Transferring Insights on Mental Training to Robot Motor Skill Learning

Übertragung von Erkenntnissen des Mentalen Trainings auf Konzepte für motorische Lernvorgänge von Robotern

Bachelor-Thesis von Sebastian Szelag aus Darmstadt

Oktober 2017

TECHNISCHE UNIVERSITÄT DARMSTADT

Transferring Insights on Mental Training to Robot Motor Skill Learning
Übertragung von Erkenntnissen des Mentalen Trainings auf Konzepte für motorische Lernvorgänge von Robotern

Vorgelegte Bachelor-Thesis von Sebastian Szelag aus Darmstadt

1. Gutachten: Prof. Dr. Jan Peters
2. Gutachten: M.Sc. Dorothea Koert

Tag der Einreichung:

Erklärung zur Bachelor-Thesis

Hiermit versichere ich, die vorliegende Bachelor-Thesis ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.
In der abgegebenen Thesis stimmen die schriftliche und elektronische Fassung überein.

Darmstadt, den 10. Oktober 2017

(Sebastian Szelag)

Thesis Statement

I herewith formally declare that I have written the submitted thesis independently. I did not use any outside support except for the quoted literature and other sources mentioned in the paper. I clearly marked and separately listed all of the literature and all of the other sources which I employed when producing this academic work, either literally or in content. This thesis has not been handed in or published before in the same or similar form.
In the submitted thesis the written copies and the electronic version are identical in content.

Darmstadt, October 10, 2017

(Sebastian Szelag)

Abstract

Humans exhibit impressive abilities to plan and improve movements with mental forecasts. Through this, highly complex motor skills can be learned and enhanced in a short amount of time. Transferring these abilities of mental imagery and mental training to the field of robotics could possibly improve motor skill learning of robots as well. The approach introduced in this thesis focuses on the subject “Learning without doing”. To this end, this thesis presents a concept to train robots in "mental environments", for improved motor skill learning. The thesis connects the basic concepts of mental imagery used by humans with existing techniques of robotics. For this connection, neuropsychological inspired concepts are discussed and linked to robotic approaches for motor skill learning such as dynamical movement primitives (DMP). The thesis presents an overview of human motor skill learning and introduces a concept to transfer these structures to a mixture of experts based architecture of movement primitives. As a proof of concept, the single components of this introduced concept are applied to the example of robot ball catching and advantages and current limitations of the approach are discussed.

Keywords: mental training, motor skill learning, mixture of experts, dynamical movement primitives

Zusammenfassung

Die Fähigkeit eines Menschen, geplante Bewegungen mental vorrausschauend auszuführen, ist grundlegend für die Entwicklung von motorischen Fähigkeiten. Hierdurch können innerhalb kürzester Zeit hoch komplexe Bewegungen erlernt und perfektioniert werden. Insbesondere diese Fähigkeiten sind in dem Forschungsfeld der Robotik von elementarer Bedeutung. Der vorgestellte Ansatz dieser Thesis beschäftigt sich mit der Thematik „Lernen ohne Ausführen“. Hierbei wird es Robotern ermöglicht, „mentale Umgebungen“ zu nutzen, um motorische Fähigkeiten zu erlenen. Diese Thesis verknüpft Grundlagen des mentalen Trainings von Menschen mit vorhanden Techniken der Robotik. Hierfür werden neuropsychologisch inspirierte Konzepte vorgestellt, die mit Techniken des maschinellen Lernens, wie Bewegungsprimitive, umgesetzt werden können. Aus der hieraus resultierende Synthese werden neue Konzepte zu kreiert und Rahmenstrukturen gebildet. Exemplarisch hierfür wird das Konzept auf ein klassisches Roboter-Ball-fang Experiment angewendet, um Vorteile und Grenzen aufzuzeigen.

Schlüsselwörter: Mentales Training, Motorisches Lernen, Mixture of Experts, Movement Primitives

Contents

1. Introduction **2**
1.1. Motivation . . . 2
1.2. Research Questions . . . 3

2. Foundations **4**
2.1. Biological Foundations . . . 4
2.2. Foundations in Machine Learning . . . 7
2.3. Mixture of Experts . . . 9
2.4. Mixture of Movement Primitives . . . 14

3. Related Work **16**
3.1. Neuropsychological Research Related to Motor Imagery . . . 16
3.2. Mental Imagery for Robot Learning . . . 17
3.3. Existing Experiments in Mental Imagery . . . 20

4. Concepts **22**
4.1. Concept for Neuropsychological Movement Creation . . . 22
4.2. Concepts for Bio-inspired Motor Skill Learning with Mental Training . . . 24

5. Experiments **28**
5.1. Learning and Generating Trajectories . . . 28
5.2. Mixture of Experts . . . 31
5.3. Using Dynamical Movement Primitives with Referred Trajectories . . . 33
5.4. Application to a Restricted Workspace in Robot Ball Catching . . . 36

6. Conclusion and Future Work **40**

Bibliography **43**

A. Appendix **45**
A.1. Neural Network for Mental Imagery . . . 45

Figures and Tables

List of Figures

1.1. Humans exhibit impressive abilities to improve and accelerate motor skill learning due to the use of mental training. Hereby they deploy internal models of their body and the environment.Transferring these basic concepts from mental training to robot motor skill learning may also result in additional benefits such as less hardware usage and safer learning processes. 2

2.1. The occourence of LDs is more likely at the age of 10-14 [4]. 4
2.2. A gating network of 4 experts, which approximate a given function 9
2.3. Similar to MoE a gating network defines the usage of each motor primitive. The output is the desired movement, for a given stimulus **x** [1] 15

3.1. Different sports and the number of respondents. R = recreation/house league, LC = local or regional competitive, PC = provincial or state competitive, NC = national/international competitive. 16
3.2. Signifiant Questions of the Questionnaire[31]. 17
3.3. Experimental device used for the experiment. The device consisted of a vertical bar, 2 parallel aluminum dowels fixed on the bar, and 9 switches (T0 ... T8) fixed on the dowels and connected to a chronometer integrated into a laptop[40]. 18
3.4. The ventral pathway consists of Prefrontal Cortex (PFC), Superior Temporal Cortex (STC) and Ventral Occipito- Temporal Cortex (VOT). The dorsal pathway consists of Premotor cortex (PMC), Parietal Cortex (PC), Visual Cortex (VC) and Somatosensory Cortex (SSC). Both pathways are linked [20]. 19
3.5. Simulation environment and used object for the experiment of mental rotation [20]. 20
3.6. The model of mental image rotation. Each box represents the model's components. The arrows defines the information flow between the different components [20]. 21

4.1. Graphical visualisation of the different brain regions used for motor imagery (in green) and physical Training. Including Frontal lobe, Parital lobe, Occipital lobe, Temporal lobe, Cerebellum. 22
4.2. Motor skill learning include many brain Regions. Basal ganglia, cerebellum, primary motor cortex, premotor cortex, supplementary motor are, parietal cortex, visual cortex, auditory cortex, somatosensory cortex and prefrontal cortex are the primary brain regions engaged in skill learning and development. 23
4.3. The dynamical model enhances over time, based on the captured dynamics of the physical training. First column (4.3a , 4.3d) represent the physical and mental training based on a linear model without gradient. Second column (4.3b , 4.3e) represent the physical and mental training based on a linear model with gradient. Third column (4.3c , 4.3f) represent the physical and mental training based on a parabolic model. 24
4.4. Simple pipeline of physical training coupled with motor imagery training. η and χ are used for counting physical and mental training iterations. The control flow of physical training is defined in Figure 4.6 , motor imagery is defined in Figure 4.5. 25
4.5. Pipeline of a mental training iteration, the iteration could be handled as a "Task" in a black-box concept. A Task is based on previous collected information of the ball Trajectory, on which a new ball with modified information can be thrown to train the robot, using the existing ball model. 25
4.6. Physical training is not very different from mental training the significant, difference is a new input of information from the real world, instead of generating new "pseudo" information of the model. 26
4.7. The concept of trajectory planning during motor imagery. 26
4.8. The pipeline of simulation defined in [26] can be used to describe differences between Simulation and motor imagery. 27

5.1. Setup of the experiment using the OptiTrack system and OptiTrack markers. 28

5.2. Experiment of Setting A, with capturing noise . . . 29
5.3. Experiment of Setting B, with capturing noise and additional Noise = 0,02. . . . 30
5.4. Prediction for different trajectories with the same learned model . . . 30
5.5. Imagined ball trajectories based on the captured dynamics during physical training. . . . 31
5.6. Pictures a) to d) show the adaption procedure of the experts. Resulting is a conditional mixture e). The dependencies for each expert are seen in Figure f). The Log-likelihood for different initialization the iterations is shown in g). The error for different initialization can be seen in h) . . . 32
5.7. Examples of DMP which imitates a desired path.5.7a, 5.7b, 5.7c, 5.7d learned data from a hand-drawn demonstration. 5.7e, 5.7f, 5.7g, 5.7h, 5.7i, imitate given functions. . . . 33
5.8. The green graph is the used demonstration for DMP, the Normal DMP (orange) reaches the goal point with a descent lose of the given characteristic. The enhance DMP reaches its goal point, while the shape of the demonstration is complete recognizable. . . . 34
5.9. For a goal point far from the goal of the feature, DMP can become unstable. The enhanced DMP with feature scaling, reaches this point, while the characteristic of the demonstration is recognizable. . . . 34
5.10. Different behaviors by different α_z. . . . 35
5.11. Basis functions which approximates a referred trajectory of a sinus like demonstration. . . . 36
5.12. Different behaviors by a different number of basis functions. . . . 36
5.13. Different behaviors by different β -Parameters. . . . 37
5.14. Different characteristics of DMP to reach the same point. DMP 1 uses a sine demonstration, DMP 2 uses a parabolic demonstration and DMP 3 is linear. . . . 37
5.15. 7 "imagined" Trajectories, for 2 different DMP. . . . 38
5.16. The robot (blue) performs a catch action for the demonstration of an incoming ball. An obstacle restricts the movement. The desired trajectory is represented by the orange line. . . . 38
5.17. Used variables for the 2-DoF robot. . . . 39

