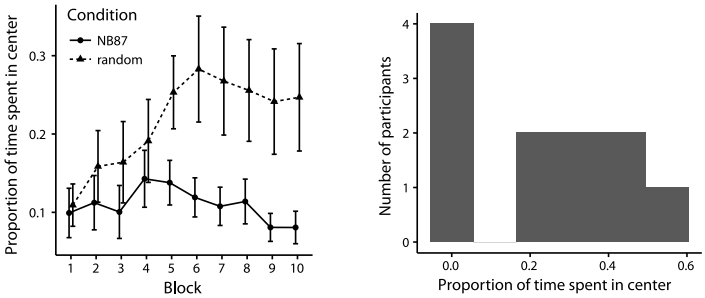


(a) A trial from the random condition, in which the next location was chosen at random, without repeats. All 11 random participants adopted a similar strategy of re-centering the cursor after each response. This is optimal in the sense that it was impossible to know which location will be highlighted next.

(b) A characteristic trial of a participant's movements during the NB87 sequence, beginning at location 4 (lower right) and ending at location 1 (upper left). These isomorphic trajectories can be compared for context effects. Only 4 NB87 participants showed centering movements in the last half of training.

Figure 4.3 | Characteristic movements in one trial from the random condition (a) and the NB87 condition (b). t_0 = red, t_{end} = yellow.

did not show this behavior. Participants in the random condition spent an increasingly larger proportion of time in the center of the screen compared to NB87 participants, $F(9, 180) = 2.51$, $p = .010$ for the interaction between block and condition. Similar centering behavior has been reported, but not quantified in the current context by Duran and Dale [38], and Dale et al. [34]. Interestingly, not all participants in the random condition displayed this centering strategy, as evidenced by the large standard errors, especially in the final half of the experiment. Instead, participants seemed to employ either a non-centering strategy or a centering strategy in which they spent almost 25% of the ISI in the center of the screen.



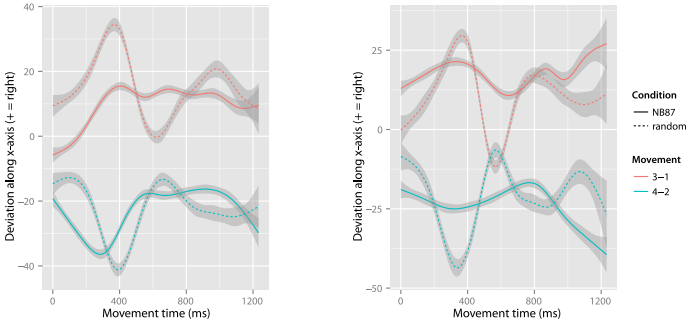
(a) Proportion of time spent in the center of the screen, defined as a 100×100 pixel square in the center of the screen. Centering behavior in the random condition is clearly visible. Error bars show +/–1 SE.

(b) Distribution of centering behavior for the last half of the experiment for the random condition. Two groups of participants can be identified: those who center during the ISI and those who do not.

Figure 4.4 | Centering behavior during the ISI.

With learning, targets are predictable in the NB87 sequence condition, thus participants are expected to show faster reaction times (RTs) as training proceeds.

The NB87 sequence, 4–2–3–1–3–2–4–3–2–1, contains only one identical transition (3–2, a diagonal movement), although other movements are isomorphic (e.g. 4–2 and 3–1). We examined the development of sequential context effects–deflections in response trajectory caused by the prior or subsequent location–by plotting the average trajectories for the isomorphic movements: 4–2 vs. 3–1. In the experiment, these movements are vertical, and we were interested in investigating the average deflections from the direct path from one stimulus center to another. We averaged position across subjects for these movements and plotted their deviation from the direct path (y-axis) over time (x-axis) in Figure 4.5, split by condition, and for each half of training. Early in training, some centering behavior is apparent in both conditions, most notably in the 4–2 movement. This movement also clearly shows the absence of center-



(a) Horizontal deviation during movement (i.e. over time) in early training. Both conditions' trajectories show some centering behavior, bending towards the middle (i.e. up for 3–1, down for 4–2). NB87 trajectories show less deviation.

(b) Horizontal deviation during movement in late training. The random condition shows more centering behavior, while the NB87 trajectories show little variation except at the end of the movements when they diverge, showing prediction of the subsequent stimulus.

Figure 4.5 | Averaged trajectories for vertical movements 4–2 and 3–1.

ing behavior late in training for the NB87 condition. The 4–2 movement also shows participants tended to move towards the left after completing the movement. As the next target in the sequence is 3, which is situated to the bottom left of the current target, this indicates they were beginning to move towards the subsequent target. These trajectory analyses corroborate that NB87 participants were making increasingly predictive movements, bending towards the next stimulus position based on their contextual knowledge.

4.2.3 Discussion

In summary, Experiment 1 replicated the results from the Nissen and Bullemer [107] serial button-pressing task with a mouse-trajectory version of the task, showing that participants learn regularities in the stim-

ulus stream and exhibit speeded responding, even though they are bad at explicitly reproducing the sequence. We have also demonstrated the advantage of the trajectory-tracking SRT task: because participants can move the mouse cursor during the interstimulus interval—before the next cue has appeared—we can distinguish predictive movements (towards the correct next stimulus) from post-cue speed-ups. Indeed, we found that participants in the NB87 sequence condition made an increasingly large proportion of their movement during the 500 ms pre-cue interval. Also, we found centering behavior similar to Dale et al. [34]. However, in addition to their findings we compared centering behavior between the random and NB87 condition, showing that participants in the random condition show significantly more centering behavior, which can be explained by uncertainty in prediction. Having established that prediction plays a role in the speed-up seen in the SRT-trajectory paradigm, in Experiment 2 we made prediction the essential goal of the task, requiring learners to move to the next location without a cue, and only giving feedback upon making a response.

4.3 Experiment 2

The results of Experiment 1 show that spatial sequences can be learned through cued learning, replicating a huge body of literature on the SRT task introduced by Nissen and Bullemer [107]. However, sequence learning in everyday action can hardly be considered cued. Instead, humans are in constant interaction with their environment, exploring it and receiving positive or negative feedback on their taken actions. In Experiment 2, we adapted the paradigm of the trajectory SRT into an exploration paradigm in which participants actively try out the alternative options and receive feedback (reinforcement or punishment). More specifically, the goal of Experiment 2 was to examine reinforcement learning within the trajectory SRT paradigm, and to compare human performance to basic baseline models. The trajectory SRT task was adapted to no longer cue participants with the next target position, forcing them to instead explore the response alternatives until the correct one was found.

Moving the mouse cursor from the previous target to another response alternative resulted in a reward (+1) or penalty (–1) that was accumulated throughout the experiment and displayed continuously. Upon reaching a valid target, it would change color to green, add to the score by 1, and allow the participant to continue exploring. Reaching for an invalid target caused it to change to red, subtract from the score by 1, while the cursor was relocated to the previously occupied target, effectively resetting the participant's progress. Target validity was determined by a recurrent sequence, taken from the Nissen and Bullemer [107] study, and adapted to fit the trajectory SRT paradigm. Designating the stimuli as numbers from left to right, top to bottom, the sequence read 4–2–3–1–3–2–4–3–2–1.

4.3.1 Methods

Participants

Participants in this experiment were 13 Leiden University students and employees (aged $M = 23.9$, $SD = 6.4$) who participated in exchange for 3.50 euros or for course credit.

Procedure

Participants were instructed that they would be presented with four target squares in the corners of the screen which they were to explore by moving the mouse, each time resulting in either a gain or loss of one point. Participants were told to try to maximize their score, which was displayed continuously at the top of the screen. Unbeknownst to the participants, only one of the four targets would be valid at any given moment, but all were colored blue, so the target could not be visually distinguished. Upon reaching a valid target, its color would change to green momentarily and the score would increase by one. The participant would then be able to continue exploring for the next target. Arriving at an invalid target caused it to change to red momentarily and the score was decreased by one, while the cursor was relocated to the previously

occupied target. Thus, although there were no instructions explicitly indicating it, participants likely inferred that they had chosen the incorrect stimulus, and should choose one of the remaining two—if they also assumed the same target was never repeated immediately, which was true. In the absence of a previous target (i.e. at the beginning of the experiment or after a rest break) the cursor was moved back to the middle of the screen.

Unbeknownst to the participants, each trial consisted of a series of 10 targets (labeled 1–4 left-to-right and top-to-bottom: 4–2–3–1–3–2–4–3–2–1) that repeated continuously, with no indication where one trial stopped and the next began. Participants completed eight blocks of 10 such trials, with a short rest break after every two blocks (i.e. 200 correct movements). A participant who somehow knew the sequence before entering the experiment and never made an error would therefore make 800 movements to valid targets, receiving the theoretical maximum of 800 points. At worst, a participant with no memory of even the previous target they had tried may make an infinite number of errors, and may never finish the experiment. Assuming enough memory to not repeat the same invalid target more than once when seeking each target (i.e. an elimination strategy), a participant using this elimination strategy would expect on average to score 0 points, as the expected value (EV) of completing one movement successfully is 0.[1] Note that participants were not told that there was a single deterministic sequence, let alone details such as how long the sequence was.

4.3.2 Results

The data from all 13 participants were analyzed. The distribution was bimodal, with four participants collecting less than 300 points and all but one of the rest accumulating more than 500 points each. Given the bimodal score distribution, a median split was used to divide the participants into high-performing ($\geq$ 526; 7 people) and low-performing ($<$

[1] 33% of chance success in one try (+1), 33% chance of success in two tries (–1+1), and 33% chance of success in three tries (–1–1+1).

526; 6 people) groups. In the high-scoring group, participants achieved almost flawless performance after only approximately 30 trials, with a final mean score of 652 (max: 725), while the low-scoring group only gradually increased their score (final mean score: 287). The remaining analyses were carried out for each group in an attempt to understand the great variability in performance—and the impressive success of the high-scoring group.

Response times

The overall median response time (RT) for all stimulus arrivals was 1,401 ms (*SD* = 4,980). Of 10,400 correct target arrival times (median = 1,078 ms, *SD* = 2,216), 317 (3%) were trimmed for being too slow (median + 2 · *SD*). Of the 4,117 incorrect stimulus arrival times (median = 2,397 ms, *SD* = 8,401), 100 were trimmed for being too slow (2.4%). Each subject's median RT for correct and incorrect movements was computed for each 10-trial block. Figure 4.6 shows the mean of subjects' median correct and incorrect RTs over the experiment, split into high- and low-performing group. RTs for correct movements improve in both groups during the first few blocks, but the high-scoring group speeds up more than the low-scoring group. Figure 4.6 also shows that the rare incorrect RTs for the high-performing group get slower over the course of the experiment, whereas the low-performing group's incorrect RTs only increase a bit. The strikingly slow errors of high-performing participants, compared to errors that are barely slower than correct movements for the low performers may indicate a different mode of behavior. A possible explanation is that low performers are simply not trying to learn a sequence, or do not expect it to to be deterministic, whereas high performers explicitly learn the sequence, and when they are uncertain they must pause to try to recall the next target.

Accuracy

The mean number of errors made over the entire experiment was 19.8 (*SD* = 21.3) for the high-scoring group, and 63.5 (*SD* = 11.9) for the low-

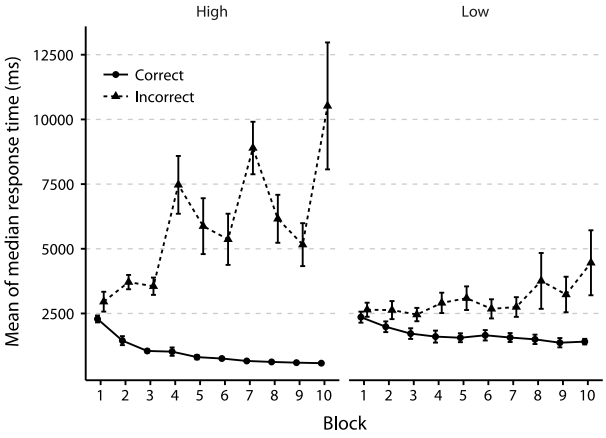
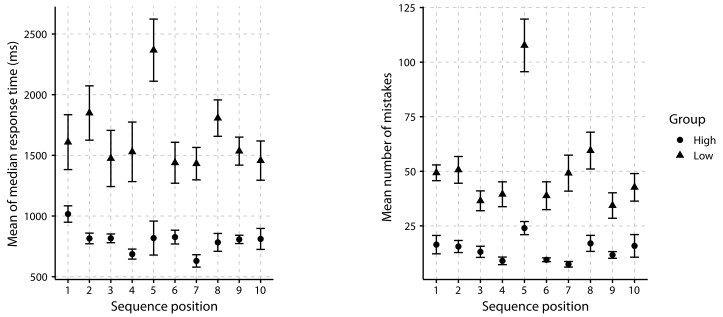


Figure 4.6 | The mean of subjects' median correct RTs by block shows that high-performers' (left panel) RTs improved more than the low-performers' (right panel) RTs over training. The mean of subjects' median incorrect RTs by block shows that the high-performing group's incorrect RTs actually increased, whereas the low-performing group's stayed roughly the same across the experiment. Error bars show +/−1 SE.

scoring group. Over time, the number of errors decreased especially for the high scoring group. Examining the errors made by each group of participants according to where they were in the sequence revealed that for both groups the fifth stimulus was particularly challenging. This is reflected in the mean number of errors for each group (see Figure 4.7b, as well as in the mean RT to the target by sequence position (see Figure 4.7a).

Comparison to Experiment 1

The pattern we observe in the accuracy and response time data bears some resemblance to the pattern observed in Experiment 1, despite the use of cues in that experiment. Although the RL SRT task in Experiment 2 was fundamentally different from the cued SRT task in Experiment 2, the same sequence was used in both experiments. We can therefore



(a) Mean of subjects' median correct response times by median split and sequential position. The correct RTs for the two performance groups were not significantly correlated, $r(8) = .17$, $p = .65$.

(b) The mean number of errors made at each position in the sequence split by performance group. The errors are highly correlated, $r(8) = .79$, $p < .01$.

Figure 4.7 | RTs and error rates by median split and sequential position. Note how much worse sequence position 5 was for the low-performing group relative to the next-worst position (8). Low-performers showed twice as many errors in position 5 as in 8, while the high-performing group showed only a 25% increase in errors. Error bars reflect +/−1 SE.

compare the scaled response time and accuracy data from the two experiments in Figure 4.8, which shows a similar pattern across experiments.

We examined errors and correct response times by their sequential position, and compared these to RTs from Experiment 1. Overall, there is a significant correlation $r(8) = .88$, $p < .001$, between correct RTs from the RL experiment and RTs from the cued SRT experiment. Comparing the cued RTs to the high- and low-scoring groups separately, revealed a difference between the groups. The cued SRT RTs do not correlate significantly with the high-scoring group's RTs, $r(8) = .51$, $p = .13$, but do correlate significantly with the number of errors made in the RL experiment, $r(8) = .83$, $p < .01$. The low-scoring group shows the opposite pattern. The cued SRT RTs correlated significantly with the RL correct RTs, $r(8) = .80$,

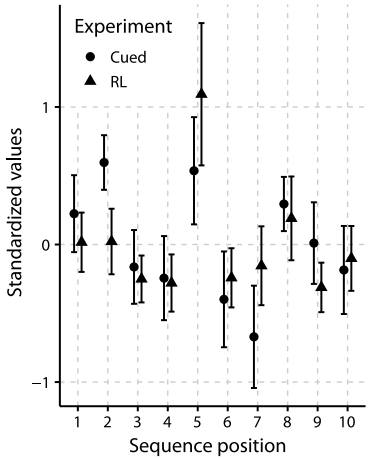


Figure 4.8 | Scaled mean number of errors in Experiment 2 (RL) against scaled correct RTs from Experiment 1's cued SRT paradigm (NB87) by sequence position. The number of errors per position and the correct RTs are significantly correlated, $r(8) = .64$, $p < .05$. Error bars show +/−1 SE.

$p < .01$, but not with the RL errors, $r(8) = .57$, $p = .09$. Comparing the two groups with each other revealed a significant correlation in errors, $r(8) = .79$, $p < .01$, but no significant correlation in RT, $r(8) = .17$, $p > .05$.