6.1. Extending the concepts presented in this thesis to enable robots to test different combinations of motor skills by using mental training is a possible line for future work. Hereby, the robot could test combinations of skills for moving its base and catching a ball via mental training. . . . 41

A.1. Model of a simple feed forward neural network, with input, and hidden output Layer. Using three Input unit, two hidden units and one output unit [2]. . . . 46
A.2. Model of a reccurent neural network by Elman, with hidden-unit activations feedback [3]. . . . 47

List of Tables

1 Introduction

1.1 Motivation

Nature can provide valuable inspiration for many problems in the research fields of robotics and machine learning. One of the most popular examples may be neural networks [29] but reinforcement learning can also be found to be inspired from biological concepts [27]. In the last decade, a small number of researchers tried to transfer the concepts of mental imagery to theses state-of-the-art approaches [20]. For humans, mental imagery is a daily companion. It is used to imagine planned situations, to learn tasks [38], and especially in sports mental imagery can lead humans to more success, faster and more precise movements and smaller error rates. Hereby, biological experiments revealed that a proportion 75% of mental training to 25% of physical training is nearly as good as physical training alone, but with less energy consumption [37].
Researches show an interest in mental imagery to solve robotic tasks. Many all approaches are based on neuronal networks, especially a combination of feed-forward neuronal networks and recurrent neuronal networks is often used to train the robot. For example, the task of mental rotation, which is an object recognition tasks, is solved with a population code approach [20]. Another experiment shows motor skill improvements on the iCub robot, using the same technique of neural networks.
This thesis focuses on the fact that mental imagery can lead to the same improvements on robotics as on humans, especially on motor skill training. Mental imagery can be seen as aa brain-internal simulator to get feedback for imagined actions, based on previously captured knowledge from the environment.
Similar to the biological use, motor imagery can be seen as an add-on framework to the performed physical training.
An additional usage of mental imagery is to use it as an advanced simulation tool. This technique is very similar to the concept of lucid dreams, which is in biological research often connected with the environment exploration by children or young adolescents. Mental imagery could enable a robot to explore its environment and to iteratively incorporate more data to capture a model of its surrounding. Based on the captured model, the robot is able to perform motor imagery training with the model of the environment to explore it and improve its own model. This can lead to improved simulations which are not static in the given data. In the presented concept, simulation data becomes dynamic, because the model update is based on the captured information of the robot.
In particular, mental training and motor imagery can lead to an accelerated and safer learning process due to the fact that movements can be "imagined" they are performed. This may additionally also result in lower energy consumption because several physical motions can be replaced in some cases with imagery and are therefore often more cost efficient.

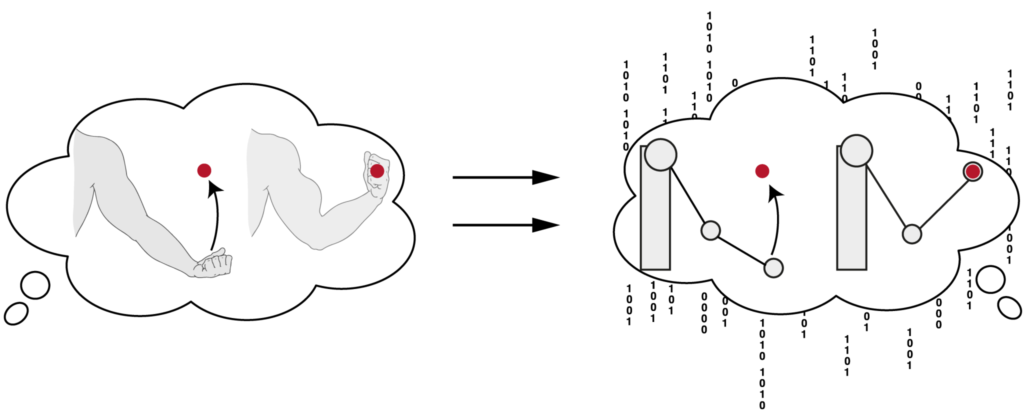


Figure 1.1.: Humans exhibit impressive abilities to improve and accelerate motor skill learning due to the use of mental training. Hereby they deploy internal models of their body and the environment.Transferring these basic concepts from mental training to robot motor skill learning may also result in additional benefits such as less hardware usage and safer learning processes.

Motor imagery is a subgroup of mental imagery, which is related to motor skill training; especially in the robotics jargon mental imagery is often defined as motor imagery. The transfer process is illustrated by Figure 1.1. All the described concepts and experiments share one basic idea in common, the concept of data capturing and data recreation. By using exactly this concept, the further aim of this thesis is to show in simulated motor imagery, that a combination of physical training and mental training can be used to improve policies of mixture of movement primitives, which are an approach of motor skill learning.

1.2 Research Questions

To develop a concept, which transfers insights from mental training to robot motor skill learning, this thesis investigates the following research questions.

- **Which biological and neuropsychological processes are used in mental imagery for motor skills learning?**
  To gain insights on biological and neuroscientific processes, Chapter 2 and Chapter 3 review existing literature from different research fields on mental imagery for motor skill learning.

- **Which Mental training concepts are used for robotics in related work, which advantages and disadvantages are provided?**
  Chapter 3 discusses established concepts of mental imagery, which are implemented on robots.

- **How to transfer neuropsychological concepts to robotics?**
  Chapter 4 presents a summary of the process of motor skill learning and introduces a concept to include mental imagery for a mixture of experts architecture of movement primitives. Chapter 5 shows experiments for the single components of the introduced concept on the example of robot ball catching.

2 Foundations

This chapter provides an overview of the foundations, for neuropsychological inspired machine learning. Mental imagery inspired motor skill learning is based on two different sciences: computer science in the form of machine learning and robotics; and biology, in the form of neuroscience. This chapter summarizes the necessary existing foundations from those research fields, to show similarities and differences between the interpretations of motor skill learning in these sciences.

2.1 Biological Foundations

Mental training is widely spread in biological motor skill training. Several kinds of mental training are possible, for example in the form of lucid dreams or imagery training for sports sciences. It seems that mental imagery helps humans to learn faster, by rethinking situations, while motor regions get stimulated [40].

2.1.1 Lucid Dreams

Related to the topic of mental imagery is the appearance of lucid dreams (LD), LD can be seen as a mentally created environment, in which the dreamer is able to interact in the learned way with a given object[4]. LD are reams where the dreamer is aware that he or she is dreaming. Mostly, the first LD occur at the age of 10 - 14 (Figure 2.1).
A study from the website klartraum.de shows that the most people choose in their dreams intuitively between three different topics: flying, talking to imagined people or having sex. Other topics that are related to the motor context are also chosen in LD, but typically related to the three main topics. A survey of Schaedlich et, al [5], found out that lucid dreamers are able to to perform intended actions in a mental environment. In the survey, a group of 301 lucid dreamers, use their lucid dreams to have fun (81%) and to turn nightmares into more pleasant ones (64%). Some of the lucid dreamers (30%) try to use their LD for problem solving and others (21%) use their dreams to train their practicing skills. So motor skill training is possible during a LD, additionally it is important, to note that this study is not directly related to motor skill learning, though it is interesting that 21% automatically use their LD for motor skill training. [4]. Experiments and research that is related to motor skill learning was not found.

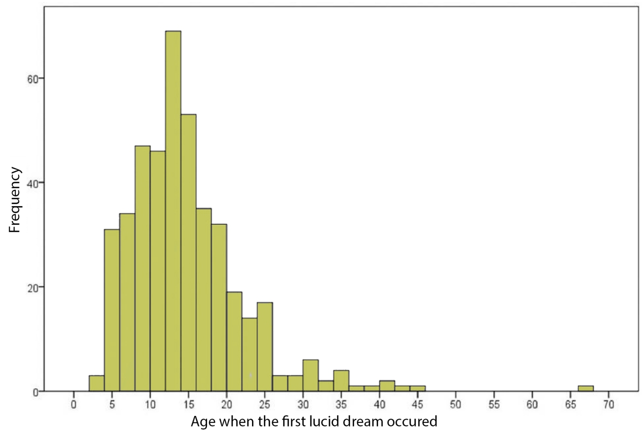


Figure 2.1.: The occourence of LDs is more likely at the age of 10-14 [4].