4.4 Models

Modeling environment

To compare human sequence acquisition with existing reinforcement learning models, we implemented three reinforcement learning models and a simple negative recency biased model (SCM; [19]) using PyBrain [139]. The environment contains all data regarding the targets, which it passes to the task, which in turn passes the current state of the environment to the agent, which selects the relevant action. The action is evaluated by the environment, which updates itself and passes a reward

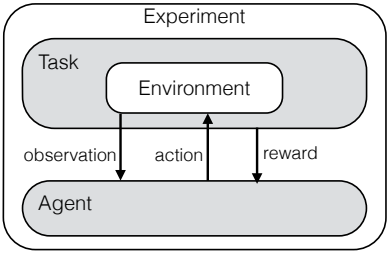


Figure 4.9 | Overview of the experimental setup for the reinforcement learning models. Each plated component is a PyBrain class, which interact with each other according to the arrows to simulate the same trial-and-error learning process that humans undergo.

to the agent. The reward is used to update the agent's strategy, and the model continues with the next step. We defined the reinforcement learning SRT task in this framework for our simulations, see Figure 4.9 for the specific design.

As in the human experiment, the data regarding the targets was only partially visible to the agent. The task acted as a veil through which a certain state would be observable. To a human participant, the current position in the sequence would be obvious, as it was colored differently from the other stimuli. At a minimum, the immediately prior occupied position was probably obvious as well, readily available in memory. Positions preceding that, however, might not be reliably accessible in memory. In the sequence we used (4–2–3–1–3–2–4–3–2–1), following Nissen and Bullemer [107], each position's identity is fully determined by the previous two positions. That is, one could perfectly predict the next position given only the two prior to it—assuming one has determined that there is a deterministic, periodically repeating sequence. The RL models we use rely on a set of third-order observations, assuming that the models know their current position and the two prior positions.

On-policy vs. off-policy learners

The reinforcement learning models differ in their learning component, which is contained within the agent and maintains a mapping between input states and action-values. For each given input state there are three action-values, corresponding to the number of movements that can be made by the agent. After receiving a reward, the agent updates the action-values using its learning algorithm. We tested three learning algorithms: SARSA [133], standard Q-learning, and Q(λ)–Q-learning with eligibility traces [168].

Off-policy learners such as Q-learning learn the value of the optimal policy independently of the agent's actions. They learn about the greedy policy, updating old action-values using the maximum of all action-values for the current state, while—depending on the action selection policy—it can stochastically select actions and explore.

The update rule in Q-learning updates Q for any state-action pair $<s, a>$ using an experience tuple $<s, a, s', r>$, with learning rate $\alpha \in [0, 1]$ and discount factor $\gamma \in [0, 1]$:

$$Q'(s, a) = (1 - \alpha)Q(s, a) + \alpha(r + \gamma Q[s', \underset{a'}{\operatorname{argmax}}(Q[s', a'])]) \tag{4.1}$$

In contrast, on-policy learners (e.g. SARSA) learn the value of the policy actually being carried out by the agent: instead of the maximum, they also take into account the action that was selected for the current state. In other words, it does not use the maximum attainable reward in state s' to update the Q-table, but instead chooses a' using the same policy it used to choose a. It therefore needs the experience tuple $<s, a, r, s', a'>$:

$$Q'(s, a) = (1 - \alpha)Q(s, a) + \alpha(r + \gamma Q[s', a']) \tag{4.2}$$

The eligibility traces in Q(λ) are temporary records of an event (e.g. an action or state) that help with temporal credit assignment by adding a trace to events that are eligible for learning updates. Theoretically, eli-

gibility traces link RL temporal difference methods (like Q-learning and SARSA) to Monte Carlo methods.

Simple condensator model

To investigate if perhaps an even more elementary mechanism could be responsible for participants' behavior, we also included a condensator model, introduced by Boyer et al. [19], and inspired by Dominey [37]. In this model, each target is assigned a corresponding unit, with activation ranging from .0 to 1.0. Summed activation across units is always 1.0, and all units were initialized at .25. Each step, the unit with the highest activation is chosen, and its activation is then distributed equally among the other three units.

These reinforcement learning models were chosen as simple baselines that differ somewhat in exploratory behavior and learning speed, and thus may be suitable to compare to human behavior which varied widely. As with the human participants, the simulated SARSA and Q-learners were tasked with iterating over the repeated sequence until the successful completion of 800 movements. For each model, a grid search over the parameters (learning rate α and discounting factor for future rewards γ) was used to find optimal values.

Modeling results

The best parameters found for the SARSA model ($\alpha = .01$, $\gamma = .98$) achieved a mean final score of 183 ($SD = 292$). The best parameters found for Q-learning ($\alpha = .38$, $\gamma = .98$) yielded a mean final score of 346 ($SD = 75$), while Q(λ) reached a mean final score of 369 ($SD = 53$, parameters: $\alpha = .001$, $\gamma = .95$, $\lambda = .99$). However, despite considerable learning by the end of the experiment, none of the models performed as well as the high-performing human learners, who averaged a final score of 652. Even the *maximum* scores achieved by the models were below the high-scoring humans average or maximum (human = 725; Q-learning = 473, Q(λ) = 440; SARSA = 477).

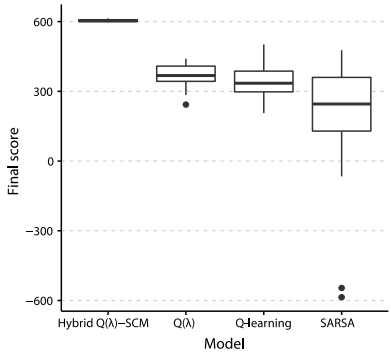


Figure 4.10 | RL task scores of the different models tested. A hybrid Q(λ)–SCM model performs better than all of the other RL models, but none of the models reach human performance.

We hypothesized that an RL model combined with a negative recency bias in early learning (with high levels of uncertainty) could perhaps yield better results. Using this technique, humans may be using a recency avoidance strategy in early learning, which would become less necessary after the sequence has been acquired. To investigate, we tested a hybrid model in which the SCM model would choose the next target when certainty (expected action value) of the RL model was low (defined by an optimized parameter: .61). This hybrid Q(λ)–SCM model averaged a final score of 604 (SD = 4). Results from all models are displayed in Figure 4.10.

Although these common RL models were unable to reach human-level performance, we thought it worthwhile to examine whether their error patterns resemble those of people. the mean number of errors made by each model at each position in the sequence, as was done earlier for humans. The errors made by the SARSA and Q-learning algorithms did not vary much by sequence position. Q(λ) made more errors in the middle of the sequence, but still did not resemble human error patterns.

4.5 General discussion

This paper introduced the trajectory serial reaction time task and found that it replicates the results of Experiment 1 of Nissen and Bullemer [107]. Thus, while the trajectory SRT paradigm retains the essence of the original SRT, it also affords the opportunity to measure a variety of more detailed statistics about subjects' continuous motions. Response trajectories can reveal uncertainty, predictive movements, reversals in decision, and other phenomena that may reveal the dynamics of the learning mechanisms at work. The present study examined the average trajectories of two isomorphic vertical movements that appear in the NB87 sequence, as well as in the random condition. The two movements have different subsequent stimuli in the NB87 condition, and were thus expected to show a sequential context effect: as participants learn where the next stimulus will be, they may start to move towards this response even as they finish the previous movement–as a piano player may reach for the next key while the current one is being sustained [145].

We found not only that the expected context effects had developed by late training, but also evidence of possibly strategic adaptive behavior in the random condition. Many participants in the random condition developed a re-centering approach after each response, waiting for the next (unpredictable) stimulus to appear. In a way this behavior is optimal, since the center of the screen is as close as possible to all stimuli. Some participants in both conditions showed this behavior to a limited extent early in training, but those trained on the NB87 sequence lost this behavior over time as they learned to predict the location of the subsequent stimulus–hinted at by the decrease in reaction times in this condition, and confirmed by the deviation in average trajectory towards the subsequent stimuli. Of the participants in the random condition, two groups could be identified: a centering group and a non-centering group. This might reflect differences in strategy similar to Tubau et al.'s [160] stimulus-based vs. plan-based control mode, or Dale et al.'s [34] reactive vs. predictive movements. How these different behavioral strategies are

related could be the focus of future research.

Overall, the behavioral results show a striking similarity to the Nissen and Bullemer [107] results. The pattern of reaction times over sequence position was strikingly similar to the pattern observed in the original study, although the movement reaction times were higher throughout training and participants showed less overall improvement. This can be explained through the mechanics of the paradigm: mouse movements require more time to be executed than single keypresses, and require some fine motor control and error correction. The sensitivity of the mouse can be adjusted to achieve a balance between RT and error; we used a very low sensitivity to reduce overall noise. Participants in the NB87 sequence condition nonetheless showed an increased number of errors during training, indicative of a speed-accuracy trade-off which was not present in the Nissen and Bullemer [107] results. It is possible that extending the training would eventually lead to a reduction of errors, as participants would gradually become aware of the sequence.

In Experiment 2, we adapted the trajectory SRT paradigm to be a reinforcement learning task. The task proved to be more challenging for some than for others, as indicated by differences in response times and accuracy. Those data also suggest that participants adopt different strategies, and tried to adapt when they were not learning. These findings are similar to those in Experiment 1: RT and accuracy were correlated across experiments. In particular, data from the high-performing participants compared remarkably well to Experiment 1, despite the task differences. The most notable similarity was the difficulty participants experienced with the fifth stimulus position.

A bimodal distribution of scores showed that half of the participants did really well, as they made very few errors after roughly 10 repetitions of the sequence. Block-by-block analysis of the response times showed a difference in speed-up across the experiment between groups, indicating the high-performing group learned the sequence much better than the low-performing group. The difference in response times to incorrect targets suggests the two groups might have used different strategies. The

rare but increasingly slow errors in the high-performing group suggest more time was spent figuring out the next stimulus, while the persistent and relatively fast errors of the low-performing group suggest participants may have adopted a probabilistic view of the task, randomly trying options instead of trying to learn a deterministic pattern.

Despite the major difference of the absence of cueing of the next response, performance in the RL experiment was quite comparable to performance in the cued SRT experiment. The pattern of correlations indicated a difference between the low- and high-performing groups that was not immediately obvious. Overall, the cued SRT response times are correlated to RTs and accuracy data from the RL experiment, whereas this is not true for both the low- and high-performing groups separately. We expect this is due to different strategies among groups, leading to a different pattern of speed and accuracy at different sequence positions.

In addition to our behavioral analyses, we tested three different reinforcement learning models to see if human behavior could be explained by simple, model-free responses to sequential stimuli. High-performing humans were still far better than the models, which on average scored roughly as well as the low-performing humans. SARSA had quite variable performance, but was lowest on average, while Q-learning with eligibility traces fared the best. Examining the models' performance by sequence position showed they did not correspond well with human errors in either group. This suggests that simple model-free reinforcement algorithms do not capture the process by which humans learn action sequences, even though they eventually converge on a proper solution. One explanation for this is the fact that the task and models used in studies like this do not fully capture the essence of human action learning, which is goal-directed by nature. Interestingly, a hybrid model in which a simple negative recency bias guides behavior in early training outperforms all reinforcement learning models. Future studies could shed light on the role of goals in the acquisition of such action sequences, and the way learning shifts from simple to more complex mechanisms, as has been shown to exist for single-step action (see, for example, Hommel

et al. [65] for one proposed mechanism of goal-directed action). The process by which humans acquire action sequences is subtle, can yield quite variable performance, and is not easily captured by simple learning algorithms. However, studying it is important, as most of human behavior is essentially sequential in nature.

CHAPTER 5

Predicting action plan formation in sequential reactive and reinforcement learning

Almost all types of everyday action can be considered sequential. From making coffee to using the bathroom, these complex actions consist of subactions that are completed one after another. The mechanisms by which we learn such action sequences and execute them has been the subject of investigation for many decades. An early theory by James [69] argued that elementary action units in a sequence are triggered by the sensory effects of the preceding unit. However, Münsterberg [102] noted that such an associative account is insufficient to explain sequential action because a directional element is required to successfully execute subactions in the correct order. Instead, he argued that the learning of action sequences relies on the acquisition of a motor program. Tubau et al. [160] suggested that these two approaches are

This chapter is an adaptation of the article *de Kleijn, R., Kuipers, M., Kachergis, G., & Hommel, B. (in preparation). Predicting action plan formation in sequential reactive and reinforcement learning.*

not mutually exclusive, but in fact reflect two different executive control modes that—under specific circumstances—can be strategically chosen.

5.1.1 Stimulus-based and plan-based control

Tubau et al. [160] compared James's stimulus-driven account of sequential action with the *prepared reflex* concept of Hommel [61], and referred to it as *stimulus-based control*. This type of executive control is characterized by the automaticity by which stimuli are attended to. Due to the highly automatized response to stimuli, the sequence itself is often not learned. Instead, what is learned is a strategy of delegating control to external stimuli [160]. In other words, people learn how to respond quickly to incoming information. *Plan-based control*, on the other hand, is assumed to rely on action plans, which are structured sequences of action effects [62, 96]. In contrast to stimulus-based control, representations in plan-based control are *internally* generated.

There is evidence to suggest that sequence learning does not rely on the prediction of sequences of external stimuli, but the prediction of the motor action to be performed. In other words, participants do not learn stimulus–event sequences, but in fact learn sequences of *responses*. As such, it is thought that sequence learning involves a shift from stimulus-based control to plan-based control, implying the generation of action plans by which participants can predict a sequence of responses even in the absence of stimuli [60, 104].

Tubau et al. [160] investigated this shift and its modulators in a comprehensive study consisting of five experiments. In a serial reaction time paradigm in which participants had to respond to the letter X appearing on the left or right side of the screen and responding with the appropriate hand, they presented participants with a repeating sequence of stimuli. In this sequence, location switches occurred four times more often than location repetitions, but stimuli were equally often presented to the left or right. They found that participants' control mode was influenced by instruction type, where intentional instruction (i.e. telling par-

ticipants that the shown sequence is deterministic, and is to be learned explicitly) induced plan-based control. Participants' control mode was assessed by the size of the frequency effect, which should be smaller under plan-based control. Participants having received intentional instructions showed a smaller frequency effect, which was attributed to the formation of an action plan. Also, these participants were more likely to have acquired explicit knowledge of the sequence, as they were able to verbally report the correct sequence at the end of the experiment[1].

However, plan-based control is not just a strategy that participants employ at their own choosing—task structure and demands have a large influence. For example, removing stimulus–response compatibility by using symbolic stimuli instead of spatially compatible stimuli seems to lead to plan-based control, as is evidenced by the elimination of the frequency effect. Also, playing irrelevant sounds that hamper symbolic encoding of the sequence prevents the successful formation of an action plan, leaving stimulus-based control the only viable mode of executive control [160]. In some circumstances (for example the exploratory paradigm discussed later), stimulus-based control is not a feasible strategy due to the lack of stimuli.

5.1.2 Studying sequence learning

The acquisition of action sequences has been the subject of study in domains ranging from linguistics [41, 136] to everyday action [18, 32], with perhaps the *serial response time task* (SRT, [107]) being the most popular paradigm.

In the SRT 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. In their original study, Nissen and Bullemer [107] compared a con-

[1]Although it should be noted that explicit sequence knowledge is not at all necessary for learning (see e.g. [89, 107])

dition using random stimulus locations with a condition using a repeating, deterministic sequence, and found evidence for implicit sequence learning: participants in the deterministic sequence showed larger reduction in response times than participants in the random condition.

Most of the sequence learning literature has focused on cued paradigms such as the SRT task, in which participants have to respond to sequences of stimuli that appear. However, it seems clear that sequence learning in daily life is often not learned by simply chaining stimulus–response associations [87]. Instead, acquiring new action sequences is better characterized as *exploratory*, in which people try several alternatives before discovering the correct one.

In one recent study, Kachergis et al. [77] adapted the SRT task to a reinforcement learning paradigm. In this task, participants were not *cued* by the stimuli, but had to *explore* the four alternatives to find out which one was correct. Participants could collect points by predicting the next stimulus correctly. A strong correlation was observed between behavior on the SRT task and its reinforcement learning adaptation in terms of response time and accuracy per sequence position. Interestingly, the final scores were bimodally distributed, suggesting that participants used different strategies. Although purely stimulus-based control is impossible in this paradigm, it is clear that the accuracy of participants' action plans showed a large range of variance. Although their study investigated both the SRT task and its reinforcement learning adaptation, the study had a between-subject design, making it impossible to examine characteristics of participants that produce effects that are common to both tasks.

5.1.3 The current study

In scenarios where both stimulus-based control and plan-based control are possible, participants may strategically (or perhaps even randomly) choose an executive control mode. In the current study, we investigated predictors of executive control mode in an SRT task and action plan formation in a reinforcement learning task in which plan-based control is

the only control mode available.

Earlier research has shown that visuospatial working memory capacity predicts both implicit and explicit sequence learning performance [16, 17]. In this study, we will look at visuospatial working memory capacity and IQ measurements as predictors of executive control mode that reflect cognitive limitations. One possibility would be that some participants simply do not have the cognitive capacity to form (long enough) action plans. Another possibility would be that control modes are chosen strategically or preferentially. The formation of an action plan might reflect individual differences in the need for structure. That is, some people may prefer to actively predict the future according to a plan or schema instead of waiting for stimuli to arrive, while others might want to delegate control to the external environment [105].

5.2 Method

5.2.1 Participants

Forty undergraduate and graduate students (13 males, 27 females) were recruited from Leiden University. Participants either received course credit or were paid 6.50 euro for participation. All participants had normal or corrected-to-normal vision. The total duration of the experiment was approximately 90 minutes.

5.2.2 Materials

In order to assess possible predictors of participant behavior, several tasks and questionnaires were administered.

Fluid intelligence

Fluid intelligence was estimated using a shortened, 10-minute version of the Raven's Standard Progressive Matrices (SPM) test [124]. It measures the individual's ability to form perceptual relations and for analogical reasoning. It is a widely used test to measure fluid intelligence, independent

of language and schooling, and is considered to have excellent reliability [24]. The number of correct responses in 10 minutes over all participants are normalized to a distribution with mean 100 and SD 15, resulting in an estimated IQ score.

Locus of control

To investigate the influence of an individual's locus of control on control mode, we administered the Levenson Multidimensional Locus of Control Scales [88], a 24-item questionnaire consisting of three subscales: (1) internality, (2) powerful others, and (3) chance. People who have an internal locus of control tend to perceive reinforcement as a result of one's behavior, while people with an external locus of control tend to perceive it as a result of factors beyond one's control. It could be hypothesized that people with an internal locus of control are more likely to engage in plan-based control, while people with an external locus of control are more environment-driven.

Personal need for structure

To assess participants' tendency to seek out structured ways of dealing with the world, we administered the Personal Need for Structure scale [158]. This questionnaire quantifies people's need for simple structure, and consists of 12 statements (e.g. "I enjoy having a clear and structured mode of life.") which the participant can either agree or disagree with, rated on a 6-point scale. It has been shown to have good reliability and validity [105]. It has been hypothesized that personal need for structure reflects a strategy for simplifying the world due to a general lack of intellectual abilities, but the correlation between the PNS scale and IQ seems to be minimal [105]. It is therefore more likely to reflect a strategy that participants can choose to employ, and participants who score high on this measure could be more likely to actively search for structure in action sequences.

Visuospatial working memory

We assessed visuospatial working memory using the computer task from Bo et al. [17]. In their study, which used an adaptation of the visual working memory task used by Luck and Vogel [92], a relationship was found between visuospatial working memory capacity and performance on a serial reaction time task. In this task, participants were presented with a sample array for 100 ms followed by a blank screen delay of 900 ms, after which a test array was presented for 2000 ms. Participants were asked to determine whether the test array was different or similar to the sample array by pressing either D or S. Arrays consisted of 2–8 colored circles, and for each trial the test array was either the same as the sample array or different with one of the colors changed. Visuospatial working memory capacity was calculated as K = array size × (hit rate – false alarm rate). The average K across all array sizes was computed to estimate visuospatial working memory capacity [17]. Participants completed 140 trials in total.

Trajectory SRT task

The trajectory SRT task is an adaptation of Nissen & Bullemer's serial response time task [107]. It maps the four buttons of the original SRT task to four squares located on the corners of a computer screen, requiring participants to move the mouse cursor to each square that lights up [74, 75]. Unbeknownst to participants, the sequence is a repeating sequence of 10 items. Speed-up over time compared to a condition with a random sequence is thought to reflect implicit learning of the sequence. In the current study, we used a different sequence (3–2–4–2–1–4–3–4–2–1) than in the original SRT task to prevent carryover effects between this task and the RL task. The complete task consisted of 800 movements (80 repetitions of the 10-item sequence).

In order to assess first-order frequency effects, the sequence was designed in such a way that it consisted of 8 straight movements, and 2 diagonal movements. After completing the 800 movements, participants were

asked if they noticed any structure within the experiment, and if so, were asked to reproduce the sequence.

Reinforcement learning task

The RL task is an adaptation of the trajectory SRT task (see above), with the difference being that the next stimulus is not cued, but to be discovered by the participant through trial-and-error [77]. Participants moved to one of the four squares, and received feedback by the square turning green in the case of a correct movement, and being returned to the center of the screen in the case of an incorrect movement. Points were awarded for correct movements (+1 point), and deducted for incorrect movements (–1 point), and participants were instructed to maximize their amount of points. The amount of points collected was continuously visible to the participant, their progress in the task, however, was not. The task ended after 800 correct movements of the original SRT sequence (4–2–3–1–3–2–4–3–2–1).

A participant having knowledge of the sequence before starting and who never made a mistake would therefore make 800 movements directly to valid targets, receiving a theoretical maximum score of 800 points. A participant with no memory of even the previous target they had tried could make an infinite number of mistakes, never finishing the experiment. If participants would not repeat the same invalid target more than once when seeking each target (i.e. an elimination strategy), a participant would expect on average to score 0 points, as the expected value of completing one movement successfully is 0 using this strategy[2]. Participants were not told that there was a repeating deterministic sequence, let alone details such as how long the sequence was.

[2] 33% chance of success in one try (+1), 33% chance of success in two tries (–1+1), and 33% chance of success in three tries (–1–1+1).

5.2.3 Design and procedure

All participants performed both the trajectory SRT task, as well as the reinforcement learning task. The order of the two tasks was counterbalanced over participants, and the two tasks used different sequences to prevent carryover effects.

Participants were seated at a computer after having given their informed consent. All subsequent tasks were performed on the computer. First, the Personal Need for Structure questionnaire was completed, followed by the Levenson Multidimensional Locus of Control questionnaire, the visuospatial working memory task, and Raven's SPM. After this, participants were given a 5-minute break. Participants then completed, in counterbalanced order, the trajectory SRT task and the reinforcement learning task.

5.3 Results

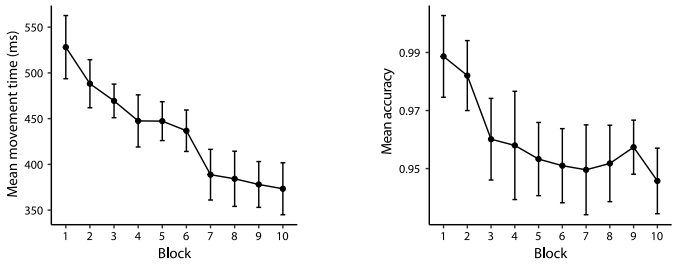
5.3.1 Trajectory SRT task

Data preparation

Prior to analysis, movement times >1500 ms were removed, and the experiment was divided into 10 blocks of 8 sequence repetitions. As an analysis of data collected earlier (described in Chapter 4) using a random sequence showed no significant difference in movement times between straight and diagonal movements, there was no correction applied for the somewhat larger distance required to make diagonal movements.

Response times

Comparative analyses were performed using the means of participants' median movement time, with the movement time defined as the time between cue onset (stimulus changing color) to touching any part of the stimulus with the mouse cursor. Median movement time to a target was 464 ms (*SD* = 223 ms). Participants' movement time decreased from 546



(a) Participants' movement time decreased over time, indicating learning of the sequence.

(b) Error rates increased during the first three blocks, but remained relatively stable during the rest of the task.

Figure 5.1 | Movement times and accuracy for the trajectory SRT task. Error bars indicate within-subject 95% CI.

ms in the first block to 413 ms in the tenth block, indicating learning of the sequence, $F(9, 360) = 15.80$, $p < .001$, $\eta^2_G = .126$.

Accuracy was high across all blocks of the experiment, but especially so during the first two blocks. There was an effect of time on accuracy, $F(9, 360) = 4.50$, $p < .001$, $\eta^2_G = .042$, indicating some degree of speed-accuracy tradeoff. However, after the third block movement times are still decreasing, while accuracy remains stable. Both movement times and accuracy are shown in Figure 5.1.

Explicit sequence knowledge

Participants were grouped into an implicit knowledge group and an explicit knowledge group. Only those 13 participants who could correctly recall the complete repeating sequence after having completed the task were considered to have explicit knowledge. Participants with explicit sequence knowledge had a significantly larger working memory capacity (2.87 vs. 2.25, $t(28.08) = 2.95$, $p = .006$, $d = 1.11$), but did not differ on estimated IQ, the Levenson Multidimensional Locus of Control scales,

Factor	*df*	*F*	η_G^2	*p*
Block	9, 342	23.94	.17	< .001
Block × Knowledge	9, 342	8.37	.07	< .001
Frequency	1, 38	106.00	.09	< .001
Frequency × Knowledge	1, 38	4.43	.004	.042
Frequency × Block	9, 342	2.75	.005	.004
Knowledge	1, 38	6.44	.089	.015

Table 5.1 | Results of analysis of variance on movement times.

or Personal Need for Structure scales (*t*s < .81, *p*s > .42).

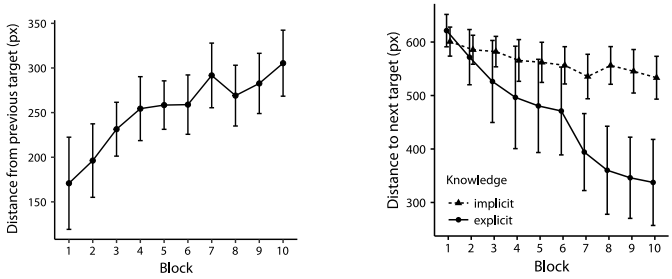
Modes of executive control

Similar to Tubau et al. [160], we used frequency effects (i.e. the facilitation of responses to frequent (straight) compared to infrequent (diagonal) transitions) to determine whether participants engaged in either stimulus-based or plan-based control. An analysis of variance revealed main effects of block, frequency, and knowledge on movement time (see Table 5.1). Overall, participants with explicit sequence knowledge had faster movement times (M = 398 ms) than participants without (M = 485 ms), and frequent (straight) movements were performed faster (M = 417 ms) than infrequent (diagonal) movements (M = 496 ms).

Predictive movements

As the task progressed, participants made an increasing amount of movement during the ITI—in the absence of a stimulus, $F(9, 342) = 6.53$, $p < .001$, $\eta_G^2 = .053$. Total ITI (predictive) movement, defined as the distance from the previous target at the onset of the next target, increased from 171 pixels in block 1 to 305 pixels in block 10. There was no main effect of knowledge. Results are shown in Figure 5.2a.

Similar to Dale et al. [34], we can then define *correct* predictive movement as the distance to the next target at target onset. An analysis of



(a) Participants made increasingly larger movements during the ITI.

(b) Larger movements during the ITI reflect correct prediction of the next stimulus, as initial distance to the stimulus decreased over time.

Figure 5.2 | Predictive movements in the trajectory SRT task. Participants made increasingly larger predictive movements, which reflects correct prediction of the next stimulus. This effect was stronger for explicit than for implicit learners. Error bars indicate 95% CI.

variance using block and knowledge as factors shows a main effect of block, meaning that distance to next target decreased from 609 pixels to 474 pixels, or that correct predictive movement increased over time, $F(9, 342) = 32.36$, $p < .001$, $\eta^2_G = .22$.

In the final block of the task, predictive movements (defined as movements larger than 300 pixels during the ITI, but not necessarily toward the correct target) appeared to show a mixed distribution over participants. Where some participants hardly showed any movement during the ITI, others had almost half of all their movements classified as predictive. Hartigan's dip test of unimodality [58] confirms this observation, $D = .079$, $p = .038$.

While implicit learners hardly increased their correct predictive movements, explicit learners showed a strong increase over time, as evidenced by a block × knowledge interaction, $F(9, 342) = 14.00$, $p < .001$, $\eta^2_G = .11$. Re-

sults are shown in Figure 5.2b.

Centering behavior

In Chapter 4, participants in the random condition showed more centering behavior than those in the deterministic condition. This finding suggested that centering is a strategy that can be employed in the absence of reliable sequence knowledge, minimizing the distance to possible targets. Indeed, centering behavior, defined as the proportion of the ISI spent in the center 100 × 100 pixels of the screen, was highest for participants without explicit sequence knowledge, $t(30.46) = 2.34$, $p = .026$, $d = .85$.

5.3.2 Reinforcement learning task

As explained in Section 5.2.2, maximum score on the reinforcement learning task was 800, with the most basic elimination strategy leading to 0 points. Mean score was 525, ranging from 140 to 774 points. Distributions of scores was non-normal, with a large group of participants scoring 700 points, and a group scoring quite low. For subsequent analyses, a midpoint split on 457 points was performed, dividing the participants into low and high performers.

Predicting task performance

Low performers on the reinforcement learning task had a significantly lower estimated IQ of 91.4, compared to high performers with an estimated IQ of 104.9, $t(39) = 3.06$, $p = .004$, $d = .98$. Also, low performers had a significantly lower visuospatial working memory capacity of 2.13 vs. the high performers' 2.65 capacity, $t(39) = 2.40$, $p = .021$, $d = .77$. Results are shown in Figure 5.3. IQ and visuospatial working memory capacity were uncorrelated, $r(39) = .213$, $p = .181$.

There was no difference between the two groups on the Levenson Multidimensional Locus of Control scales, $t(39) = .27$, $p = .790$, and no difference on the Personal Need for Structure scale, $t(39) = .28$, $p = .780$.

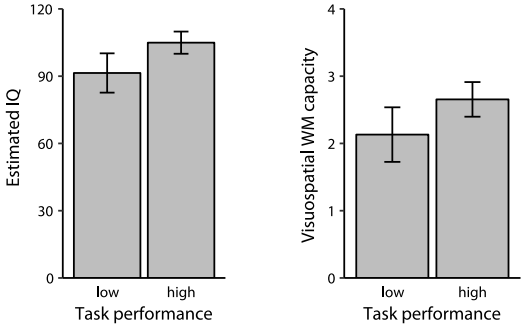


Figure 5.3 | Differences in estimated IQ and visuospatial working memory capacity between low and high performers on the reinforcement learning task. Error bars indicate 95% CI.

Explicit sequence knowledge was strongly related to task performance, as the 23 participants with explicit sequence knowledge had a far higher final score (M = 634) than participants without explicit knowledge (M = 375), $t(24.67) = 4.61$, $p < .001$, $d = 1.86$.

Stimulus- vs. plan-based control In the SRT task, two measures of executive control mode are used. First, explicit knowledge of the sequence is considered to be an indicator of a plan-based control mode. Second, the amount of correct predictive movements is evidence of the existence of an action plan, implying a plan-based control mode.

Participants with explicit sequence knowledge in the SRT task were more likely to have acquired explicit sequence knowledge in the reinforcement learning task, McNemar's $\chi^2(1, N = 40) = 4.5$, $p = .034$. This suggests that the acquisition of explicit knowledge in both tasks relies on a common mechanism or dependency. However, the amount of correct predictive movements in the SRT task was not related to the final score in the reinforcement learning task, $r(38) = -.025$, $p = .880$, nor did explicit knowledge in the reinforcement learning task relate to correct predictive movements in the SRT task, $t(38) = 1.32$, $p = .195$.

In summary, participants using plan-based control in the SRT task did not score higher on the reinforcement learning task, but participants with explicit knowledge formation in the SRT task *were* more likely to acquire explicit knowledge on the RL task. This suggests that predictive movements and explicit knowledge do not similarly reflect successful plan formation, and may not be equally good indicators for a plan-based control mode.

5.4 Discussion

5.4.1 Movement trajectories

Learning was evident in both the trajectory SRT task and the reinforcement learning task. In the trajectory SRT task, the findings of Tubau et al. [160] were replicated. The trajectory paradigm allowed us to find further evidence for a plan-based mode of control: participants made increasingly large movements toward the next stimulus, but participants with explicit knowledge of the sequence did more so than those with implicit knowledge. Instead, participants without explicit knowledge showed centering behavior during the ITI, moving the mouse to a position equidistant to all possible targets.

This centering strategy has been described in the literature (e.g. [34]), but has not before been associated with quality of prediction or sequence knowledge. Our results show that this behavior is a function of explicit sequence knowledge. It has been suggested that this centering behavior is an artifact of the spatial layout of the task, but we hypothesize that the centroid of any polygon defined by response locations should be a preferred (optimal) resting place when waiting for an uncertain stimulus. Future studies should be able to shed light on this theory.

5.4.2 Limitations preventing plan formation

In the reinforcement learning task, final scores showed a bimodal distribution, similar to what has been reported in [77]. The low-performing

and high-performing groups differed in IQ and working memory capacity, but did not differ in personal need for structure or locus of control. This suggests that sequence learning performance in an exploratory paradigm is not determined by personal characteristics or preferences, but by cognitive limitations.