2.1.2 Brain regions used for Mental imagery

Several different brain regions are related to mental imagery. All of these brain regions are equally related to motor skill development. The following sections describe the functions of the related brain regions

Parietal Cortex

The parital cortex is involved in action prediction. Electrophysiological studies show that the parietal cortex joins the motor planning in an early stage. The neurons of the parietal cortex can predict sensory changes in anticipation of intended movements. Studies of Anderson and Buneo [6] show that this region is used for planning high-level cognitive trajectories for movements. Thus it takes a central role in combining sensory and motor information in a common reference frame. According to this view the sub-regions in the posterior parietal cortex contain maps of intention, that are related to the planning process of several movements. To these movements belong eye movements, reaching movements and grasping movements. The cerebellum and the parital lobe, which interact with the parietal cortex, appear to be involved in the cancellation process of self-produced sensory stimulation. The detection of mismatches between desired and actual movement correction also belong to the tasks of the parital cortex. Studies show that in particular the visual feedback is important for this process. A recent study of Desmurget et al. [7] in which subjects should reach a goal with their arms, while the vision of the arm was prevented, the target was visible for the subject. In some trails the target was moved during the procedure. In these cases, the accuracy decreases, the subjects tended to reach the first position. A similar behavior is known from patients with bilateral damaged posterior parital cortex. [8] This effect shows how strongly visual cortex and parital cortex are connected, this connection may influence the concept of motor imagery. During motor imagery training, the actual visual information, comes from the existing knowledge, not from the real-world information on the environment.
Prediction requires the ability to prepare and visualize moments through imagery, in this context the parietal cortex, seems to play a critical role. During movement imagery, several brain regions are activated, including the cerebellum, parietal cortex, and premotor cortex, when performing physical movements these regions are equally active. Thus, imagined and executed movement are normally equal in execution time. Even according to Fitt's Law accounting equally well for both types of movement, imagery and real execution show the same activation in the parietal cortex. This was found to be true via an experiment with a motor cortex damaged patient. Whereas in patients with parital lesions, actual movement execution was modulated by the size of the target, but not during motor imagery. [8]

Posterior Parital Cortex

In the 19th century, the parietal cortex was believed to be an area, especially the posterior parietal cortex, which pervades association tasks by associating different sensory modalities. Research in the last years shows, that there are two functional pathways in visual cortex, a dorsal pathway that in includes the posterior parietal, so these regions are involved in spatial perception; and a ventral pathway which is involved in object perception. [6]

Basal Ganglia

Voluntary and unintentional movements are based on several spatial and temporal patterns of different muscles, the central nervous, system and different brain regions like the premotor supplementary motor cortex. All these actions are initiated by the basal ganglia, which plays an important role in the learning process of new movements. By planing, learning and executing motor skills it is crucial for meaningful trajectories. Ample research indicates it cooperates with several other brain regions in a topographically organized and segregated loop ,linking motor cortical regions like the primary and supplementary and cingulate motor cortex, which belongs to the cerebral cortex, to produce and learn motor skills. [9] Especially the combination of the planning abilities and the cerebellum that supports the generation of movements by "selecting" meaningful movements is important for coordinated movements. Several brain structures are included in the basal ganglia, it consisting of the striatum, the pallidum, the subthalamic nucleus ant the substantia nigra. Theses region are important for the cortex-to-cortex communication, the striatum receives inputs from the cerebral cortex and the thalamus [10]. These two brain regions are crucial for all sensory systems, for computing these information they work in an intriguing back-and-forward aspect [11]. It appears that the basal ganglia plays a significant role in sensorimotor, cognitive and behavioral processes that relate to the learning and execution of motor functions. A disease like Tourette's syndrome is related to a lesion of the basal ganglia, the results are spontaneous unwanted, stereotyped movements or unwanted behavioral acts. These symptoms can be interpreted as the result of a defective suppression mechanism inside the basal ganglia [10].

Cerebellum

The cerebellum is crucial for skillful smooth movements [12]. Patients with damaged a cerebellum show a very poor performance on coordinated movements, balancing problems and unclear pronunciation. Another factor which illustrates the absence of the computational usage of the cerebellum, is alcohol which also leads to these three aftermaths.

There are a number of theories of the cerebellar function; many involve the idea that it generates a “model” of how the motor systems work – a kind of virtual reality simulator of your own body, inside your head. It builds this model using the synaptic plasticity that is embedded into its intricate network [12]. Recent works shows that trained mental imagers shows weaker activation’s of the cerebellum [13], they use instead regions of parietal and ventrolateral premotor regions, so areas like the prefrontal cortex are used more often.

Prefrontal Cortex

The prefrontal cortex (PFC) occupies nearly the complete frontal lobe, the PFC works as association cortex. Research indicates that the PFC is involved in high level cognitive processes. Due to the fact that the prefrontal cortex is crucial for selecting a behavioral goal to be achieved by a movement, it is also part of the reinforcement learning process by evaluating movements [14]. This is demonstrated by processing languages, emotional processing and social behavior. It receives input from other limbic structures by way of other prefrontal cortical regions. Especially the development of the PFC, correlates with phenomena described in LD, the development of the PFC starts immediately after the birth, reaches its maximum at 6 years and ends after 20 years [15].

Motor Areas

The motor cortex is partitioned into three regions the primary motor cortex (M1), the premotor cortex and the supplementary motor cortex.

Primary Motor Cortex

The primary cortex prepares motor information in spatial form to be send to the spinal cord. During physical training, the primary motor cortex (PMC) and the supplementary motor area are active, though during mental motor training the primary motor cortex shows no activation. Single cell recordings show that the primary motor cortex codes movements in terms of space, but not in terms of specific muscle commands. The generated data is projected to the spinal cord, which uses motor-neurons to innervate muscles. Lesions of the PMC lead to paralysis, this could be attributed to the important connection of PMC to the spinal cord [14].

Premotor Cortex

The premotor cortex is deeply integrated in the perceptual-motor integration process, together with the posterior parital lobe. This processes is crucial for goal oriented movements with used hardware, which does not belong to the body itself, like racquets or other tools. A number of sigle-cell recordings studies indicate, that the premotor cortex outputs are associated with visually guided movements, so it fires preferentially visually guided movements. This is verified, by the fact that functional imaging studies show also strong activities in this region. [14].

Supplementary Motor Cortex

Recent theories of motor control emphasize neural prediction of movement outcomes from an internal reference copy [16]. Accordingly, the supplementary motor cortex (SMA) suggests motor predictions and interpretations by spontaneous movements receiving somatosensoric inputs. For example, the sensations produced when trying to tickle oneself are perceptually less ticklish than the same movements made by an external agent. Lesion studies in human and non-human primates show that lesions of the premotor cortex have a profound impact on the learning process relationship between perceptual cues and movements [16].

Visual cortex

The visual cortex (VC) is part of the occipital lobe, it is used to process visual stimuli. It consists of several areas, which deals with various aspects, like shapes, colors, movements and distances. Structured by a large number of cells in rows, experiment shows that neurons of each column fires to edges of the same orientation. Additional the neighbor neurons fires a slightly different output across the surface of the VC. This means cortical visual cells have an intrinsic organization for interpreting the same orientation, additional not an organization that is immutable [12]

2.1.3 Mental imagery as Product of Different Brain Regions

An experiment of Decety et al. [17], shows that PFC, SMA and cerebellum corresponds to mental motor imagery; all these regions are significantly activated during this process. During this experiment participants should imagine to draw the numbers: one, two and three in their mind. Additionally the subjects were instructed to imagine this scenario in a first person view and try to percept the imagined sensory data.

2.2 Foundations in Machine Learning

For the implementation of mental imagery several machine learning and robotic approaches are needed, the following sections present the techniques used in this thesis.

2.2.1 Dynamic Movement Primitives

A very popular approach to generate robot movements and trajectory representations are Dynamic Movement Primitives (DMP). Many state-of-the-art robot learning successes are based on this approach [18]. DMP can be used to describe movements of a robot. For this it can be assumed, that the DMP describes the position, velocity, and acceleration of a robot limb. This is done by defining attracted points, which will be reached due to the attractor dynamics of a DMP.

Planning Dynamical Movement Primitives with Demonstrations

For planning discrete or rhythmic movement plans, two different DMP approaches are possible: point attractive systems and limit-cycle systems. A useful way to use movement primitives consists of methods that don't need manual parameter tuning with the danger that DMP become unstable. In basic point attractive systems, the DMP are formulated with a second order dynamic equation 2.1. In which g defines the goal state, α_z and β_z are time constants, τ represents a temporal scaling factor, to increase or decrease the length of the DMP, and $y, \dot{y}$ which define the desired position and velocity generated by the used equation, as interpretation of the planed movement.
If the parameter setting is appropriate and $f = 0$, this equation form is a globally linear stable dynamic system and uses the goal attractor g. To change the function of f to a nonlinear function, for a trivial exponential convergence of y, and allow move complex trajectories to reach the goal, we enter the domain of nonlinear dynamics. The consequences are arbitrary complex resulting equations. This problem can be fixed by an additional canonical dynamic system 2.2 and an additional nonlinear function f 2.3 in which ψ_i is represented as 2.4 The equation of 2.2 defines a second order dynamic system similar to equation 2.2. It is linear and offers a monotonic global convergence to g with a proper choice of α_v and β_v, additionally it is critically damped. DMP need initial conditions to work properly as phase variable to anchor the Gaussian basis functions of ψ_i, all state variable x, ν, y, z have to be set to zero. The Gaussian basis functions are defined by a center c_i and a bandwidth h_i. The parameter ν enable to use it as a "gating term" in the nonlinear function 2.3. The combination of the equations of 2.1, 2.2, and 2.3 as a system allows an asymptotically converging behavior to the point attractor g.
[19]

$$\tau \dot{z} = \alpha_z(\beta_z(g-y)-z), \quad \tau \dot{y} = z + f \tag{2.1}$$

$$\tau \dot{v} = \alpha_v(\beta_z(g-y)-v), \quad \tau \dot{x} = \nu \tag{2.2}$$

$$f(x,v,g) = \frac{\sum_{i=1}^{N} \psi_i \omega_i \nu_i}{\sum_{i=1}^{N} \psi_i} \tag{2.3}$$

$$\psi_i = \exp(-h_i(\frac{x}{g} - c_i)^2) \tag{2.4}$$

Imitating a desired trajectory is a supervised learning problem, in which we focus problem, in which we the parameters of w_i. The linear parameters ω_i of the dynamical system, allow to apply a variety of learning algorithms. By assuming that a desired behavior is given by a demonstration, from the desired trajectory we are able to calculate a desired velocity and acceleration, which are defined as $(y_d(t), \dot{y}(t), \ddot{y}(t))$, with $t \in [1, ..., T]$. Learning of the desired trajectory is performed in two steps: First, determining the high-level parameters of the goal g, start point y_0 and time scaling parameter τ. Second, learning the weight parameters of ω_i.

First Step

In discrete systems, the goal is simply the end position of the given trajectory $g = y_d(t = T)$, y_0 is defined as starting point of the desired trajectory $y_0 = y_d(t = 0)$. The high-level parameter τ should be compatible with the duration of the desired trajectory.

Second Step

Leaning of ω_i is performed with locally weighted regression (LWR). Other function approximators, like Gaussian Processes and mixture models, could also be used. In this case, LWR was used due to its fast one-shot learning procedure, additionally the fact that the distinct kernels learn independently of each other, is very useful to achieve a stable parametrization that is great for movement recognition evaluations.

To define the approximation problem, an arrangement of the attractor dynamics is needed. So the standard damped spring model 2.5 could be rewritten in first order notation 2.6 with 2.7.

$$\tau \ddot{y} = \alpha_z(\beta_z(g - y) - \dot{y}) + f \tag{2.5}$$

$$\tau \dot{z} = \alpha_z(\beta_z(g - y) - z) + f \tag{2.6}$$

$$\tau \dot{y} = z \tag{2.7}$$

For approximating a desired trajectory, 2.6, must be rewritten as 2.8

$$f_{\text{target}} = \tau^2 \ddot{y}_{\text{demo}} - \alpha_z(\beta_z(g - y_{\text{demo}}) - \tau \dot{y_{\text{demo}}}) \tag{2.8}$$

Equation 2.8 defines the approximation problem, in which f should be as close as possible to f_{target}. Locally weighted regression defines for each kernel Ψ_i in f the corresponding ω_i, this minimizes the locally quadratic error 2.9.

$$J_i = \sum_{t=1}^{T} \Psi_i(t)(f_{\text{target}} - \omega_i \xi(t))^2 \quad (t) = x(t)(g - y_0) \tag{2.9}$$

This equation can be solved by linear regression. [19]

$$\omega_i = \frac{s^T \Gamma_i f_{\text{target}}}{s^T \Gamma_i s} \tag{2.10}$$

$$s = \begin{pmatrix} \xi(1) \\ \xi(2) \\ \dots \\ \xi(T) \end{pmatrix} \quad \Gamma_i = \begin{pmatrix} \Psi_i(1) & & & \\ & \Psi_i(2) & & \\ & & \dots & \\ 0 & & & \Psi_i(T) \end{pmatrix} f_{\text{target}} = \begin{pmatrix} f_{\text{target}}(1) \\ f_{\text{target}}(2) \\ \dots \\ 8 f_{\text{target}}(T) \end{pmatrix} \tag{2.11}$$

2.3 Mixture of Experts

Mixture of Experts (MoE) is a popular machine learning approach. By using the divide and conquer principle, the data is decomposed by networks experts, controlled by a gating network.
The approach uses a modular architecture of local experts, in which the decomposed complex problems are much easier to learn [20]. The model has been used in numerous regression and classification algorithms and is practically used in health care, finance, surveillance, and recognition applications. The mixture of experts model can easily be combined with many different models, such as support vector machines (SVM), Gaussian processes (GP), and Hidden Markov models (HMM) [21]. Especially the well-studied statistical basis of MoE and models that can easily learn via techniques, such as expectation maximization (EM) and Markov chain Monte Carlo (MCMC) provide ME as useful and versatile algorithm. To represent the success story behind this model, we can look at a survey from 2008 that identified the top 10 most influential algorithms in data mining: k-Means, SVM. Apriori, EM, PageRank, AdaBoost, k-nearest neighborhood, naive Bayes, and classification and regression trees (CART). MoE is deeply related to most of these algorithms and improves its performance to by combining models of some of them [21].

2.3.1 Gating Network

As shown in 2.12 and (Figure 2.2) the mixture is defined as the sum of the gating network output $g_j(\mathbf{x})$ multiplied by the output value of $\mathbf{y}$ for each input vector $\mathbf{x}$. The gating network defined by 2.13, in which $g_j(\mathbf{x}, \nu)$ are the gating networks of the experts of $j = 1...K$. α denoted the prior probability, and $P(x, \nu_j)$ is a density function, given by the Gaussian 2.14. A probabilistic interpretation of Moe can be be given in the context of mixture models for conditional probability distributions 2.15 [21]. $g_j(x, \nu)$ is actually the posterior probability $P(j|\mathbf{x})$, x is assigned to the corresponding partition to the j-th expert net, referred to the Bayes' rule 2.16.

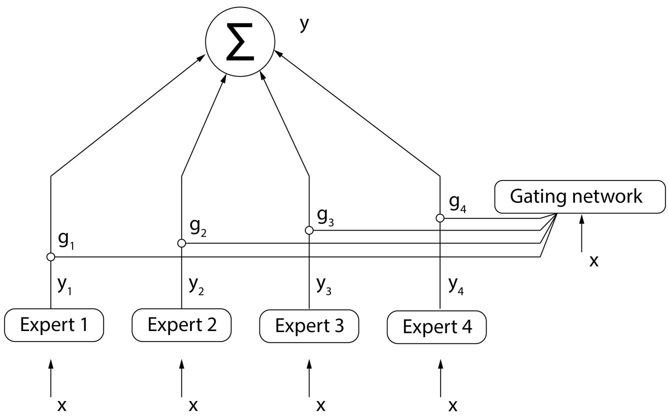


Figure 2.2.: A gating network of 4 experts, which approximate a given function

α_j	Prior of the j-th expert
$\mathbf{\Sigma}_j$	Variance of the j-th expert
$\boldsymbol{\mu}_j$	Mean of the j-th expert
g_j	Gating network component of the j-th expert
φ_j	Feature of the j-th expert
$\boldsymbol{t}_j$	Target vector for the j-th expert
z_j	Gating network outputs before thresholding

$$y(\mathbf{x}) = \sum_{j=1}^{m} g_j(\mathbf{x}) y_j(\mathbf{x}) \tag{2.12}$$

$$g_j(\mathbf{x}, \boldsymbol{\nu}) = \frac{\alpha_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \mathbf{\Sigma}_j)}{\sum_i \alpha_i \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \mathbf{\Sigma}_j)}, \qquad \sum_j \alpha_j = 1, \quad \alpha_j \geq 0 \tag{2.13}$$

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}_j|^{\frac{1}{2}}}\exp\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^T\boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_j)\} \tag{2.14}$$

$$p(\mathbf{t}|\mathbf{x}) = \sum_{j=1}^{m} g_j(\mathbf{x})\phi_j(\mathbf{t},\mathbf{x}) \tag{2.15}$$

$$g_j(\boldsymbol{x},\boldsymbol{\nu})) = P(j|\boldsymbol{x}) = \alpha_j\frac{P(\boldsymbol{x}|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)}{P(\boldsymbol{x}|\boldsymbol{\mu},\Sigma)}, P(\boldsymbol{x},\boldsymbol{\mu},\Sigma) = \sum_i \alpha_i P(\boldsymbol{x}|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j) \tag{2.16}$$

An analytical maximum likelihood estimation is not possible with this equation 2.17, but it is possible to rewrite this equation to 2.18

$$P(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}) = \sum_j \frac{\alpha_j P(\boldsymbol{x}|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)}{P(\boldsymbol{x}|\boldsymbol{\mu},\Sigma)} P(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}_j) \tag{2.17}$$

$$P(\boldsymbol{y},\boldsymbol{x}) = P(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})P(\boldsymbol{x},\boldsymbol{\mu},\Sigma) = \sum_j \alpha_j P(\boldsymbol{x}|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)P(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}_j) \tag{2.18}$$

2.3.2 Mixture of Experts Error Function

The architecture of the mixture of experts model consist of an expert network and a gating network, both access the input vector of x. Common models which are used in the mixture of experts are generalized linear models and multilayer perceptions. The following formula describes the mixture model of the mixture of experts model, in which are $g_j(x)$ the gating network outputs 2.19.