In both the SRT task and the reinforcement learning task, explicit sequence knowledge was predicted by visuospatial working memory capacity. Earlier research by Bo et al. [17] showed a relationship between visuospatial working memory capacity and performance on a non-trajectory SRT task, but the current study shows that this holds in an exploratory paradigm as well and predicts explicit sequence knowledge. The observation that participants who were more likely to acquire explicit sequence knowledge in the SRT task were also more likely to acquire it in the reinforcement learning task further corroborates this finding.

5.4.3 Suggestions for future research

A promising approach to investigating this relationship is by modeling the learning process in the reinforcement learning task (see Chapter 4 for an example). IQ and visuospatial working memory could be compared to the learning rate and state space in reinforcement learning models that are fit to the performance of individual participants. This may shed further light on the exact mechanisms that explain the wide range of performance on exploratory sequence learning.

Another possible explanation of the diverse learning outcomes could be rooted in different beliefs about the task. Participants were not told that the response locations would be a repeating, deterministic sequence. They may have instead believed it was to some extent probabilistic—as many psychological tasks are. Different assumptions about the task may lead participants to arrive at different strategies, with variable success in the task. Participants expecting a random sequence may be less inclined to predict the next stimulus and are—in the current paradigm—indistinguishable from participants expecting a deterministic sequence

but unable to learn it due to intellectual limitations. However, manipulating these variables is straightforward and could be an interesting avenue for future research.

CHAPTER 6

Optimized behavior in a robot model of sequential action

Sequential action is one of the cornerstones of human everyday action. Most of our everyday activities, such as coffee making or driving a car, can be regarded as complex but sequential actions. How humans perform these sequential actions has been the subject of study for at least a century. As described in Chapter 2, sequential action can be represented on a symbolic (what will my next action be?) level, as well as a subsymbolic, sensorimotor (what motor parameters should I use?) level [173]. Interaction effects between the two levels of representation have been observed, and integration between the two is necessary to produce smooth sequential action. Due to their embeddedness (i.e. an implementation in a physical environment), (virtual, humanoid) robots are suitable subjects for developing and investigating models of behavior in which interaction with the environment is important (see [9] for an extensive overview). Robot paradigms have been successfully used to investigate psychological phenomena that require such embeddedness like hand–eye coordination [84], object handling [67], and imitation learning

This chapter is an adaptation of the article *de Kleijn, R., Kachergis, G., & Hommel, B. (in preparation). Optimized behavior in a robot model of sequential action.*

[138]. Used in the proper way, they hold promise to investigate the relation between symbolic planning of actions and the subsymbolic execution of these actions.

6.1.1 Optimization of motor control

The specific motor parameters used in the execution of motor commands is influenced by several effects and constraints. A good example is the *end-state comfort effect* [30], in which the grasp location of an object is a function of the expected end state of the arm. In other words, the arm end state is optimized. Other optimization is seen in the form of contextual lip rounding [35], where the lips are rounded in preparation for pronouncing the /u/ sound well in advance, and bending of mouse trajectories when sequentially reaching for stimuli with a mouse cursor by predicting its future location [75].

Other authors have investigated such predictive movements using similar measures. In earlier work, Dale et al. [34] used a paradigm similar to the one used in Chapters 4 and 5, with different levels of sequence complexity[1]. As sequence complexity decreased, participants were found to make larger predictive movements (i.e. movements toward the next stimulus) and be more likely to have explicit sequence knowledge. Participants not making predictive movements were observed to move their mouse cursor to the center of the screen, equidistant from all stimuli. The authors mention that "even participants with low pattern awareness engaged in this form of behavior" (p. 204), but our findings described in Chapter 5 show that it is specifically this group without explicit sequence awareness that engages in this type of behavior. Duran and Dale [38] agree with this finding, and report that this centering strategy is likely employed to compensate for lack of sequence knowledge, making it impossible to accurately predict the next target. In those circumstances, moving the mouse cursor to a position equidistant to all alternatives would be an effective strategy.

[1] More specifically, a measure of grammatical regularity was used inverse to the first-order entropy of the sequence, as used in [70].

6.1.2 The current study

In the current study, we directly manipulated prediction quality in a sequential reaching task with a virtual robot hand controlled by an artificial neural network. The task was similar in nature to the task described by Dale et al. [34] and Kachergis et al. [77]: reaching for targets that appeared or changed color in a repeating sequence.

In any modeling problem using artificial neural networks, the connection weights between the artificial neurons (or units) have to be optimized. In other words, the goal is to find those connection weights that cause the artificial agent to produce the behavior that most closely approaches the required behavior as measured by a fitness or cost function determined by the researcher. One of the most popular methods for determining suitable connection weights is known as *backpropagation* [132], in which the network is presented with an input vector, after which the output produced by the network is compared to the desired output, and network weights are then updated according to their error value, starting with the output units and working back through the network.

Evolutionary algorithms such as *neuroevolution* (e.g. [5]) can find suitable network weights not by directly calculating an error measure for each input–output pair presented to the network, but by quantifying the performance of agents controlled by the network. In its most simple form, the method of neuroevolution generates a large number of agents with randomly initialized networks and quantifies how well they perform on the required task during a fixed period of time. Next, the best performing agents are allowed to "reproduce", and are copied to the following generation in a slightly modified way (e.g. by adding random noise to the connection weights). In subsequent generations, this procedure is repeated until some predefined fitness criterion is reached. Neuroevolution is considered an efficient approach to solving reinforcement learning problems. Past studies have shown neuroevolution to be faster and more efficient than reinforcement learning methods such as Q-learning (see Chapter 4) on several tasks, including robot arm control [99, 100, 150].

Evolutionary algorithms have been used to simulate a wide range of psychological phenomena, ranging from reciprocity [4] to selective attention [114] and category learning [101].

6.2 Method

6.2.1 Task design

The task used for the virtual robots was analogous to the task described by Kachergis et al. [77]. It was designed as an environment of size 50 × 50 represented in continuous space (i.e. as floating-point values). Over the course of one run of 500 discrete time steps, target stimuli appeared sequentially in one of the four corners of the environment (distance 10 from the environment border), following a simple repeating 1–2–3–4 sequence. In one condition, networks were provided with accurate information about the next stimulus. In a second condition, the information was not predictive of the next stimulus. In a third condition, no information about the next stimulus was provided to the network. The exact implementation is described below under *Network design*.

A virtual robot arm was to touch the target (come within a square of size 6 × 6 centered on the target) as quickly as possible. After touching a target, no targets were visible for 20 time steps as an inter-stimulus interval (ISI), after which the next target would appear. Every run (one network-controlled virtual robot arm performing the task for 500 time steps), the starting location was initialized to the center of the screen. During each run, the amount of targets touched and the total distance moved was calculated. Also, to encourage fast movement, a reward with decaying value was associated with each target. Rewards were initialized to value 100, decreasing by 1 with each time step. After completion of the run, network fitness was calculated by

$$\textit{fitness} = \textit{touched stimuli} + \textit{total reward} - (.0001 \times \textit{distance moved})$$

An agent with perfect prediction capability (i.e. immediately touching the stimulus that just appeared by already being in its location) would

therefore be able to reach a theoretical maximum fitness score of 2525.

6.2.2 Network design

The virtual robot arm was controlled by a two-layer feedforward neural network with four sensory neurons, two prediction neurons, eight internal (hidden) neurons, and two motor neurons (see Figure 6.1). All sensory and prediction neurons were normalized in the range [0.0, 1.0], with Gaussian noise sampled from $N(0, .05)$ added to the input. The two motor neurons were truncated to the range $[-2.0, 2.0]$, and allowed for movement in the two-dimensional plane. For simplicity we did not model the kinematics of an articulated effector.

The input to the two prediction neurons was constant (i.e. also present during the ISI) and represented either (1) the correct location of the next stimulus, (2) the location of one of the four stimuli, randomly chosen, or (3) a constant input of [0.0, 0.0]. So although in the second condition the prediction neurons were provided with the location of a stimulus, this location was not informative of the actual location of the next stimulus. These conditions will be referred to as accurate prediction, random prediction, and no prediction, respectively.

The output O_j of a hidden or motor neuron j was determined by the sigmoid activation function

$$O_j = \frac{1}{1 + \exp(-\sum_{i=1}^{N} w_{ij} O_i - b_j)} \tag{6.1}$$

in which N represents the number of input neurons i, O_i their output, w_{ij} the connection weight from i to j, and b_j the bias. Of the four sensory neurons, two were used for sensing the target, and two for sensing the location of the agent.

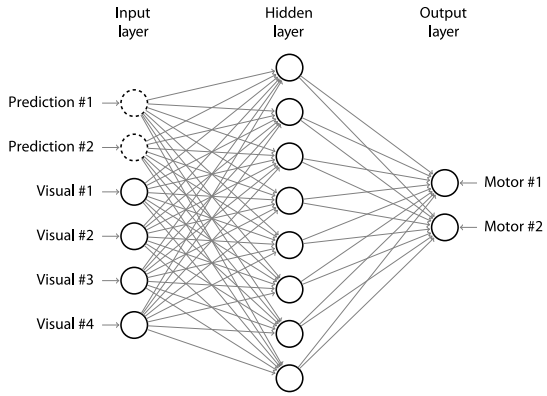


Figure 6.1 | Two-layer feedforward network architecture used. Six input units (two prediction units and four sensory units), eight hidden units, and two output units controlled the virtual robot arm.

6.2.3 Evolution of the network

Network weights were optimized using a neuroevolution algorithm using a direct encoding scheme (i.e. there was a one-to-one mapping of genotype to phenotype) similar to Nolfi et al. [109]. Although direct encoding schemes have been criticized for being biologically implausible [108], and having difficulties with scalability[2], direct encoding provided a good trade-off between simplicity and performance for the relatively simple networks used in this study. The initial population consisted of 100 networks with weights uniformly random $\in [-2.0, 2.0]$. For each subsequent generation, the twenty networks with the highest fitness value were allowed to reproduce by generating four copies each, with Gaussian noise sampled from $N(0, .3)$ added to the network weights. In addition, each of the twenty best networks was kept unmodified and added to the next generation, keeping the population size a constant 100. In pseu-

[2]The search space in direct encoding schemes increases exponentially with network size.

docode, the evolutionary algorithm was:

Algorithm 1: High-level description of neuroevolution algorithm

```
initialize 100 networks with random weights;
for 1000 generations do
    foreach network do
        evaluate fitness;
    end
    sort networks by fitness;
    for 20 best networks do
        generate 4 mutated copies;
        generate 1 identical copy;
    end
end
```

All simulations were run 30 times per condition, so a total of 90 simulations were run.

6.3 Results

Maximum fitness of the networks differed between conditions, $F(2, 87) = 9.29$, $p < .001$, $\eta^2_G = .176$. Post-hoc pairwise t-tests showed that networks with accurate predictions fed into the prediction neurons developed a higher maximum fitness ($M = 1868$) than networks with no prediction ($M = 1262$), $t(58) = 2.76$, $p = .008$, $d = .72$, and than networks with random prediction ($M = 947$), $t(58) = 4.12$, $p < .001$, $d = 1.08$. These differences remained significant after Holm–Bonferroni correction.

Figure 6.2 shows the evolution of fitness over time. Although the networks with no prediction evolved somewhat faster than networks with accurate prediction, maximum fitness leveled off after ~250 generations. For the networks with accurate prediction the network weights evolved slower, but surpassed the fitness of the non-predicting networks after 320 generations and continued to increase. Networks with random prediction evolved slower overall, and attained lowest maximum fitness.

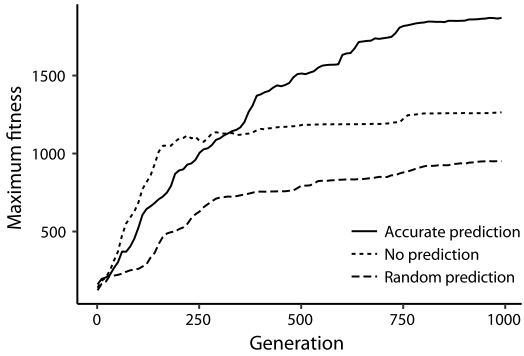


Figure 6.2 | Networks with accurate prediction attained higher maximum fitness than networks with no prediction or random prediction. These networks evolved to make efficient use of the information from the prediction neurons. Displayed are means over 30 simulations per condition.

Centering behavior differed between conditions, $F(2, 86) = 8.09$, $p < .001$, $\eta^2_G = .158$. Post-hoc pairwise t-tests showed that networks with accurate prediction spent a smaller proportion of ITI time in the center 10 × 10 units ($M = .195$) than both networks with no prediction ($M = .415$), $t(57) = 4.64$, $p < .001$, $d = 1.23$, and networks with random prediction ($M = .340$), $t(58) = 2.96$, $p = .004$, $d = .778$. These differences remained significant after Holm–Bonferroni correction. The networks with no prediction and random prediction did not differ significantly, $p = .277$. Results are shown in Figure 6.3.

Movement across the environment is displayed in Figure 6.4. The networks with random prediction (Figure 6.4b) learned that the information provided was not informative, and reached their maximum fitness by returning to the center of the environment after touching each stimulus, whereas networks with accurate prediction (Figure 6.4a) moved toward the next target, waiting for it to appear.

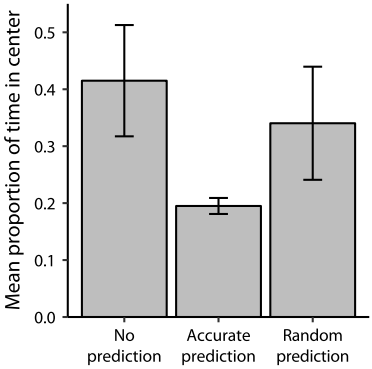
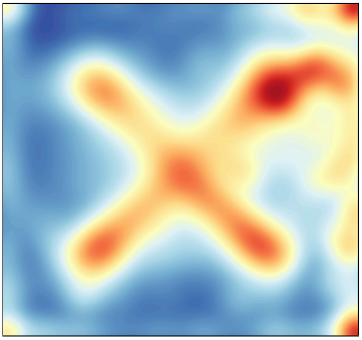
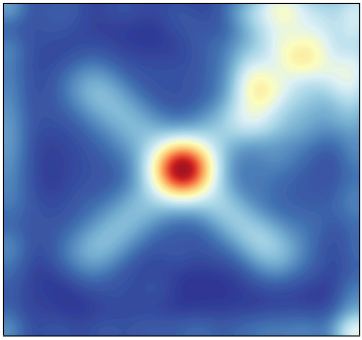


Figure 6.3 | The mean proportion of ITI time spent in the center of the screen for all three conditions. Networks with accurate prediction spent less time in the center. Error bars indicate 95% CI.



(a) In the condition with accurate prediction, position density is clustered around the stimuli, indicating active movement toward stimuli.



(b) With random prediction, the networks evolve to produce centering behavior. Most time is spent in a position equidistant to all targets.

Figure 6.4 | Density heat map showing the relative amount of time spent across locations in the accurate prediction and random prediction conditions, ranging from blue (little time spent) to red (most time spent).

6.4 Discussion

In this study, we investigated the behavior found in earlier work by Duran and Dale [38], Dale et al. [34], and the work described in Chapters 4 and 5. These studies describe a centering behavior in which participants moved their mouse to the center of the screen under some circumstances. In Chapter 5, we describe how this seems to be related to the quality of the action plan, or the capability to predict the next stimulus. This also makes sense on a theoretical level, as a centered position, equidistant to all possible stimuli is optimal under maximum uncertainty.

In the current study we evolved artificial neural networks that controlled a robotic arm, with a task analogous to the one used in Chapters 4 and 5. In one condition, an accurate prediction of the next stimulus was provided to the network as part of the input. In the second condition, the input given was randomly determined, and unrelated to the next stimulus. In a third condition, input to the prediction neurons was kept constant at zero. Under the last two conditions, centering behavior developed, with the networks that were provided random input and networks that were given no input developing the same centering strategy as human participants in Chapters 4 and 5 that had not developed explicit sequence knowledge. In summary, we showed that centering behavior evolved in a robotic arm controlled by an artificial neural network as a function of prediction quality, analogous to the findings described in Chapters 4 and 5.

Future research could shed light on the differences between the random prediction condition and the no prediction condition. From our results, it seems that performance was worse under the random prediction condition (although not significantly so), and developed more slowly. Apparently, the networks had trouble ignoring the dynamic, but uninformative input. In comparative studies with human participants, it would be interesting to distinguish between participants who *know* that they are unaware of the sequence (no prediction), and participants who are

actively, but unsuccessfully, trying to predict the sequence (random, or at least partly incorrect prediction).