$$\mathbf{y}(\mathbf{x}) = \sum_{j=1}^{m} g_j(\mathbf{x})\mathbf{y}_j(\mathbf{x}) \tag{2.19}$$

The gating network g_j uses the divide and conquer principle to and can be regarded as the probability that expert j belong to the input x.
Due to the fact that mixture of experts can be interpreted probabilistic, the activation function can be defined as softmax function 2.20. In that equation z_i are the gating network output before the threshold. A benefit of the softmax function is that it is non-negative and its sums to unity.

$$g_j = \frac{\exp(z_j)}{\sum_{i=1}^{m}\exp(z_i)} \tag{2.20}$$

The probabilistic version of MoE 2.21 is given by the context of mixture models for conditional probability distributions. In which ϕ_j describes the conditional densities of the target vector for each expert j. The combination of ϕ and the softmax function guarantees that the distribution is always normalized.

$$p(\mathbf{t},\mathbf{x}) = \sum_{j=1}^{m} g_j(\mathbf{x})\phi(\mathbf{t}|\mathbf{x}) \tag{2.21}$$

Error Function

The standard way to motivate the error function, is the principle of maximum likelihood of the training data in combination with the input vector $\mathbf{x}_n$ and target vector t_n. The following equation 2.22 describes the dependence on these two vectors.

$$\mathcal{L} = \prod_n p(\mathbf{x}_n,\mathbf{t}_n) = \prod_n p(\mathbf{t}_n,\mathbf{x}_n)p(\mathbf{x}_n) \tag{2.22}$$

By taking the negative logarithm of the given likelihood, we can create the error function for the MoE model. Which is defined by 2.3.3.

$$E = -\sum_n \ln p(\mathbf{t}_n, \mathbf{x}_n) \tag{2.23}$$

For using the MoE error function based on a mixture of conditional probability densities, the following equation will be used. [22]

$$E = -\sum \ln \sum_{j=1}^{m} g_j(\mathbf{x}_n)\phi_j(\mathbf{t}_n, \mathbf{x}_n) \tag{2.24}$$

2.3.3 Expectation Maximization

Many standard optimization methods can be used to minimize the MoE error. In this thesis, expectation maximization is used, due to expandability and decoupled parameter estimation for the different components of the MoE model. [20]. Methods like Gradient Descent, are also possible solution methods. The rewritten formula 2.18 suggests an asymmetrical representation of the joint density. With methods like gradient descend it is possible to perform a maximum likelihood estimation based on $L' = \sum_n \ln(y_{(n)}, x_{(n)})$. This is done by determination of the parameters $\boldsymbol{\alpha}_j$, $\boldsymbol{\nu}_j$, $\boldsymbol{\theta}_j$ of the gating net and the corresponding expert nets.
Another way is to to minimize the error is expectation maximization (EM), which will be used. EM can be done by the following algorithm:

Expectation and Maximization Steps

The idea behind the maximization of the likelihood in mixture of experts, is to use a set latent variables, that indicate which expert belongs to each pattern. So each pattern is associated with only one expert, through the variable $z_{j,n}$ which is 1 if the pattern is generated by the expert j and o in other cases. Based on the previous given density function, the approach allows to see $p(t, z|x)$ as product of densities instead of the sum in 2.21.

$$p(\boldsymbol{t}, \boldsymbol{z}|\boldsymbol{x}) = \sum_{j=1}^{m} z_j(g_j(\boldsymbol{x})\psi_j(\boldsymbol{t}, \boldsymbol{x})) = \Pi(g_j(\boldsymbol{x})\psi_j(\boldsymbol{t}|\boldsymbol{x}))^{z_j} \tag{2.25}$$

The complete error function is given by, if the error function gets substituted .

$$E_c = -\sum_n \sum_{j=1}^{m} z_j^n \ln(g_j(\boldsymbol{x}_n)\psi_j(\boldsymbol{t}_n|\boldsymbol{x}_n)) \tag{2.26}$$

This substitution allows to reduce the product to a sum of separate error functions. An EM algorithm is able to reduce the error functions, this is done by two-step approach consisting of an expectation step and an additional maximization step. The behavior of this algorithm and the given error function guarantees the convergence to a local minimum. [22]

E-step

The expectation error function is defined as 2.27, the expected values are given by 2.28, by using Bayes' rule. This enables a probabilistic interpretation of MoE, with resulting definition of λ_j 2.29

$$\mathscr{E}(E_c) = -\sum_n \sum_{j=1}^{m} \mathscr{E}(z_{j,n}) \ln(g_j(\mathbf{x}_n)\phi_j(\mathbf{t}_n|\mathbf{x}_n)) \tag{2.27}$$

$$\mathscr{E}(z_{j,n}) = P(z_{j,n} = 1|\mathbf{t}_n, \mathbf{x}_n) = \frac{p(\mathbf{t}_n|z_{j,n} = 1, \mathbf{x}_n)P(z_{j,n} = 1|\mathbf{x}_n)}{p(\mathbf{t}_n, \mathbf{x}_n)} \tag{2.28}$$

$$\frac{g_j(\mathbf{x}_n)\phi_j(\boldsymbol{t}_n|\boldsymbol{x}_n)}{\sum_{i=1}^{m} g_i(\mathbf{x}_n\phi_i(\boldsymbol{t}_n|\boldsymbol{x}_n))} = \lambda_j(\mathbf{x}_n|\boldsymbol{t}_n) \tag{2.29}$$

$$\mathscr{E}(E_c) = -\sum_n \sum_{j=1}^{m} \lambda_j(\boldsymbol{x}_n, \boldsymbol{t}_n)\ln(g_j(\boldsymbol{x}_n)) - \sum_n \sum_{j=1}^{m} \lambda_j(\mathbf{x}_n, \mathbf{t}_n)\ln(\phi_j(\boldsymbol{t}_n|\boldsymbol{x}_n)) \tag{2.30}$$

By substituting the expected values of 2.29 the complete error function for the EM algorithm is defined by 2.30. This equation opens the possibility to minimize booth terms separately in each M-step. The first term defines the cross-entropy of the distributing pattern x among the expert network. This term increases when the experts share the given pattern 2.27. The second term has the general form of a weighted maximum likelihood problem, here a higher value for λ_j implies the weighting of the expert.

M-step

During the M-steps two equations are minimized, the gating network 2.31 and the expert 2.32. The minimizing step can be performed by using weighted least squares or by using the gating network with Gaussian kernels.

$$E_{\text{gate}} = -\sum_n \lambda_j(\mathbf{x}_n, \mathbf{t}_n)\ln(\phi_j(\mathbf{t}_{n|\mathbf{x}_n})) \tag{2.31}$$

$$E_{\text{expert}} = -\sum_n \ln(\phi_j(\mathbf{t}_n|\mathbf{x}_n)) \tag{2.32}$$

[22]

Gating Network with Gaussian Kernels

A way to reduce the M-step for the gating network is, to use a modified gating network consisting of normalized kernels. This is done by using Bayes' rule. For this the sum of the weights is defined as 1 $\sum_i \alpha_i = 1$, where each α has to be bigger than zero $\alpha_i > 0$. The probability density functions are given by P_i. By substituting in 2.21.

The gating network of a Gaussian Kernel is defined as 2.33.

$$g_j(\mathbf{x}) = P(j|\mathbf{x}) = \frac{\alpha_j P(\mathbf{x})_j}{\sum_i \alpha_i P_i(\mathbf{x})} \tag{2.33}$$

$$p(t|x) = \sum_{j=1}^{m} \frac{\alpha_j P_j(\mathbf{x})}{\sum_i \alpha_i P_i(\mathbf{x})} \phi_j(\mathbf{t}|\mathbf{x}) \tag{2.34}$$

$$P_j(\mathbf{x}) = \frac{1}{(2\pi\boldsymbol{\sigma}_j^2)^{\frac{d}{2}}} \exp\left(\frac{||\boldsymbol{x} - \boldsymbol{\mu}||^2}{2\boldsymbol{\sigma}_j^2}\right) \tag{2.35}$$

$$\alpha_j = \frac{1}{N}\sum_n h_j(\boldsymbol{x}_n, \boldsymbol{t}_n) \tag{2.36}$$

$$\mu_j = \frac{\sum_n h_j(\mathbf{x}_n, \mathbf{t}_n)\mathbf{x}_n}{\sum_n h_j(\mathbf{x}_n, \mathbf{t}_n)} \tag{2.37}$$

$$\sigma_j^2 = \frac{1}{d}\frac{\sum_j(\mathbf{x}_n, \mathbf{t}_n)||\mathbf{x}_n - \boldsymbol{\mu}_j||^2}{\sum_n h_j(\mathbf{x}_n, \mathbf{t}_n)} \tag{2.38}$$

Maximum Likelihood Estimate of μ and Σ

For understanding how the algorithm updates the used parameters, the update equations are defined by the Maximum likelihood estimate of the given error function. For this the experts of the estimate are defined as linear, these are substituted in the mean of the Gaussian kernel. By using the natural logarithm the product of the gating network and the experts becomes a sum, which is simpler to derive . By using λ_{jn} as $\mathcal{E}(z_{jn})$ to describe the error function in the EM-context,to minimize the error-function the following equation results. The conditional mixture is described as $p(\mathbf{x}|\mathbf{A_k}, \boldsymbol{\Sigma})$.