CHAPTER 7

Summary and general discussion

7.1 Summary of this dissertation

The research described in this dissertation tried to shed light on the relation between complex action control in humans and robots. Taking the acquisition of action sequences as an example, a paradigm for the study of sequential action was introduced, and several models were discussed that can account for sequence learning and execution.

7.1.1 How human and robotic complex action control are related

First, the main obstacles in the way of autonomous, everyday action execution by robots were discussed from a cognitive psychological viewpoint in Chapter 2. Four main categories of problems are identified that need to be dealt with in order to make truly flexible, autonomous robots: (1) the integration of symbolic and subsymbolic planning; (2) the integration of feedforward and feedback planning and execution mechanisms; (3) the structure of action representation; and (4) the contextualization of action control.

Early AI planners, such as STRIPS [47], were designed to reach an intended goal state from an initial state through symbolically represented subac-

tions. This symbolic nature of action representation has many advantages: it allows, for example, for easy manipulation of action components leading to efficient planning. Early approaches in the study of human sequential action also assumed a symbolic representation of action sequences, with subsymbolic (sensorimotor) triggers responsible for timing. Both James [69] and Washburn [167] suggested a *chaining theory* of sequential action, in which the sensory feedback produced by executing the subaction at t_0 would trigger the execution of the subaction at time t_1. However, several empirical findings seem to be incompatible with a chaining account of sequential action. For example, such models cannot account for context effects as found in studies into anticipatory lip rounding, in which facial muscles adapt to sounds that are to be produced later in time [14] or Gentner's typewriting studies that showed a large amount of movement in anticipation of subactions several units ahead [51]. Instead, models that integrate symbolic and subsymbolic representations such as the typewriting model suggested by Rumelhart and Norman [131], seem to be more promising. In this model, the correct temporal order of subaction execution is ensured by feedforward inhibition.

As the field of robotics advances from repetitive, predictable actions such as factory work to highly dynamic and complex actions in everyday life, feedforward control systems alone are no longer sufficient. On the other hand, feedback systems are often slow as they require information from the environment to be produced and detected. Successful integration of feedforward and feedback control systems is needed to create agents that are both fast and adaptive. The existence of feedforward planning mechanisms in humans is demonstrated by the relation between onset delay and sequence complexity in finger and arm movements [59], as well as Eriksen et al.'s [44] linguistic studies on number pronunciation. However, feedback control mechanisms are essential for filling in parameters unavailable or unreliable at planning time, such as object weight and required grip strength. Hybrid architectures, in which skeleton action plans are generated by feedforward mechanisms, and where parameters are filled in by feedback processes seem to combine the best of both

worlds [53, 64].

Another difficulty in complex action planning is that the meaning and purpose of subactions vary with the goal that they serve to accomplish. In AI planners, the function of goals is to guide the selection of task components, and in cognitive processing models such as ACT-R goals reduce the search space, making task preparation more efficient [33]. Some authors have argued against the representation of goals for two reasons [18]. First, goals themselves may be context-dependent, and as such require different subactions to accomplish them. Second, many everyday activities such as taking a walk do not always have clearly defined goals. Others, however, emphasize that it is the representation of goals that makes useful action plan manipulation such as subaction substitution or skipping possible [33]. Alternatively, *implicit* goal representation from a TEC viewpoint can be viewed as a kind of "intentional weighting" mechanism in which relevant features are activated more than others, priming the agent to execute different subactions [65, 95]. Whatever the exact nature of goal representation, it is clear that some form of end-state representation is necessary to generate flexible behavior.

Chapter 3 discussed how the relationship between cognitive psychology and cognitive robotics developed over time. After breaking away from philosophy, psychology found itself depending on unreliable, subjective information. In a push toward reliable, empirical observation as the basis of a scientific psychology, behaviorism emerged as the method that could put psychology on par with the natural sciences. However, behaviorism proved untenable as a general theory of human behavior as it could not account for fundamental cognitive processes such as language and memory, leading to what is now known as the neocognitive revolution. Meanwhile, in the 1950s the field of artificial intelligence arose from cybernetics, mathematics, and computer science, and over the following decades expert systems such as MYCIN and symbolic AI (now known as GOFAI, *good old-fashioned AI*) were able to show impressive results. Also, computers were slowly beginning to gain public interest. Cognitive psychologists started to wonder if humans are like computers: input–output devices

with sensory information as input and behavior as output, known as the so-called computer analogy. Meanwhile, roboticists were considering animal behavior as a foundation for robot control. Some early cognitive robots were roughly inspired by biology [21], but even more specific parallels could be drawn between humans and robots.

The problem of integrating feedforward and feedback control in robotics had gained interest as task demands for robots became less predictable. Where the absence of a feedback loop in a factory environment may not be a big problem as long as all manipulanda are in the correct location and orientation, feedback is required in almost all situations in the outside world. Brooks' *subsumption architecture* [21] was a response to traditional GOFAI and showed that complex behavior could emerge without the traditional separation of feedforward and feedback systems. However, this architecture worked for rather low-level behavior such as wandering, avoiding, and homing, and it is unclear how well it would scale up to more complex situations. More complex, goal-directed behavior in robots is usually the product of a *planner*[1]. This component takes an intended state, compares it with the initial state, and determines the actions to take in order to successfully reach the intended state. Traditional planners such as STRIPS fail when one of the subactions cannot be successfully completed, and backtrack to try alternative subactions.

7.1.2 Empirical studies on sequence learning

One of the most widely used paradigms in sequence learning is the serial reaction time (SRT) task [107]. In this task, participants are asked to press the button associated with one of four horizontally distributed stimuli. Unbeknownst to the participants the four stimuli appear in a repeating, deterministic sequence. Over time, participants show a larger decrease in response times compared to a random sequence, indicating learning of the sequence. However, due to the discrete nature of this task it is

[1]Although both Brooks [22] and Braitenberg [20] are excellent examples of apparent complex behavior *without* a planner.

impossible to investigate interstimulus processes such as prediction or context effects [146].

In Chapter 4, we described an adaptation of the SRT task into the continuous domain. Instead of four discrete buttons associated with stimuli, we presented four squares in the corners of a computer screen, with the instruction of moving the mouse cursor as fast as possible to the square that changes color. This type of data collection allows researchers to capture the temporal dynamics of cognitive processes and the interaction between them [48, 146, 148]. First, we were able to replicate Nissen & Bullemer's [107] original findings: more speedup in the deterministic, repeating sequence than in a random sequence. Second, we showed that this speedup was due to predictive responses made during the ITI, and that participants employed different strategies. While some participants actively moved the cursor to the next target during the ITI, others used a centering strategy in which they moved cursor to a central location equidistant from all possible alternatives, a phenomenon reported earlier in the literature [34, 38].

Due to the questionable ecological validity of the SRT task—after all, everyday sequence learning is not often characterized by merely responding to attention-grabbing stimuli—we adapted the SRT task to a reinforcement learning paradigm. In this task, participants no longer could respond to squares changing color but had to actively explore the alternatives, receiving a 1-point reward when choosing the correct alternative, and a reward of –1 for choosing an incorrect alternative. Participants varied widely in the amount of points collected. To investigate possible causes, we fit three model-free reinforcement learning models: (1) Q-learning, (2) SARSA, and (3) Q-learning with eligibility traces.

Reinforcement learning models are a class of machine learning models that learn what to do in order to maximize reward, roughly inspired by operant conditioning in cognitive psychology. As such, the learner is not told explicitly what to do—as is the case in supervised learning—but has to discover which actions produce the highest reward through trial-and-error. In traditional reinforcement learning models, each possible action

that can be taken in a given state has a certain *value*: the immediate reward the action will yield plus the total amount of reward that can be expected in the future. In order to keep track of these values, they are often stored in a table[2], mapping discrete actions in discrete states to Q-values.

The models as used in their current form were not able to approach the final scores of the best human participants. However, Q-learning performed better than SARSA, and Q(λ) produced even better results. The relatively bad performance of Q-learning—which was quite surprising given the relative simplicity of the task—could be due to the specific action selection policy used. This is further explained in Section 7.2.2.

In the study described in Chapter 4, we found centering behavior to be a function of uncertainty, and a large variance in scores attained on the reinforcement learning task. To further examine these phenomena, the study described in Chapter 5 used a larger sample and a within-subject design. We wanted to investigate the factors that predict successful plan formation, and compare performance between the responsive SRT task and the exploratory reinforcement learning task. Participants in an SRT task can rely on two modes of executive control: stimulus-based control and plan-based control [160]. Under stimulus-based control, participants are prepared to respond to stimuli in an automatized fashion, delegating control to the external stimulus. Under plan-based control, an internal representation of the motor plan is made. These two modes of executive control can be strategically chosen under some circumstances. In a reinforcement learning paradigm, stimulus-based control is not a viable strategy, as there are no external stimuli to respond to. Participants were asked to perform both tasks described in Chapter 4 in randomized order, as well as complete measures of IQ, visuospatial working memory, need for structure, and locus of control.

For the SRT task, we used three measures of plan-based control: (1) the

[2]The action–value function need not necessarily be represented as a table. In fact, much progress has been made in the last years using (deep) neural networks as action-value function approximators, see e.g. [98].

acquisition of explicit knowledge about the sequence, (2) predictive movement toward the correct target in the inter-stimulus interval, and (3) the magnitude of frequency effects. Participants who acquired explicit sequence knowledge made increasingly larger predictive movements over the course of the task, whereas participants without explicit sequence knowledge hardly did so. Of all predictors, only visuospatial working memory predicted the acquisition of explicit sequence knowledge.

For the reinforcement learning task, both visuospatial working memory and IQ predicted final score. This suggests that the formation of an action plan in the current paradigm is limited by cognitive capacity, although another explanation could be that people with high IQ or WM are more likely to *actively look* for structure in sequential tasks.

In Chapter 6, we investigated the centering behavior described in Chapters 4 and 5 in more detail. We used a simulated robotic arm controlled by an artificial neural network to perform the same task as the one described in the earlier chapters: moving the mouse toward a stimuli that appear in a deterministic, repeating order. In one condition, the networks were provided with accurate information about the next stimulus, similar to human participants that have learned the sequence and are able to predict the next one. In another condition, the networks were given a random stimulus location as a prediction, making the prediction uninformative in that it contains no useful information about the next stimulus. In a third condition, we did not provide any stimulus location as a prediction, i.e. the input to the prediction units were fixed at zero.

We found that the networks that were given accurate predictions evolved predictive behavior. They moved toward the next stimulus after touching the current one, but before the next one appeared. The networks with either random or no prediction developed a centering strategy similar to the one described in Chapters 4 and 5: they moved the cursor to the center of the screen, an optimal location to wait for the next stimulus to appear.

7.2 Discussion and future directions

7.2.1 Sequential action under stimulus-based and plan-based control

Two modes of executive control were discussed and studied in this dissertation: stimulus-based and plan-based control. Our paradigm was a hybrid between our earlier trajectory SRT work and Tubau et al.'s [160] study into stimulus-based vs. plan-based control. In our design, we used a sequence with straight (left–right or up–down) movements being more frequent than diagonal movements in order to examine frequency effects, which were found by Tubau et al. [160] to decrease under plan-based control. However, due to the increased dimensionality of our paradigm there are many more possible frequency effects in play: horizontal repeat or switch, vertical repeat or switch, diagonal or straight, stimulus location, etc. This reduced the usability of frequency effects as a measure of plan-based control.

Other shortcomings with the used paradigm can be identified. Although the use of the original SRT sequence allows for a straightforward comparison with earlier work (e.g. [107]), this sequence is not specifically designed for the analyses conducted in our work. For example, with four alternatives the distances between alternatives are not identical, as diagonal movements require longer distances than straight movements. Although an analysis of response times between diagonal and straight movements in the random condition did not show an effect of movement type on response times, other properties of these movements could affect our results. For example, Burk et al. [23] found that movement distance affects decision making, and this could have made diagonal movements a less attractive choice for participants because they require more effort to perform. Additionally, location and transition probabilities are not balanced in the standard SRT sequence. A similar, balanced three-alternative paradigm could be used in future research to remove these confounds.

Another interesting avenue of research would be the role of stimulus probability on centering behavior. If the centering behavior described in

this dissertation is indeed due to minimization of mean travel distance to stimuli, altering stimulus probabilities would cause the centering location to shift toward more probable stimuli. This can be investigated both by using human participants as subjects, or in a simulated robotics paradigm such as the one described in Chapter 6.

7.2.2 Reinforcement learning: action selection and parameter fitting

The studies described in this dissertation compared human performance on a sequential reinforcement learning task with the performance of three reinforcement learning models: Q-learning, SARSA, and Q(λ). For a reinforcement learning model to perform well, the method of action selection it uses needs to balance between *exploitation*, using the information it has gathered from experience and that is stored in its Q-table, and *exploration*, allowing the model to try other and possibly better actions. At the start of any task or learning process, the Q-table may have been initialized to zero, or filled with small, random values. Either way, the information it contains is uninformative, and therefore should not be used for action selection. Different action selection policies deal differently with this problem. Several different action selection policies are used in the literature:

- **greedy**: the agent always selects the action that maximizes the value estimate;
- **random**: the agent always selects an action at random;
- ϵ-**greedy**: the agent selects the action that maximizes the value estimate Q with probability $1 - \epsilon$, otherwise it selects an action at random;
- **softmax**: the agent selects an action based on weighted probabilities by applying a softmax function over the value estimates. A temperature parameter τ can be used to control the spread of the softmax distribution.

The greedy policy could be considered purely exploitative, while the random policy is purely explorative. It should be clear that neither policy will provide good results in the paradigms described in this dissertation, as the greedy policy will always choose the action that happens to have the associated highest random value at Q-table initialization, while the random policy will never use the information stored in the Q-table. In the study described in Chapter 4, an ϵ-greedy policy was used. However, preliminary analyses of the data (not described in this dissertation) show that both softmax and another policy have the potential of outperforming even humans. The policy involves temporal decay of random action rate ϵ in the ϵ-greedy policy. ϵ is initialized to a relatively high value at the start of the sequence, exploring all possible actions and updating the Q-table with associated rewards. As the Q-values stabilize over the course of the experiment, ϵ decreases, making use of the informative Q-values that now populate the Q-table.

Also, the learning rule and action selection policy interact, as is clear from their definitions. The update rule in Q-learning updates Q for any state-action pair <s, a> using an experience tuple <s, a, s', r>, with learning rate $\alpha \in [0, 1]$ and discount factor $\gamma \in [0, 1]$:

$$Q'(s,a) = (1-\alpha)Q(s,a) + \alpha(r + \gamma Q[s', \underset{a'}{\operatorname{argmax}}(Q[s', a'])]) \tag{7.1}$$

SARSA, on the other hand, does not use the maximum attainable reward in state s' to update the Q-table, but instead chooses a' using the same policy it used to choose a. It therefore uses the experience tuple <s, a, r, s', a'>:

$$Q'(s,a) = (1-\alpha)Q(s,a) + \alpha(r + \gamma Q[s', a']) \tag{7.2}$$

Under a greedy action selection policy, Q-learning and SARSA are equivalent[3], and will update Q with the maximum attainable reward in state s': Q-learning by definition, and SARSA by virtue of always selecting

[3]Although note that Q-learning first updates Q, and selects the next action based on the updated Q, while SARSA chooses the action first and then updates Q.

the action that will yield the maximum attainable reward. Future studies should investigate the influence of action selection policies and their parameters on model performance in the paradigms discussed in this dissertation.

Also, if these reinforcement learning models are shown to be able to outperform humans in the task described in Chapters 4 and 5, parameter fitting could shed light on the nature of individual differences between human participants if the models turn out to be identifiable using specific cost metrics. For example, a final score of only 200 points could be due to either a low value of learning rate α, placing too little weight on the latest reward, or a too high value of random action rate ϵ, taking too many exploratory actions instead of exploiting the information in the Q-table. Instead, by looking at the learning trajectory, and using it as an error function, it could be possible to make these models identifiable. As another interesting manipulation, the reward schedule of the reinforcement learning task could be manipulated. By making certain rewards contingent on (a series of) earlier actions, differences in discount rate γ could be investigated, making this paradigm quite versatile for explaining individual differences.

7.3 Conclusion

This dissertation concerned itself with everyday action, and the mechanisms by which humans and robots are able to perform it. First, we described the fundamentals of everyday action, and explained that it is not as simple as the word implies. Also, we described the capacities a robot should have in order to perform everyday action. Next, we investigated the similarities and differences between human and robotic action control. Several mechanisms by which motor control is learned (e.g. motor babbling and reinforcement learning) are already common to both human and robotic action control.