$$E_c = -\sum_n^N\sum_j^m \lambda_{j,n}\Big(\ln\big(g_j(\mathbf{x}_n)\big)\phi_j(\mathbf{t}_n|\mathbf{x}_n)\Big) \tag{2.39}$$

$$= -\sum_n^N\sum_j^m \lambda_{j,n}\Big(\ln\big(g_j(\mathbf{x}_n)\big) + \ln\big(\phi_j(\mathbf{t}_n|\mathbf{x}_n)\big)\Big) \tag{2.40}$$

$$= -\sum_n^N\sum_j^m \lambda_{j,n}\Big(-\frac{n}{2}\ln(2\pi) - \frac{1}{2}\ln(\boldsymbol{\Sigma}_j) - \frac{1}{2}\big(t_n - \mathbf{A_k}\mathbf{x}^T\big)(\boldsymbol{\Sigma}_j)^{-1}\big(\mathbf{t}_n - \mathbf{A_k}\mathbf{x}^T\big) + \ln\big(\phi_j(\mathbf{t}_n, \mathbf{x}_n)\big)\Big), \tag{2.41}$$

Derivative $\mathbf{A_k}$ The parameter $\mathbf{A_k}$, which describes the mean is estimable by using the following transformations 2.42, 2.43,2.44, 2.45 and results as 2.46.

$$\frac{\partial E_c}{\partial \mathbf{A}_j} = -\sum_n^N \lambda_{j,n}\big((\mathbf{t}_n - \mathbf{A}_j\mathbf{x}_n)\boldsymbol{\Sigma}_j^{-1}\mathbf{x}_n^T\big) \tag{2.42}$$

$$0 = -\sum_n^N \lambda_{j,n}\big((\mathbf{t}_n - \mathbf{A}_j x_n)\boldsymbol{\Sigma}_j^{-1}\mathbf{x}_n^T\big) \tag{2.43}$$

$$0 = -\sum_n^N \lambda_{j,n}\mathbf{t}_n\boldsymbol{\Sigma}_j^{-1}\mathbf{x}_n^T - \lambda_{j,n}\mathbf{A}_j\mathbf{x}_n\boldsymbol{\Sigma}_j^{-1}\mathbf{x}_n^T \tag{2.44}$$

$$\sum_n^N \lambda_{j,n}\mathbf{A}_j\mathbf{x}_n\boldsymbol{\Sigma}_j^{-1}\mathbf{x}_n^T = -\sum_n^N \lambda_{j,n}\mathbf{t}_n\boldsymbol{\Sigma}_j^{-1}\mathbf{x}_n^T \tag{2.45}$$

$$\mathbf{A}_j = -\frac{\sum_n^N \lambda_{j,n}\mathbf{t}_n\boldsymbol{\Sigma}_j^{-1}\mathbf{x}_n^T}{\sum_n^N \lambda_{j,n}\mathbf{x}_n\boldsymbol{\Sigma}_j^{-1}\mathbf{x}_n^T} \tag{2.46}$$

Derivative Σ The parameter $\boldsymbol{\Sigma}$, which describes the variance is estimable by using the following transformations 2.47, 2.48, 2.49, 2.50, until it is determinated by 2.51.

$$\frac{\partial E_c}{\partial \boldsymbol{\Sigma}} = -\sum_n^N \sum_j^m \lambda_{j,n} \Big(\frac{\partial}{\partial \boldsymbol{\Sigma}} \big(\frac{1}{2} \ln \boldsymbol{\Sigma}_j^{-1} \big) - \frac{\partial}{\partial \boldsymbol{\Sigma}_j} \big(\frac{1}{2} (\mathbf{t}_n - \mathbf{A_k x})^T \boldsymbol{\Sigma}_j^{-1} (\mathbf{t}_n - \mathbf{A_k}) \mathbf{x}) \big) \Big) \tag{2.47}$$

$$\frac{\partial E_c}{\partial \boldsymbol{\Sigma}} = -\sum_n^N \sum_j^m \lambda_{j,n} \Big(\frac{1}{2} \ln \boldsymbol{\Sigma}^{-1} - \frac{\partial}{\partial \boldsymbol{\Sigma}_j} \big(\mathrm{tr} \big[\frac{1}{2} (\mathbf{t}_n - \mathbf{A_k x})^T \boldsymbol{\Sigma}_j^{-1} (\mathbf{t}_n - \mathbf{A_k}) \mathbf{x}) \big] \big) \Big) \tag{2.48}$$

$$\frac{\partial E_c}{\partial \boldsymbol{\Sigma}} = -\sum_n^N \sum_j^m \lambda_{j,n} \Big(\frac{1}{2} \ln \boldsymbol{\Sigma}^{-1} - \frac{\partial}{\partial \boldsymbol{\Sigma}_j} \big(\mathrm{tr} \big[\frac{1}{2} (\mathbf{t}_n - \mathbf{A_k x})^T (\mathbf{t}_n - \mathbf{A_k}) \mathbf{x}) \big] \boldsymbol{\Sigma}_j^{-1} \big) \Big) \tag{2.49}$$

$$0 = \sum_n^N \lambda_{j,n} \frac{1}{2} \boldsymbol{\Sigma}_j - \frac{1}{2} \sum_n^N \lambda_{j,n} (\mathbf{t}_n - \mathbf{A_k x})(\mathbf{t}_n - \mathbf{A_k x})^T \tag{2.50}$$

$$\Sigma_j = \frac{\sum_n^N \lambda_{j,n} (\mathbf{t}_n - \mathbf{A_k x})(\mathbf{t}_n - \mathbf{A_k x})^T}{\sum_n^N \lambda_{j,n}} \tag{2.51}$$

2.4 Mixture of Movement Primitives

Mixture of Movement Primitives (MoMP) combines the benefactions of the MoE algorithm with the usage of DMP. Additional the ability to use reinforcement learning, in this approach, appears promising for implementation of neuropsychological concepts.

2.4.1 Skill Representation

MoMP uses the same equations for DMP generations, as described in 2.2.1. So DMP consists of two different differential equations, which are defined as canonical 2.52 and transformed 2.53 system. The canonical system acts as a phase z and is generated by 2.52.

$$\dot{z} = h(z) \tag{2.52}$$

$$\dot{y} = b(y, z, \omega) \tag{2.53}$$

The canonical systems influences the transformed system 2.53, and works as a kind of internal clock. By working with discrete movements with fixed start and end points or for reaching or pointing actions the canonical system can be chosen as 2.54

$$\tau \dot{z} = \alpha_z z \tag{2.54}$$

with τ defined as time constant and α_z as predefined constant parameter which defines the systems as stable.
f is called the transformation function which is given by 2.55.

$$f = \frac{\sum_i \boldsymbol{\omega}_i \phi_j(z) z}{\sum_i \phi_i(z)} \tag{2.55}$$

the Gaussian kernel of ϕ_i can be defined as $\phi_i(z) = \exp(-\rho(z - \mu_i^2))$ with center μ_i and bandwidth ρ_i.
To create mixtures of movement primitives, the new motor primitive $\pi(\mathbf{x})$ defined by 2.56 will be generated. $\pi(\mathbf{x})$ represents a composition of the used DMP. In the equation the number of used DMP is defined by c, while γ_i is used to describe individual weighting for the DMP. The kernel $k(\mathbf{x} - \mathbf{x_i})$ is similar to the gating network from the MoE approach of section 2.3. By a given stimulus $\mathbf{x}$, the kernel defines the probability for each DMP activation. This is done by using the predefined goals $\mathbf{x}_i$ from each DMP. Resulting is a gating network, which defines the dependencies for each DMP (Figure 2.3). [1]

$$\pi(\mathbf{x}) = \frac{\sum_{i=1}^c \pi_i \gamma_i k(\mathbf{x} - \mathbf{x_i})}{\sum_{i=1}^c \gamma_i k(\mathbf{x} - \mathbf{x_i})} \tag{2.56}$$

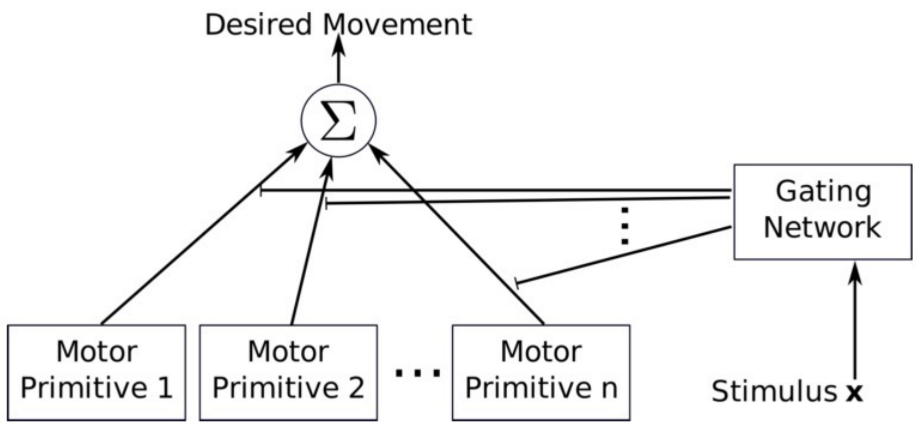


Figure 2.3.: Similar to MoE a gating network defines the usage of each motor primitive. The output is the desired movement, for a given stimulus **x** [1]