The adaptation of the SRT task into a trajectory paradigm allows for the observation of predictive processes in sequential action control, and

shows that participants tend to adopt either a predictive or reactive strategy. Our results suggest that the quality of the action plan that is formed is a function of individual limitations in IQ and visuospatial working memory. The reinforcement learning models investigated did not perform as well as humans, but we suspect that the specific action selection policy used was partly to blame.

Participants who did not generate a reliable action plan tended to engage in centering behavior: moving the cursor to the center of the screen in anticipation of the next stimulus. We showed, using an evolutionary robotics approach, that this behavior evolves in an artificial neural network that controls a robot arm as a function of prediction quality. This suggests that this behavior is an emerging strategy caused by task constraints. The optimality of this behavior should be investigated further by manipulating target frequency and location.

Overall, the paradigms presented in this dissertation are well-suited to investigate both symbolic sequential action in the form of reinforcement learning, as well as sensorimotor action control in the form of evolved motor behavior in a robot arm controlled by an artificial neural network. Both paradigms provide ample opportunity for manipulation to further investigate the commonalities between complex human and robot action control.

Acknowledgments

The authors would like to thank Denis O'Hora and an anonymous reviewer for helpful comments and feedback on the study described in Chapter 4.

References

[1] J. A. Adams. A closed-loop theory of motor learning. *Journal of Motor Behavior*, 3:111–149, 1971. (p. 42).

[2] E. E. Aksoy, B. Dellen, M. Tamosiunaite, and F. Worgotter. Execution of a dual-object (pushing) action with semantic event chains. In *IEEE-RAS International Conference on Humanoid Robots*, 2011. (p. 51).

[3] J. R. Anderson. *Rules of the mind*. Erlbaum, Hillsdale, NJ, 1993. (p. 30).

[4] J. B. André and S. Nolfi. Evolutionary robotics simulations help explain why reciprocity is rare in nature. *Scientific Reports*, 6:32785, 2016. (p. 106).

[5] P. J. Angeline, G. M. Saunders, and J. B. Pollack. An evolutionary algorithm that constructs recurrent neural networks. *IEEE Transactions on Neural Networks*, 5:54–65, 1994. (p. 105).

[6] R. C. Arkin. Motor schema-based mobile robot navigation. *International Journal of Robotics Research*, 8:92–112, 1989. (p. 44).

[7] R. C. Arkin. *Behavior-based robotics*. MIT Press, Cambridge, MA, 1998. (p. 44).

[8] W. R. Ashby. *An introduction to cybernetics*. Chapman & Hall, London, 1956. (p. 40).

[9] C. G. Atkeson, J. G. Hale, F. Pollick, M. Riley, S. Kotosaka, S. Schaul, T. Shibata, G. Tevatia, A. Ude, S. Vijayakumar, E. Kawato, and M. Kawato. Using humanoid robots to study human behavior. *IEEE Intelligent Systems and their Applications*, 15(4):46–56, 2000. (p. 103).

[10] D. Badre. Cognitive control, hierarchy, and the rostro-caudal organization of the frontal lobes. *Trends in Cognitive Sciences*, 12:193–200, 2008. (p. 27).

[11] L. W. Barsalou. Flexibility, structure, and linguistic vagary in concepts: Manifestations of a compositional system of perceptual symbols. In A. C. Collins, S. E. Gathercole, and M. A. Conway, editors, *Theories of memory*, pages 29–101. Lawrence Erlbaum Associates, London, 1993. (p. 22).

[12] L. W. Barsalou. Perceptual symbol systems. *Behavioral and Brain Sciences*, 22:577–660, 1999. (p. 22).

[13] M. Beetz, L. Mösenlechner, and M. Tenorth. CRAM: A cognitive robot abstract machine for everyday manipulation in human environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010. (p. 33).

[14] F. Bell-Berti and K. S. Harris. Anticipatory coarticulation: Some implications from a study of lip rounding. *Journal of the Acoustical Society of America*, 65(5):1268–1270, 1979. (pp. 20, 116).

[15] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987. (p. 51).

[16] J. Bo and R. D. Seidler. Visuospatial working memory capacity predicts the organization of acquired explicit motor sequences. *Journal of Neurophysiology*, 101:3116–3125, 2009. (p. 89).

[17] J. Bo, S. Jennett, and R. D. Seidler. Working memory capacity correlates with implicit serial reaction time task performance. *Experimental Brain Research*, 214:73–81, 2011. (pp. 89, 91, 100).

[18] M. Botvinick and D. C. Plaut. Doing without schema hierarchies: A recurrent connectionist approach to routine sequential action and its pathologies. *Psychological Review*, 111:395–429, 2004. (pp. 29, 30, 31, 33, 52, 57, 58, 87, 117).

[19] M. Boyer, A. Destrebecqz, and A. Cleeremans. Processing abstract sequence structure: Learning without knowing, or knowing without learning? *Psychological Research*, 69:383–398, 2005. (pp. 57, 75, 78).

[20] V. Braitenberg. *Vehicles: Experiments in synthetic psychology*. MIT Press, Cambridge, MA, 1984. (pp. 41, 118).

[21] R. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2:14–23, 1986. (pp. 41, 44, 118).

[22] R. Brooks. Intelligence without reason. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, volume 1, pages 569–595, 1991. (pp. 49, 118).

[23] D. Burk, J. N. Ingram, D. W. Franklin, M. N. Shadlen, and D. M. Wolpert. Motor effort alters changes of mind in sensorimotor decision making. *PLoS ONE*, 9:1–10, 2014. (p. 122).

[24] H. R. Burke and W. C. Bingham. Raven's progressive matrices: More on construct validity. *Journal of Psychology*, 72:247–251, 1969. (p. 90).

[25] D. Caligiore, D. Parisi, N. Accornero, M. Capozza, and G. Baldassarre. Using motor babbling and Hebb rules for modeling the development of reaching with obstacles and grasping. In *Proceedings of the International Conference on Cognitive Systems*, 2008. (p. 49).

[26] M. Campbell, A. J. Hoane Jr., and F. Hsu. Deep Blue. *Artificial Intelligence*, 134:57–83, 2002. (p. 1).

[27] N. Chomsky. A review of B. F. Skinner's *Verbal Behavior*. *Language*, 35: 26–58, 1959. (p. 39).

[28] A. Clark. *Being there: Putting brain, body and world together again*. MIT Press, Cambridge, MA, 1997. (p. 18).

[29] A. Cleeremans and J. L. McClelland. Learning the structure of event sequences. *Journal of Experimental Psychology: General*, 120:235–253, 1991. (pp. 57, 58).

[30] R. G. Cohen and D. A. Rosenbaum. Where grasps are made reveals how grasps are planned: Generation and recall of motor plans. *Experimental Brain Research*, 157:486–495, 2004. (pp. 20, 104).

[31] RoboHow Consortium. RoboHow project website. URL `www.robohow.eu`. (pp. 3, 17).

[32] R. Cooper and T. Shallice. Contention scheduling and the control contention scheduling and the control of routine activities. *Cognitive Neuropsychology*, 17:297–338, 2000. (pp. 57, 87).

[33] R. Cooper and T. Shallice. Hierarchical schemas and goals in the control of sequential behavior. *Psychological Review*, 113:887–916, 2006. (pp. 30, 31, 52, 117).

[34] R. Dale, N. D. Duran, and J. R. Morehead. Prediction during statistical learning, and implications for the implicit/explicit divide. *Advances in Cognitive Psychology*, 8:196–209, 2012. (pp. 7, 8, 11, 12, 14, 65, 66, 69, 80, 95, 99, 104, 105, 112, 119).

[35] R. Daniloff and K. Moll. Coarticulation of lip rounding. *Journal of Speech and Hearing Research*, 11:707–721, 1968. (pp. 20, 104).

[36] Y. Demiris and A. Dearden. From motor babbling to hierarchical learning by imitation: A robot developmental pathway. In *Proceedings of the Fifth International Workshop on Epigenetic Robotics*, pages 31–37, 2005. (p. 49).

[37] P. F. Dominey. Influences of temporal organization on sequence learning and transfer: Comments on Stadler (1995) and Curran and Keele (1993). *Journal of Experimental Psychology: Learning, Memory, and Cognition Learning Memory and Cognition*, 24:234–248, 1998. (p. 78).

[38] N. D. Duran and R. Dale. Predictive arm placement in the statistical learning of position sequences. In *Proceedings of the 31st Annual Meeting of the Cognitive Science Society*, pages 893–898, Amsterdam, 2009. Cognitive Science Society. (pp. 14, 66, 104, 112, 119).

[39] R. M. Eenshuistra, M. A. Weidema, and B. Hommel. Development of the acquisition and control of action–effect associations. *Acta Psychologica*, 115:185–209, 2004. (p. 48).

[40] D. Elliott, S. Hansen, L. E. Grierson, J. Lyons, S. J. Bennett, and S. J. Hayes. Goal-directed aiming: Two components but multiple processes. *Psychological Bulletin*, 136:1023–1044, 2010. (p. 27).

[41] J. L. Elman. Finding structure in time. *Cognitive Science*, 14:179–211, 1990. (pp. 57, 87).

[42] B. Elsner and B. Hommel. Effect anticipation and action control. *Journal of Experimental Psychology: Human Perception and Performance*, 27:229–240, 2001. (p. 47).

[43] B. Elsner and B. Hommel. Contiguity and contingency in the acquisition of action effects. *Psychological Research*, 68:138–154, 2004. (p. 47).

[44] C. W. Eriksen, M. D. Pollack, and W. E. Montague. Implicit speech: Mechanism in perceptual encoding? *Journal of Experimental Psychology*, 84:502–507, 1970. (pp. 23, 116).

[45] S. Fagioli, B. Hommel, and R. I. Schubotz. Intentional control of attention: Action planning primes action-related stimulus dimensions. *Psychological Research*, 71:22–29, 2007. (p. 33).

[46] P. Fendrick. Hierarchical skills in typewriting. *Journal of Educational Psychology*, 28:609–620, 1937. (p. 57).

[47] R. Fikes and N. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208, 1971. (pp. 19, 40, 115).

[48] M. H. Fischer and M. Hartmann. Pushing forward in embodied cognition: may we mouse the mathematical mind? *Frontiers in Psychology*, 5: 1315, 2014. (pp. 7, 119).

[49] C. A. Fowler. Coarticulation and theories of extrinsic timing control. *Journal of Phonetics*, 8:113–133, 1980. (p. 20).

[50] Q. Fu, X. Fu, and Z. Dienes. Implicit sequence learning and conscious awareness. *Consciousness and Cognition*, 17:185–202, 2008. (p. 58).

[51] D. R. Gentner, J. Grudin, and E. Conway. *Skilled finger movements in typing (technical report 8001)*. Center for Human Information Processing, San Diego, CA, 1980. (pp. 21, 116).

[52] D. R. Gentner, S. LaRochelle, and J. Grudin. Lexical, sublexical, and peripheral effects in skilled typewriting. *Cognitive Psychology*, 20:524–548, 1988. (p. 57).

[53] S. Glover. Separate visual representations in the planning and control of action. *Behavioral and Brain Sciences*, 27:3–24, 2004. (pp. 25, 42, 117).

[54] M. A. Goodale and A. D. Milner. Separate visual pathways for perception and action. *Trends in Neurosciences*, 15:20–25, 1992. (p. 42).

[55] M. A. Goodale, D. Pelisson, and C. Prablanc. Large adjustments in visually guided reaching do not depend on vision of the hand or perception of target displacement. *Nature*, 320:748–750, 1986. (pp. 25, 42).

[56] M. Hägele, K. Nilsson, and J. Norberto Pires. Industrial robotics. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, pages 963–986. Springer, Berlin, 2008. (p. 23).

[57] E. Harless. Der Apparat des Willens. *Zeitschrift für Philosophie und philosophische Kritik*, 38:50–73, 1861. (p. 47).

[58] J. A. Hartigan and P. M. Hartigan. The dip test of unimodality. *The Annals of Statistics*, 13:70–84, 1985. (p. 96).

[59] F. M. Henry and D. E. Rogers. Increased response latency for complicated movements and a "memory drum" theory of neuromotor reaction. *Research Quarterly*, 31:448–458, 1960. (pp. 6, 20, 23, 25, 42, 116).

[60] J. Hoffmann and I. Koch. Stimulus–response compatibility and sequential learning in the serial response time task. *Psychological Research*, 60:87–97, 1997. (pp. 11, 86).

[61] B. Hommel. The prepared reflex: Automaticity and control in stimulus–response translation. In S. Monsell and J. Driver, editors, *Control of cognitive processes: Attention and performance XVIII*, pages 247–273. MIT Press, Cambridge, MA, 2000. (pp. 7, 11, 86).

[62] B. Hommel. Planning and representing intentional action. *TheScientificWorldJOURNAL*, 3:593–608, 2003. (p. 86).

[63] B. Hommel. Action control according to TEC (theory of event coding). *Psychological Research*, 73:512–526, 2009. (pp. 18, 47).

[64] B. Hommel. Grounding attention in action control: The intentional control of selection. In B. J. Bruya, editor, *Effortless attention: A new perspective in the cognitive science of attention and action*, pages 121–140. Cambridge, MA: MIT Press, 2010. (pp. 25, 42, 49, 117).

[65] B. Hommel, J. Müsseler, G. Aschersleben, and W. Prinz. The Theory of Event Coding (TEC): A framework for perception and action planning. *Behavioral and Brain Sciences*, 24:849–937, 2001. (pp. 18, 32, 47, 83, 117).

[66] G. Houghton. The problem of serial order: A neural network model of sequence learning and recall. In R. Dale, C. Mellish, and M. Zock, editors, *Current research in natural language generation*, pages 287–319. Academic Press, San Diego, CA, 1990. (p. 29).

[67] M. Ito, K. Noda, Y. Hoshino, and J. Tani. Dynamic and interactive generation of object handling behaviors by a small humanoid robot using a dynamic neural network model. *Neural Networks*, 19:323–337, 2006. (p. 103).

[68] R. Jain and T. Inamura. Bayesian learning of tool affordances based on generalization of functional feature to estimate effects of unseen tools. *Artificial Life and Robotics*, 18:95–103, 2013. (p. 51).

[69] W. James. *Principles of psychology*, volume 1. Holt, New York, 1890. (pp. 6, 19, 47, 85, 116).

[70] R. K. Jamieson and D. J. K. Mewhort. Applying an exemplar model to the serial reaction-time task: Anticipating from experience. *Quarterly Journal of Experimental Psychology*, 62:1757–1783, 2009. (p. 104).

[71] M. Jeannerod, M. A. Arbib, G. Rizzolatti, and H. Sakata. Grasping objects: The cortical mechanisms of visuomotor transformation. *Trends in Neurosciences*, 18:314–320, 1995. (p. 20).

[72] M. N. Jones and D. J. K. Mewhort. Representing word meaning and order information in a composite holographic lexicon. *Psychological Review*, 114: 1–37, 2007. (p. 45).

[73] P. Joshi and W. Maass. Movement generation with circuits of spiking neurons. *Neural Computation*, 17:1715–1738, 2005. (p. 35).

[74] G. Kachergis, F. Berends, R. de Kleijn, and B. Hommel. Reward effects on sequential action learning in a trajectory serial reaction time task. In *IEEE International Conference on Development and Learning and on Epigenetic Robotics*, 2014. (pp. 59, 91).

[75] G. Kachergis, F. Berends, R. de Kleijn, and B. Hommel. Trajectory effects in a novel serial reaction time task. In *Proceedings of the 36th Annual Conference of the Cognitive Science Society*, pages 713–718, Québec, QC, 2014. (pp. 7, 8, 21, 46, 59, 91, 104).

[76] G. Kachergis, D. Wyatte, R. C. O'Reilly, R. de Kleijn, and B. Hommel. A continuous time neural model for sequential action. *Philosophical Transactions of the Royal Society B*, 369:20130623, 2014. (pp. 5, 30, 52, 58).

[77] G. Kachergis, F. Berends, R. de Kleijn, and B. Hommel. Human reinforcement learning of sequential action. In *Proceedings of the 38th Annual Conference of the Cognitive Science Society*, pages 193–198, Philadelphia, PA, 2016. (pp. 88, 92, 99, 105, 106).

[78] M. A. Khan, I. M. Franks, and D. Goodman. The effect of practice on the control of rapid aiming movements: Evidence for an interdependency between programming and feedback processing. *Quarterly Journal of Experimental Psychology: Human Experimental Psychology*, 51:425–444, 1998. (p. 27).

[79] S. T. Klapp. Reaction time analysis of programmed control. *Exercise and Sport Sciences Reviews*, 5:231–253, 1977. (pp. 6, 20).

[80] R. Klein. Nonhierarchical control of rapid movement sequences. *Journal of Experimental Psychology: Human Perception and Performance*, 9:834–836, 1983. (p. 28).

[81] H. D. Knapp, E. Taub, and A. J. Berman. Movements in monkeys with deafferented forelimbs. *Experimental Neurology*, 7:305–315, 1963. (p. 42).

[82] D. Kraft, E. Baseski, M. Popovic, A. M. Batog, A. Kjær-Nielsen, N. Krüger, R. Petrick, C. Geib, N. Pugeault, M. Steedman, T. Asfour, R. Dillmann, S. Kalkan, F. Wörgötter, B. Hommel, R. Detry, and J. Piater. Exploration and planning in a three-level cognitive architecture. In *Proceedings of the International Conference on Cognitive Systems*, Karlsruhe, 2008. (pp. 35, 48).

[83] S. Kühn, A. Keizer, S. A. R. B. Rombouts, and B. Hommel. The functional and neural mechanism of action preparation: Roles of EBA and FFA in voluntary action control. *Journal of Cognitive Neuroscience*, 23:214–220, 2011. (p. 33).

[84] M. Kuperstein. Neural model of adaptive hand–eye coordination for single postures. *Science*, 239:1308–1311, 1988. (pp. 49, 103).

[85] J. E. Laird, A. Newell, and P. S. Rosenbloom. SOAR: An architecture for general intelligence. *Artificial Intelligence*, 33:1–64, 1987. (p. 30).

[86] S. Larochelle. Some aspects of movements in skilled typewriting. In H. Bouma and D. G. Bouwhuis, editors, *Attention and performance. Control of language processes*, volume 10, pages 43–54. Erlbaum, Hillsdale, NJ, 1984. (p. 21).

[87] K. S. Lashley. The problem of serial order in behavior. In L. A. Jeffress, editor, *Cerebral mechanisms in behavior*, pages 112–136. Wiley, New York, 1951. (pp. 6, 20, 21, 27, 30, 42, 46, 58, 88).

[88] H. Levenson. Differentiating among internality, powerful others, and chance. In H. M. Lefcourt, editor, *Research with the locus of control construct*, volume 1, pages 15–63. Academic Press, New York, 1981. (p. 90).

[89] P. Lewicki, T. Hill, and E. Bizot. Acquisition of procedural knowledge about a pattern of stimuli that cannot be articulated. *Cognitive Psychology*, 20:24–37, 1988. (p. 87).

[90] G. D. Logan and M. J. C. Crump. Response to M. Ullsperger and Danielmeier's E-Letter. *Science E-Letters*, February 9, 2011, 2011. (pp. 2, 23).

[91] R. H. Lotze. *Medizinische Psychologie oder Physiologie der Seele*. Weidmann, 1852. (p. 47).

[92] S. J. Luck and E. K. Vogel. The capacity of visual working memory for features and conjunctions. *Nature*, 390:279–281, 1997. (p. 91).

[93] Z. Macura, A. Cangelosi, R. Ellis, D. Bugmann, M. H. Fischer, and A. Myachykov. A cognitive robotic model of grasping. In *Proceedings of the Ninth International Conference on Epigenetic Robotics*, Venice, 2009. (p. 32).

[94] M. Maniadakis and P. Trahanias. Temporal cognition: A key ingredient of intelligent systems. *Frontiers in Neurorobotics*, 5:2, 2011. (p. 33).

[95] J. Memelink and B. Hommel. Intentional weighting: A basic principle in cognitive control. *Psychological Research*, 77:249–259, 2013. (pp. 33, 117).

[96] G. A. Miller, E. Galanter, and K. H. Pribram. *Plans and the structure of behavior*. Holt, Rinehart & Winston, New York, 1960. (pp. 27, 30, 31, 86).

[97] A. D. Milner and M. A. Goodale. *The visual brain in action*. Oxford University Press, Oxford, 1995. (p. 26).

[98] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015. (p. 120).

[99] D. E. Moriarty. *Symbiotic evolution of neural networks in sequential decision tasks*. PhD thesis, University of Texas at Austin, 1997. (p. 105).

[100] D. E. Moriarty and R. Miikkulainen. Efficient reinforcement learning through symbiotic evolution. *Machine Learning*, 22:11–32, 1996. (p. 105).

[101] G. Morlino, C. Giannelli, A. Borghi, and S. Nolfi. Category learning through action: A study with human and artificial agents. *Cognitive Processing*, 13:47–48, 2012. (p. 106).

[102] H. Münsterberg. *Beiträge zur experimentellen Psychologie [Contributions to experimental psychology]*, volume 4. Mohr, Freiburg, 1892. (pp. 6, 85).

[103] C. Nakatani and J. Hirschberg. A corpus-based study of repair cues in spontaneous speech. *Journal of the Acoustical Society of America*, 95:1603–1616, 1994. (pp. 21, 46).

[104] D. Nattkemper and W. Prinz. Stimulus and response anticipation in a serial reaction task. *Psychological Research*, 60:98–112, 1997. (p. 86).

[105] S.L. Neuberg and J.T. Newsom. Personal need for structure: Individual differences in the desire for simple structure. *Journal of Personality and Social Psychology*, 65:113–131, 1993. (pp. 89, 90).

[106] S. B. Niku. *Introduction to robotics: Analysis, control, applications*. Wiley, 2010. (p. 44).

[107] M. J. Nissen and P. Bullemer. Attentional requirements of learning: evidence from performance measures. *Cognitive Psychology*, 19:1–32, 1987. (pp. 6, 7, 11, 12, 46, 57, 58, 59, 60, 61, 62, 63, 65, 68, 69, 70, 76, 80, 81, 87, 91, 118, 119, 122, 145).

[108] S. Nolfi and D. Parisi. Evolution of artificial neural networks. In M. A. Arbib, editor, *Handbook of brain theory and neural networks*, pages 418–421. MIT Press, Cambridge, MA, 2002. (p. 108).

[109] S. Nolfi, D. Parisi, and J. L. Elman. Learning and evolution in neural networks. *Adaptive Behavior*, 3:5–28, 1994. (p. 108).

[110] J. K. O'Regan and A. Noe. A sensorimotor account of vision and visual consciousness. *Behavioral and Brain Sciences*, 24:939–1031, 2001. (p. 18).

[111] R. C. O'Reilly. *The LEABRA model of neural interactions and learning in the neocortex*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1996. (pp. 5, 30).

[112] R. C. O'Reilly, T. E. Hazy, and S. A. Herd. The LEABRA cognitive architecture: How to play 20 questions with nature and win! In S. E. F. Chipman, editor, *Oxford Handbook of Cognitive Science*. Oxford University Press, Oxford, in press. (p. 5).

[113] C. Palmer and P. Q. Pfordresher. Incremental planning in sequence production. *Psychological Review*, 110:683–712, 2003. (p. 24).

[114] G. Petrosino, D. Parisi, and S. Nolfi. Selective attention enables action selection: evidence from evolutionary robotics experiments. *Adaptive Behavior*, 21:356–370, 2013. (p. 106).

[115] G. Pezzulo, L. W. Barsalou, A. Cangelosi, M. H. Fischer, K. McRae, and M. J. Spivey. The mechanics of embodiment: A dialog on embodiment and computational modeling. *Frontiers in Psychology*, 2:5, 2011. (p. 32).

[116] P. J. Phillips and A. J. O'Toole. Comparison of human and computer performance across face recognition experiments. *Image and Vision Computing*, 32:74–85, 2014. (p. 1).

[117] T. A. Plate. Holographic reduced representations. *IEEE Transactions on Neural Networks*, 6:623–641, 1995. (p. 6).

[118] M. Plooij, M. de Vries, W. Wolfslag, and M. Wisse. Optimization of feedforward controllers to minimize sensitivity to model inaccuracies. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013. (p. 27).

[119] D. Powell. How computers were finally able to best poker pros. Washington Post, February 4th, 2017. (p. 1).

[120] C. Prablanc and O. Martin. Automatic control during hand reaching at undetected two-dimensional target displacements. *Journal of Neurophysiology*, 67:455–469, 1992. (p. 25).

[121] J. Pratt and R. A. Abrams. Practice and component submovements: The roles of programming and feedback in rapid aimed limb movements. *Journal of Motor Behavior*, 28:149–156, 1996. (p. 26).

[122] J. J. Prinz and L. W. Barsalou. Steering a course for embodied representation. In E. Dietrich and A. Markman, editors, *Cognitive dynamics: Conceptual change in humans and machines*, pages 51–77. MIT Press, Cambridge, MA, 2000. (p. 32).

[123] L. Proteau, R. G. Marteniuk, Y. Girouard, and C. Dugas. On the type of information used to control and learn an aiming movement after moderate and extensive practice. *Human Movement Science*, 6:181–199, 1987. (pp. 26, 27).

[124] J. Raven, J. C. Raven, and J. H. Court. *Manual for Raven's Progressive Matrices and Vocabulary Scales. Section 4: The Advanced Progressive Matrices*. Harcourt Assessment, San Antonio, TX, 1998. (p. 89).

[125] D. A. Rosenbaum. Successive approximations to a model of human motor programming. *Psychology of Learning and Motivation*, 21:153–182, 1987. (pp. 6, 20).

[126] D. A. Rosenbaum. *Human motor control*. Academic Press, New York, 1991. (p. 20).

[127] D. A. Rosenbaum. The Cinderella of psychology: The neglect of motor control in the science of mental life and behavior. *American Psychologist*, 60:308–317, 2005. (p. 18).

[128] D. A. Rosenbaum, R. J. Weber, W. M. Hazelett, and V. Hindorff. The parameter remapping effect in human performance: Evidence from tongue twisters and finger fumblers. *Journal of Memory and Language*, 25:710–725, 1986. (p. 33).

[129] D. A. Rosenbaum, R. G. Cohen, S. A. Jax, R. van der Wel, and D. J. Weiss. The problem of serial order in behavior: Lashley's legacy. *Human Movement Science*, 26:525–554, 2007. (pp. 6, 20, 21, 46).

[130] D. E. Rumelhart and J. L. McClelland. *Parallel distributed processing: Explorations in the microstructure of cognition. Volume I.* MIT Press, Cambridge, MA, 1986. (p. 40).

[131] D. E. Rumelhart and D. A. Norman. Simulating a skilled typist: A study of skilled cognitive-motor performance. *Cognitive Science*, 6:1–36, 1982. (pp. 2, 22, 29, 116).

[132] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986. (p. 105).

[133] G. A. Rummery and M. Niranjan. On-line Q-Learning using connectionist systems. Technical Report CUED/F-INFENG/TR 166, Cambridge University, 1994. (p. 77).

[134] E. D. Sacerdoti. Planning in a hierarchy of abstraction spaces. *Artificial Intelligence*, 5:115–135, 1974. (p. 45).

[135] R. Saegusa, G. Metta, G. Sandini, and S. Sakka. Active motor babbling for sensory-motor learning. In *IEEE International Conference on Robotics and Biomimetics*, pages 794–799, 2008. (p. 49).

[136] J. Saffran, E. Newport, and R. Aslin. Word segmentation: The role of distributional cues. *Journal of Memory and Language*, 1996. (pp. 57, 87).

[137] I. Saunders and S. Vijayakumar. The role of feed-forward and feedback processes for closed-loop prosthesis control. *Journal of NeuroEngineering and Rehabilitation*, 8:60, 2011. (p. 24).

[138] S. Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3:233–242, 1999. (p. 104).

[139] T. Schaul, J. Bayer, D. Wierstra, Y. Sun, M. Felder, F. Sehnke, T. Rückstieß, and J. Schmidhuber. PyBrain. *Journal of Machine Learning Research*, 11: 743–746, 2010. (p. 75).

[140] R. A. Schmidt. A schema theory of discrete motor skill learning. *Psychological Review*, 82:225–260, 1975. (pp. 25, 42).

[141] R. D. Seidler, D. C. Noll, and G. Thiers. Feedforward and feedback processes in motor control. *NeuroImage*, 22:1775–1783, 2004. (pp. 25, 42).

[142] Y. K. Shin, R. W. Proctor, and E. J. Capaldi. A review of contemporary ideomotor theory. *Psychological Bulletin*, 136:943–974, 2010. (p. 47).

[143] B. F. Skinner. *The behavior of organisms: An experimental analysis*. B. F. Skinner Foundation, Cambridge, MA, 1938. (p. 38).

[144] B. F. Skinner. Are theories of learning necessary? *Psychological Review*, 57: 193–216, 1950. (p. 59).

[145] J. F. Soechting, A. M. Gordon, and K. C. Engel. *Sequential hand and finger movements: Typing and piano playing*. MIT Press, 1996. (p. 80).

[146] J.-H. Song and K. Nakayama. Hidden cognitive states revealed in choice reaching tasks. *Trends in Cognitive Sciences*, 13:360–366, 2009. (pp. 7, 12, 119).

[147] M. J. Spivey and R. Dale. Continuous dynamics in real-time cognition. *Current Directions in Psychological Science*, 15:207–211, 2006. (p. 59).

[148] M. J. Spivey, M. Grosjean, and G. Knoblich. Continuous attraction toward phonological competitors. *Proceedings of the National Academy of Sciences of the United States of America*, 102:10393–10398, 2005. (pp. 7, 119).

[149] M. A. Stadler. Statistical structure and implicit serial learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 18:318–327, 1992. (p. 57).

[150] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:99–127, 2002. (p. 105).

[151] S. Sternberg, R. L. Knoll, S. Monsell, and C. E. Wright. Motor programs and hierarchical organization in the control of rapid speech. *Phonetica*, 45: 175–197, 1988. (p. 24).

[152] A. Stoytchev. Autonomous learning of tool affordances by a robot. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI)*, 2005. (p. 50).

[153] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT Press, Cambridge, MA, 1998. (pp. 9, 59).

[154] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. arXiv:1312.6199 [cs.CV], 2014. (p. 52).

[155] E. Taub, I. A. Goldberg, and P. Taub. Deafferentation in monkeys: Pointing at a target without visual feedback. *Experimental Neurology*, 46:178–186, 1975. (p. 42).

[156] M. Tenorth and M. Beetz. Knowledge processing for autonomous robot control. In *AAAI Spring Symposium on Designing Intelligent Robots: Reintegrating AI*, 2012. (p. 33).

[157] M. Tenorth and M. Beetz. KnowRob: A knowledge processing infrastructure for cognition-enabled robots. Part 1: The KnowRob system. *International Journal of Robotics Research*, 32:566–590, 2013. (p. 51).

[158] M. M. Thompson, M. E. Naccarato, and K. Parker. Assessing cognitive need: The development of the Personal Need for Structure and Personal Fear of Invalidity scales. In *Proceedings of the Annual Meeting of the Canadian Psychological Association*, Halifax, Nova Scotia, 1989. (p. 90).

[159] E. Tubau and J. López-Moliner. Spatial interference and response control in sequence learning: the role of explicit knowledge. *Psychological Research*, 68:55–63, 2004. (p. 11).

[160] E. Tubau, B. Hommel, and J. López-Moliner. Modes of executive control in sequence learning: From stimulus-based to plan-based control. *Journal of Experimental Psychology: General*, 136:43–63, 2007. (pp. 6, 11, 59, 80, 85, 86, 87, 95, 99, 120, 122, 147).

[161] A. M. Turing. Computing machinery and intelligence. *Mind*, 59:433–460, 1950. (p. 39).

[162] S. Uithol, I. van Rooij, H. Bekkering, and W. F. G. Haselager. Hierarchies in action and motor control. *Journal of Cognitive Neuroscience*, 24, 2012. (p. 28).

[163] R. P. R. D. van der Wel and D. A. Rosenbaum. Coordination of locomotion and prehension. *Experimental Brain Research*, 176:281–287, 2007. (p. 20).

[164] S. A. Verschoor, M. Spapé, S. Biro, and B. Hommel. From outcome prediction to action selection: Developmental change in the role of action–effect bindings. *Developmental Science*, 16:801–814, 2013. (p. 48).

[165] S. A. Verschoor, M. Paulus, M. Spapé, S. Biro, and B. Hommel. The developing cognitive substrate of sequential action control in 9- to 12-month-olds: Evidence for concurrent activation models. *Cognition*, 138:64–78, 2015. (p. 22).

[166] W. G. Walter. *The living brain*. Duckworth, London, 1953. (p. 40).

[167] M. F. Washburn. *Movement and mental imagery*. Houghton Mifflin, Boston, MA, 1916. (pp. 6, 19, 116).