Primitive adaption by trail and error is one property of the MoMP approach. The adaption is used to optimize the internal parameters $\boldsymbol{\omega}$, which define the characteristics of each DMP. The adaption is learned by using the "Policy Learning by Weighting Exploration with the Return" (PoWER) approach. PoWER is an EM like approach for policy learning.
To update the parameter ω with PoWER, equation 2.57 is used.

$$\boldsymbol{\omega}_i^{N+1} = \boldsymbol{\omega}_i + \frac{\sum_j^N \varepsilon_i^j \frac{k(\mathbf{x_j}-\mathbf{x_i})}{\sum_{j=1}^{N} k(\mathbf{x_j}-\mathbf{x_l})} r_i^j}{\sum_{j=1}^{N} r^j} \tag{2.57}$$

This algorithms is used to update a single DMP. $\varepsilon_i \in \mathcal{N}(\mathbf{0}|\boldsymbol{\Sigma})$ is denotes the normally distributed exploration in which $\boldsymbol{\Sigma}$ is a meta-parameter that, that can also be optimized. r_i is the reward and N represents the number of samples. Reinforcement learning in MoMP is also able to adapt to responsibilities. The responsibility parameter γ_i is initialized with 1 and updated according to its contribution to successful and un-successful movements by 2.58.
[1]

$$\gamma_i^{N+1} = \exp\left(\kappa \frac{s_i - n_i}{\xi n_i + 1}\right) \tag{2.58}$$

In equation 2.58 κ is the counterpart of the learning rate. ξ is the trade-off parameter between exploration and exploitation. For using this equation the reward is defined as $r \in [0,1]$ [1].

$$n_i = \frac{\sum_{t=1}^{N} \gamma_i^N k(\mathbf{x_t} - \mathbf{x_i})}{\sum_{j=1}^{c} \gamma_j^N k(\mathbf{x_i} - \mathbf{x_j})} \tag{2.59}$$

$$s_i = \frac{\sum_{t=1}^{N} \gamma_i^N k(\mathbf{x_t} - \mathbf{x_i}) r_t}{\sum_{j=1}^{c} \gamma_j^N k(\mathbf{x_i} - \mathbf{x_j})} \tag{2.60}$$

3 Related Work

This chapter presents existing neuropsychological findings on mental imagery and motor learning without doing. Additionally it explains existing concepts that transfer mental imagery to robotics more detailed.

3.1 Neuropsychological Research Related to Motor Imagery

Mental imagery research is often connected with sport sciences, in which usually the mount of mental imagery is benchmarked. Only a few experiments compare in detail which benefits are expected by using mental imagery as training enhancement.

3.1.1 Mental Imagery in Sport Sciences

Sport science is an provides useful information about the performance increasement achieved during motor skill training sessions, that are supported by mental imagery.

The Use of Imagery by Athletes in Selected Sports

A researcher team of the University of Western Ontario, was interested in correlations between the mental imagery and success in sports. To get representative information, the team started a study with a 37-item questionnaire to samples of 381 male and female athletes of six different sports: football, ice hockey, soccer, squash, gymnastics, and figure skating. The level of the athletes starts at recreational/house league, up to national/international competitive leagues.

Questionnaire

The athlete respondents in the study came from six different sports (Figure 3.1). The questionnaire consists of 37 questions, in with scale between 1 and 7.

			R		LC		PC		NC	
Sports	Mean age	N. of athletes	M	F	M	F	M	F	M	F
Football	20.2	85	0	0	6	0	47	0	32	0
Hockey	18.8	107	0	0	22	0	78	0	7	0
Soccer	19.1	61	11	0	10	21	2	16	0	1
Gymnastics	14.7	50	3	4	2	6	2	22	3	8
Squash	26.6	32	5	3	7	4	1	2	6	4
Figure skating	14.2	46	0	3	3	21	5	10	2	2

Figure 3.1.: Different sports and the number of respondents. R = recreation/house league, LC = local or regional competitive, PC = provincial or state competitive, NC = national/international competitive.

Results

The results of the questionnaire show clearly that, athletics, which perform in higher leagues use mental imagery more often to optimize their performance. The following table show the most significant results, which indicate that mental imagery is more often used by athletes who perform in higher leagues. The original questionnaire is much longer 3.2

Inventory items	R	LC	PC	NC
To what extent do you use mental imagery in your training ?	3.62	3.77	4.49	3.39
To what extent do you use mental imagery in competition?	3.83	4.27	4.99	4.92
Do you use mental imagery before a practice?	2.45	2.94	2.79	3.22
Do you use mental imagery during a practice?	2.73	3.45	3.70	4.04
Do you use mental imagery after a practice?	2.47	2.64	2.74	2.74
Do you use mental imagery before an event?	3.39	3.87	4.04	3.89
Do you use mental imagery after an event?	2.51	3.42	3.43	2.91
How vivid is this image?	4.38	4.18	4.84	3.71
How easily can you see the entire skill?	4.23	4.41	5.05	5.09
When using mental imagery, to what extent do you actually feel yourself performing?	4.31	4.40	4.86	4.72

Figure 3.2.: Signifiant Questions of the Questionnaire[31].

The athletes reported in this study that they extensively use mental imagery. Especially successful athletes seem to perform mental practice more often. Athletes use imagery more in conjunction with competition than with physical practice. This suggest that mental imagery is more considered as a technique for performance enhancement than as learning aid [31].

Motor Learning Without Doing

Motor imagery is a mental process in which the subject uses its mind to create an internally simulation, for a movement, which results without corresponding motor output only his mind for learning a physical process. These kinds of learn processes are often called trail-by-trail improvement in motor performance. Furthermore, neurophysiological studies have reported that mentally simulated and physically executed movements trigger similar motor representations and share overlapping neural substrates [28]. Several experiments and clinical evidences show that mental training with motor imagery can improve motor performance of subjects.
The evaluation of an experiment in which probands have to train their movement skills with motor imagery shows that the learning process with mental training (MIT) was nearly as good as physical training (PT). For the experiment procedure a test group of forty subjects was divided into 4 groups: The first group uses conventional physical training, the second uses mental imagery, the third group was and active control group and the last group was an passive control group. The aim of the experiment was to point at numbers of an experimental device designed for this test. Two horizontal parallel aluminum dowels (length 75 cm, diameter 1 cm (Figure 3.3)) are placed on a vertical bar in a distance of 44 cm, each with 4 switches, numbered with numbers one to eight. All participants have to point these switches in an arbitrary sequence given by an instructor. The experiment shows, that the PT group and the MIT group show improved their motor skill performance. The arm movement were faster and straighter immediately after the training session, additionally the quality of the movement was enhanced by booth groups. The team deploy the hypothesis, that the learning rate during mental imagery training is smaller than during physical training. Due to the fact, that in the same amount of training trails, the PT group was slightly better than the MIT group.
This finding after the experiments was, that motor-imagery training induced a true form of motor learning, similar to physical training and not a transient change in motor performance. [40]

3.2 Mental Imagery for Robot Learning

Todays tasks for robots become more and more complex, they have to solve tasks precise, efficiently and faster. The accuracy to solve this actions depends on the level of accuracy of its internal representations. Thus, the real world is to complex and it is therefore extremely hard to collect and store all needed data, to develop a solution. The role of cognitive control integrated with concepts of motor imagery might offer an improved approach to increase the capabilities of robots. [20].
The ideas to represent mental imagery as motor imagery on robots are given by a combination of two different neural

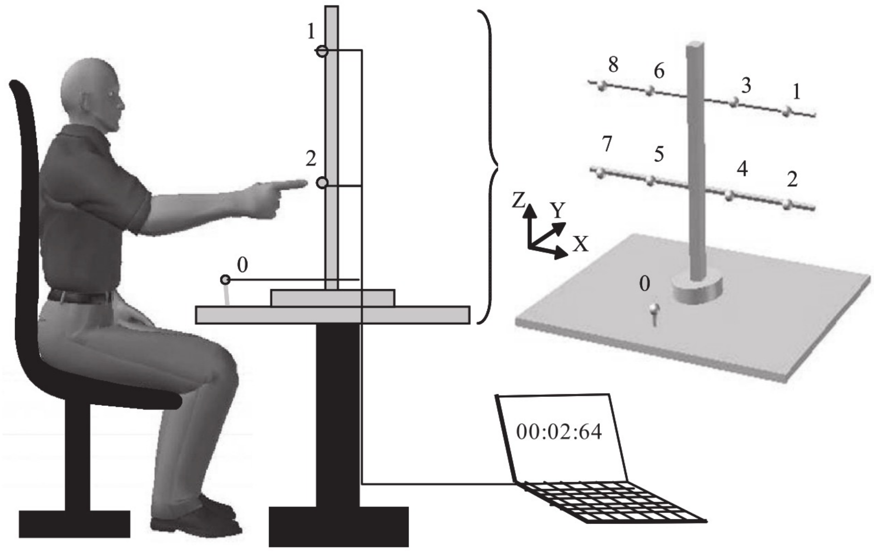


Figure 3.3.: Experimental device used for the experiment. The device consisted of a vertical bar, 2 parallel aluminum dowels fixed on the bar, and 9 switches (T0 ... T8) fixed on the dowels and connected to a chronometer integrated into a laptop[40].

networks. A feed forward neural network that controls the robot by solving its task, and a dual recurrent network that generates data for motor imagery to improve the feed forward network. (For further information the appendix describes these two networks. A) Since neural networks are inspired by the way neurons work in the human brain this is trivial approach. As described in (Section 2.3) this thesis will focus on mixture of experts to implement motor imagery, to understand in detail how theses algorithms perform on robots.