[168] C. J. C. H. Watkins. *Learning from delayed rewards*. PhD thesis, Cambridge University, 1989. (p. 77).

[169] J. B. Watson. Psychology as the behaviourist views it. *Psychological Review*, 20:158–177, 1913. (p. 38).

[170] D. B. Willingham, M. J. Nissen, and P. Bullemer. On the development of procedural knowledge. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 15:1047–1060, 1989. (pp. 58, 61, 65).

[171] R. S. Woodworth. The accuracy of voluntary movement. *Psychological Review*, 3:1–119, 1899. (p. 42).

[172] A. Wykowska, A. Schubö, and B. Hommel. How you move is what you see: Action planning biases selection in visual search. *Journal of Experimental Psychology: Human Perception and Performance*, 35:1755–1769, 2009. (p. 33).

[173] Y. Yamashita and J. Tani. Emergence of functional hierarchy in a multiple timescale neural network model: A humanoid robot experiment. *PLOS Computational Biology*, 4(11): e1000220, 2008. (p. 103).

Summary in Dutch
Nederlandse samenvatting

Robots en kunstmatige intelligentie zijn de afgelopen jaren een steeds belangrijkere rol gaan spelen in zowel ons alledaagse leven (bijvoorbeeld zelfrijdende auto's) als in het bedrijfsleven. Maar naarmate de taken die robots moeten uitvoeren complexer worden, en uiteindelijk zelfs alledaagse, menselijke taken zouden moeten overnemen, wordt de aansturing van deze robots steeds lastiger.

Hoewel alledaagse taken zoals koken en stofzuigen op het eerste gezicht niet erg ingewikkeld klinken, blijkt het behoorlijk uitdagend om robots te ontwikkelen die deze taken succesvol kunnen uitvoeren. Het ontwikkelen van zo'n robot was het doel van het *RoboHow*-project. Het proefschrift dat voor u ligt is het resultaat van dit project, uitgevoerd door een internationaal consortium van robotici, computerwetenschappers en cognitief psychologen verbonden aan vijf universiteiten en twee onderzoeksinstituten verspreid over Europa.

De wisselwerking tussen cognitief-psychologisch onderzoek en kunstmatige intelligentie werd goed duidelijk tijdens de *neo-cognitieve revolutie*. Behaviorisme, het standpunt dat alleen meetbare en observeerbare gedragingen van mens en dier onderwerp zouden moeten zijn van onderzoek, bleek niet houdbaar vanwege de beperkte verklaringscapaciteit. Typisch "menselijke" fenomenen zoals taal en geheugen konden vanuit een puur behavioristisch framework niet onderzocht worden; zij vereisen een onobserveerbare mentale toestand. De neo-cognitieve revolutie opende de deur voor theorieën die deze mentale toestand—of cognitieve

processen—probeerden te beschrijven en te begrijpen. Maar hoe wordt die mentale inhoud nu precies gerepresenteerd?

Cognitie in mensen en robots

In de kunstmatige intelligentie wordt er in deze gevallen vaak gebruikt gemaakt van een *planner*. In planners worden subacties vaak symbolisch gerepresenteerd als losse eenheden waarop bewerkingen kunnen worden toegepast, en zo kan worden berekend welke subacties er in welke volgorde nodig zijn om vanuit een beginpositie een eindpositie te bereiken. Bij mensen werd lange tijd gedacht dat een soort actieketting verantwoordelijk was voor het uitvoeren van zulke sequentiële actie; de zintuiglijke waarneming van de effecten van een actie zouden functioneren als een *trigger* voor het uitvoeren van de volgende actie. Halverwege de twintigste eeuw werd echter duidelijk dat deze theorieën niet correct konden zijn, onder meer omdat bleek dat sequentiële actie ook kan worden uitgevoerd door mensen waarbij zintuiglijke terugkoppeling verstoord is. Naarmate het bewijs tegen deze symbolische theorieën zich opstapelde werd duidelijk dat het subsymbolische, sensorimotorische aspect van motoracties essentieel was voor sequentiële actieplanning.

Na het *plannen* van een actie moet deze ook *uitgevoerd* worden door het motorsysteem. Een robot die aan de lopende band werkt kan voorgeprogrammeerde acties uitvoeren met een *feedforward*-systeem, waarin informatie uit het programma (bijvoorbeeld "roteer motor 12 naar positie 82,5°") direct wordt omgezet in een motorbeweging. Hoewel dit een zeer snelle manier van aansturing is, kan dit echter voor problemen zorgen in minder voorspelbare omgevingen: wanneer het te manipuleren object zich in positie 83,5° bevindt zal de actie mogelijk mislukken. Een *feedback*-systeem gebruikt informatie uit de omgeving om de actie moduleren. Dit vergroot de kans op een succesvol uitgevoerde actie, maar afhankelijk van de snelheid van de feedback-loop zal de uitvoer minder snel zijn. Menselijk gedrag is het product van een hybride feedforward-feedback-systeem, waarbij een feedforward actieplan wordt gegenereerd

waarin onbekende parameters online kunnen worden ingevuld door een feedback-mechanisme.

Het leren van sequentiële actie

Een manier om deze actieplanning en -uitvoer te onderzoeken is de serial response time (SRT)-taak, geïntroduceerd door Nissen & Bullemer [107]. In deze taak zit de proefpersoon tegenover een beeldscherm waar aan de onderkant een visuele stimulus verschijnt op één van vier mogelijke posities. Wanneer een stimulus verschijnt, drukt de proefpersoon zo snel mogelijk op een knop die onder deze stimulus is gepositioneerd. De proefpersonen weten niet dat de stimuli verschijnen in een vaste, herhalende volgorde. Hoewel deze taak veelvuldig in de literatuur is gebruikt, heeft het als groot nadeel dat de informatie die verzameld wordt gelimiteerd is door de discrete vorm van de respons. Hierdoor is het niet mogelijk om informatie te verzamelen over processen die actief zijn tijdens het inter-trial interval (ITI), zoals voorspellende bewegingen.

In hoofdstukken 4 en 5 worden studies beschreven die een continue variant van de SRT-taak gebruiken, waarbij de vier stimuli en knoppen zijn omgezet in vier zwarte vierkanten op een beeldscherm. Analoog aan de originele SRT-taak worden proefpersonen gevraagd om zo snel mogelijk te reageren op een oplichtende stimulus (het target) door de muiscursor erheen te bewegen. In deze studie werden twee condities gebruikt: een deterministische conditie waarin de stimuli oplichtten in een vaste, herhalende reeks van 10 targets, en een conditie waarin de volgorde van targets willekeurig werd bepaald. In totaal werden 800 targets gepresenteerd. Deze variant van de SRT-taak produceerde dezelfde effecten als de originele taak, waarin proefpersonen sneller worden naarmate het experiment vorderde, maar dit effect was sterker voor de deterministische conditie. Dit duidt op het impliciet leren van de reeks in de deterministische groep, een conclusie eerder getrokken door Nissen & Bullemer [107]. Dankzij de continue aard van de adaptatie die in onze studie is gebruikt, kon duidelijk worden gemaakt dat deze versnelling *niet* kwam

door een simpele versnelling van de motoractie, maar (mede) werd veroorzaakt door voorspellende bewegingen richting de volgende stimulus tijdens het ITI.

Ook werd duidelijk dat proefpersonen twee strategieën kunnen gebruiken om zo snel mogelijk te reageren. Eén groep proefpersonen maakte actief een voorspelling van de volgende target, en bewoog de muiscursor al voordat de target zichtbaar werd in de juiste richting. De tweede groep proefpersonen bewoog de muiscursor naar het midden van het scherm, op gelijke afstand van alle stimuli. Dit is een optimale positie als er geen voorspelling kan worden gemaakt van de volgende target. Dit bleek ook bij het uitvragen van de reeks na afloop van het experiment: proefpersonen die de reeks expliciet hadden geleerd maakten meer voorspellende bewegingen, proefpersonen die de reeks niet expliciet hadden geleerd waren meer geneigd om de muiscursor naar het midden van het scherm te bewegen. Dit laat zien dat mensen—onafhankelijk van hun kennis—een strategie hanteren die optimaal is gegeven hun kennis.

Het modelleren van reinforcement learning

De SRT-taak heeft een beperkte ecologische validiteit, omdat mensen in het dagelijks leven niet simpelweg reageren op stimuli, maar hun omgeving exploreren en leren van interactie met objecten. Om deze reden hebben we een tweede variant van de SRT-taak gemaakt, gebruikmakend van een *reinforcement learning*-paradigma. In deze taak werd niet langer gereageerd op één van de vier oplichtende stimuli, maar moesten de verschillende alternatieven worden uitgeprobeerd waarna feedback werd gegeven over de correctheid van de keuze. Op deze manier werd dezelfde reeks als in de deterministische conditie van de SRT-taak afgewerkt, opnieuw samengesteld uit een 80 maal herhalende reeks van lengte 10. Voor iedere correcte beweging verdiende de proefpersoon 1 punt, voor iedere foutieve beweging verloor de proefpersoon 1 punt. Er bleek verrassend veel variatie te zitten in het aantal behaalde punten na het voltooien van 800 correcte bewegingen.

Reinforcement learning is een techniek uit machine learning die kan leren welke actie moet worden genomen in welke staat om een beloning te maximaliseren, geïnspireerd door operante conditionering. Reinforcement learning-modellen onderscheiden zich van andere technieken zoals supervised learning doordat zij geen gebruik maken van gelabelde trainingsdata, maar door een proces van trial-and-error leren welke acties de meeste beloning opleveren. Dit doen zij door een verwachte beloning, een *Q-value*, toe te kennen aan combinaties van *states* en *actions*.

Om het gedrag van proefpersonen beter te onderzoeken hebben we geprobeerd drie bestaande reinforcement learning-modellen toe te passen op de verzamelde data: (1) Q-learning, (2) SARSA, en (3) Q-learning met eligibility traces. Geen van de onderzochte modellen kon de hoogste scores van proefpersonen evenaren. Dit heeft vermoedelijk te maken met de gebruikte actie-selectiestrategie. Bij het gebruik van een *softmax* actieselectiestrategie zouden mogelijk betere resultaten kunnen worden verkregen, dit is onderwerp van vervolgonderzoek.

Inter-individuele verschillen in prestatie

Hierna hebben we een grotere groep proefpersonen getest, en hebben we een aantal additionele taken afgenomen om te onderzoeken of de verschillen tussen proefpersonen werden veroorzaakt door het al dan niet moedwillig kiezen van verschillende strategieën of beperkingen in werkgeheugen of IQ. Uit eerder onderzoek is bekend dat proefpersonen onder sommige omstandigheden een strategische keuze kunnen maken tussen verschillende manieren van handelen [160]. In *stimulus-based control* delegeert de proefpersoon controle aan de externe stimulus. Versnelling zal hier veroorzaakt worden door het versneld reageren op de stimulus. In *plan-based control* maakt de proefpersoon een interne representatie van een motorplan. Hier kan de proefpersoon actief een voorspelling maken van de volgende stimulus.

In de SRT-taak werd leerprestatie, gemeten door de hoeveelheid expliciete kennis van de reeks, voorspeld door de capaciteit van het visuospatieel

werkgeheugen. In de reinforcement learning-taak werd prestatie, gemeten door de totale score aan het einde van de taak, voorspeld door zowel IQ als de capaciteit van het visuospatieel werkgeheugen. Dit suggereert dat de verschillen in prestatie niet veroorzaakt werden door het kiezen van verschillende strategieën, maar door cognitieve beperkingen.

Het modelleren van optimale bewegingen

Om de geoptimaliseerde muisbewegingen die zichtbaar waren in hoofdstukken 4 en 5 nader te onderzoeken, hebben we in hoofdstuk 6 een robotarm gesimuleerd, aangestuurd door een kunstmatig neuraal netwerk. Deze robotarm kreeg dezelfde SRT-taak als proefpersonen, en de netwerken werden met een evolutionair algoritme getraind om zo snel mogelijk de stimulus die actief werd aan te raken. Er werden drie condities gebruikt: (1) een conditie waarin het netwerk nauwkeurige informatie kreeg over de volgende target (perfecte voorspelling), (2) een conditie waarin het netwerk willekeurige informatie kreeg over de volgende target (niet-informatieve voorspelling), en (3) een conditie waarin geen informatie werd verstrekt aan het netwerk (geen voorspelling).

De beste prestatie, gemeten door snelheid en nauwkeurigheid, werd geleverd door de netwerken die perfecte informatie over de volgende target kregen. Deze netwerken stuurden de robotarm naar de volgende target, nog voordat deze target actief werd. Zij leerden dus gebruik te maken van de informatie die aan ze werd verstrekt. De netwerken met niet-informatieve en geen voorspellingen scoorden minder goed. Zij evolueerden een strategie analoog aan die van mensen zonder expliciete kennis van de reeks: ze bewogen de robotarm naar het midden van de stimuli, op gelijke afstand van alle potentiële targets. Ook was zichtbaar dat de netwerken met niet-informatieve voorspellingen langzamer evolueerden dan de netwerken zonder voorspellingen. Het kost blijkbaar tijd om de willekeurige invoer te negeren. Vervolgonderzoek zou kunnen uitwijzen of dit vergelijkbaar is met proefpersonen die *denken* expliciete kennis over de reeks te bezitten, maar dit in feite niet hebben.

Curriculum vitae

Roy de Kleijn was born on July 27th, 1982 in The Hague, Netherlands. He finished high school at the Dalton Scholengemeenschap in The Hague in 1999. After initially being trained as an air traffic controller but not quite happy with his chosen career path, and after brief stints in mathematics and political science, he found his place in cognitive psychology. In 2010 he received his M.S. in cognitive neuroscience from Leiden University based on research carried out together with and supervised by Prof. Jay McClelland at Stanford University, combined with coursework in artificial intelligence, resulting in the thesis *Computational modeling of individual differences using stochastic information accumulation models.* After this, he started his Ph.D. under supervision of Prof. Bernhard Hommel as part of the European RoboHow consortium, while pursuing an M.S. in computer science at Georgia Tech. After receiving his Ph.D., Roy will remain at Leiden University as an assistant professor, working on computational models of cognition.

Acknowledgments
Dankwoord

There are no happy endings.
Endings are the saddest part,
So just give me a happy middle
And a very happy start.

—Shel Silverstein, *Happy Ending?*

Vijf jaren promotieonderzoek zitten er op. Hoe mooi het ook is om bezig te zijn met wat je leuk vindt, in mijn eentje zou het veel minder aangenaam zijn geweest. Dit proefschrift zou niet compleet zijn zonder de mensen te noemen aan wie ik het te danken heb.

Bernhard, toen je me tegen het einde van mijn studie vroeg of ik al enig idee had wat ik erna zou willen doen had ik dat natuurlijk niet. Toen bleek dat er een mogelijkheid was om onderzoek te doen naar robots én pannenkoeken (twee van mijn favoriete dingen) was de keuze snel gemaakt. Ik hoop dat we nog lang mogen samenwerken, want gesprekken met je zijn altijd inspirerend.

George, sharing an office with you was great. Many memories in great places: Grande Allée, Emerald Isle, Stubu, New York, detours through Kentucky, and Leiden and Amsterdam. And, of course, writing papers and preparing talks together, some during the day, some during the night. Thanks for the good yet unexpectedly productive times.

Marc en Manja, leuk om samen met jullie onderzoek te doen in vakgebieden anders dan die van mij. Onze samenwerking is hét bewijs dat de leuk-

ste interdisciplinaire wetenschappelijke ideeën ontstaan op de minst wetenschappelijke plaatsen.

Mijn collega's Bernadet, Kerwin, Marieke, Fenna, Henk, Hilâl, Roderik, Guido, Sander, Wido, Lorenza, Jop, Pascal, het ERC-team, bedankt voor het voorrecht dat ik áltijd met plezier naar mijn werk ga.

Wally, eeuwig bedankt voor je mentale ondersteuning en met name het nalezen van dit proefschrift. Volgens mij zijn de grootste foutjes er nu uit.

Tot slot. Tijdens je studietijd maak je vrienden voor het leven, zeggen ze. Hoe waar dat is bewijst onze vriendengroep die is ontstaan tijdens onze eerste week in Leiden, na twee lustra is dit wel duidelijk. Bedankt Anouk, Nathalie, en Liesbeth.

Nathalie en Merel, ik ben zeer dankbaar dat jullie de onmisbare taak van het paranimfschap hebben aanvaard. Jullie zijn me zeer dierbaar en ik kan me geen betere paranimfen bedenken.

Jordy, goed dat je je draai hebt gevonden in de klinische psychologie, en ik vind het fijn dat je inmiddels naast broer ook een goede vriend bent.