3.2.1 Existing Frameworks

The idea of mental imagery, performed as computational operation is not new, actually it is a very old approach, 1977 Kosslyn et. al. [30] introduced concepts and implementations, how to perform visual mental imagery on humans. Since then new frameworks which use bio-inspired concept are published. One of them is TRoPICALS.

3.2.2 TRoPICALS

TRoPICALS is a computational model of object affordance, which is designed to account an action-language and stimulus-response with compatibility effects, studied in cognitive psychology The abbreviation TRoPICALS relays on theses principles "Two route, prefrontal Instruction, Competition of affordance, language simulation. Theses principles derive the four brain organisation principles incorporated in this architecture.[20]

1. The two route organisation of the sensorimotor brain into the ventral and dorsal neural pathways
2. The action selections based on prefrontal cortex "instructions"
3. The selection of actions in the premotor cortex based on the competition between different affordances with bias from prefrontal cortex
4. The capability of language to trigger internal simulations of the referents of words

The model is used to reproduce the compatibility effect, which is linked to stimulus-response, as agreement or disagreement, for the top-down pre-frontal cortex bias, which is used to produce slow or fast reactions times. TRoPICALS consists of several different parts, which are grouped into two pathways: dorsal and ventral. In the scheme 3.4, the brain dorsal pathway takes information localisation informations, to answer the "where" question while the ventral collects data, to answer the "what" question of the given task.

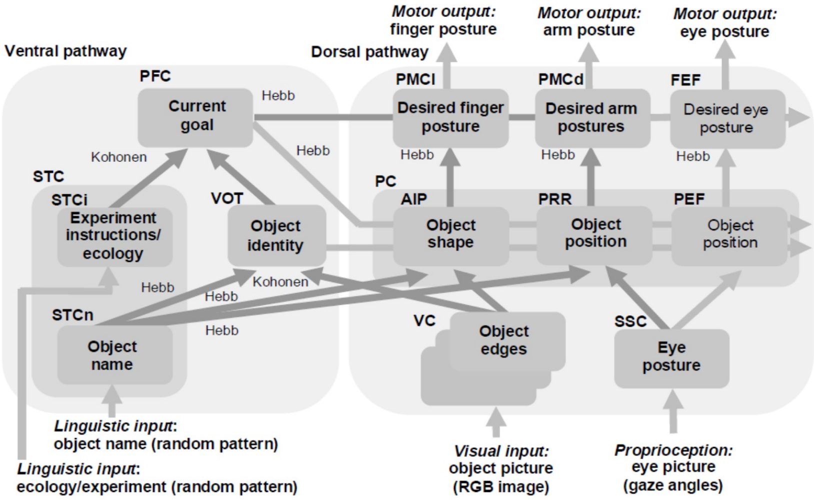


Figure 3.4.: The ventral pathway consists of Prefrontal Cortex (PFC), Superior Temporal Cortex (STC) and Ventral Occipito-Temporal Cortex (VOT). The dorsal pathway consists of Premotor cortex (PMC), Parietal Cortex (PC), Visual Cortex (VC) and Somatosensory Cortex (SSC). Both pathways are linked [20].

In the concept of TRoPICALS, the PFC is responsible for the high level cognitive ability for example decision making, while the PMC is responsible for generating motor outputs. The VC is used, to extract abstract information from the images that are precieved.
There are three types connections between booth pathways, hand coded connections, Hebbian, and Kohonen. Hand coded connections follow predefined wights while Hebbian and Kohonen, need competitive learning.
[20]

3.2.3 Population Coding

Based on TRoPICALS, neural maps on the basis of concept of population codes represents the stored information. Population codes are inspired by the visual cortex, encodes many perceived stimuli like orientations, colours, directions, through clusters of cells. So the Information will be encoded by artificial neurons, which work together in clusters. [20]

3.2.4 Hebbian

Hebbian is as supervised Learning Algorithm, the principle of Hebbian was inspired by the famous quote "the cell that fire together wire together" published by Donald Hebb [20]. The idea behind this learning method focus on the existence of connections, synapses between the neurons. By connecting two neurons the post-synaptic neuron is only able to fire , if the activation of the is strong enough and the multiplication of the current action potential meets a proper constraint, by multiplication. Then the result is that the neuron will be fired.
The following equations will show how the the weights are updates and changes
The first equation 3.1, called Oja rule, solves the problem of the basic Hebb rule, by causing a weight growing that is not bounded

$$\Delta w_{ij} = \eta \alpha_i(\alpha_j - w_{ij}) \tag{3.1}$$

$$w(t) = w(t-1)_{ij} + \Delta w_{ij} \tag{3.2}$$

The second equation 3.2 defines how the weights changes from neuron i to neuron j, additional 3.1 denotes the changes in a_i, and a_j [20].

3.3 Existing Experiments in Mental Imagery

The Ph.D. Thesis "Mental Imagery in humanoid Robots" by Kristsana Seepanomwan describes how mental robot imagery can solve, typical mental imagery problems like mental rotation. The model of mental rotation, is linked to a macro architecture, which behaves like the brain areas used in mental imagery. The used model is departed from the TRoPICALS framework, within the " computational embodied neuroscience framework, aiming to link embodied cognition and behavior and the brain system mechanics which underlying them.

3.3.1 The Experiment

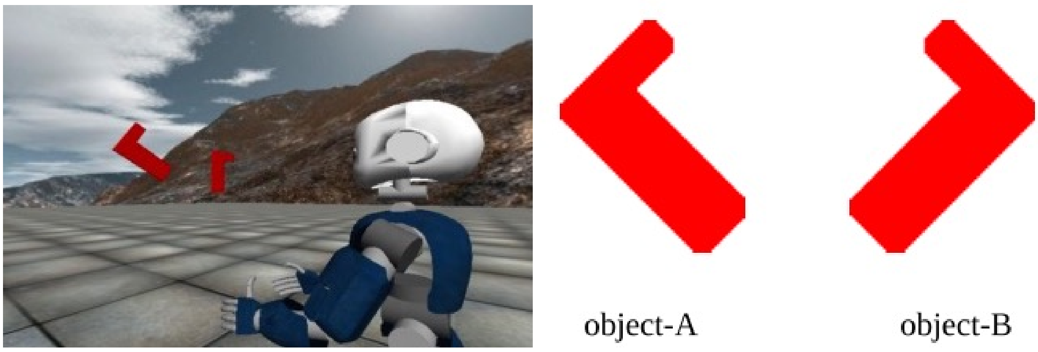
The experiment tries to reproduce the concept how, parital and premotor areas might by involved in mental rotation, by investigating a neural operational hypothesis. For this a simple forward model was used, which divides the process in four sub-tasks.

1. Stimulus encoding, generation of mental images
2. Planning and execution processes of mental rotation
3. Comparison of the stimuli
4. Execution of the response

Especially the TRoPICALS framework is useful for all these processes, it supports the generation mental images by reproducing key features of the parital-premotor cortex. For comparison and execution of the response. In this respect, to address the core mental rotation process, to solve the planning and execution processes, TRoPICALS need to develop some new key features. At First it need premotor-parital feedback loops, that allow mental rotation and sensory prediction based on forward models. And second The visual and motor system has to be scaled in and 3D environment for realistic purposes[20].

3.3.2 Mental Rotation using the iCub simulator

For experiments the iCub simulator was used 3.5a. The iCub arm has 16 joint, for the experiment only joint number 5 was used, which directly affects the robot wrist's angle. The robot holds the given object 3.5b in its right hand. The rotation of joint number 5 was chosen to only change the orientation in the object plane. By varying the object by 15 degrees (0, 15, 30, 45, 60, 75) the robot decides if the robot is rotated or mirrored. Here the robot does not interact with physical objects, only with simulated shapes [20].



(a) The iCub simulator in its mental environment, while it solves the task of mental rotation.

(b) Graphic of the used object that the iCub robot was able to recognize during the experiment .

Figure 3.5.: Simulation environment and used object for the experiment of mental rotation [20].

3.3.3 Architecture

The used architecture represents an operational assumption, on how mental rotation could be implemented with visual and motor neural processes. The model relies on 3 specific brain areas, the parital cortex (PC), the premotor cortex (PMC), and the prefrontal cortex (PFC). All of them are represented by distinct neural maps which use population code. In this experiment that neurons of the PC map, consists out of 32 by 32 neurons which are able to encode the shape and orientation of the mental rotated object. The PMC consist of two 2 neural maps, PMC_1 and PMC_2 which encode different arm parts by resolutions of 31 by 105 neurons and 10 by 20 neurons. The Figure of 3.6 describes how these brain areas are connected and how they are learnable[20].

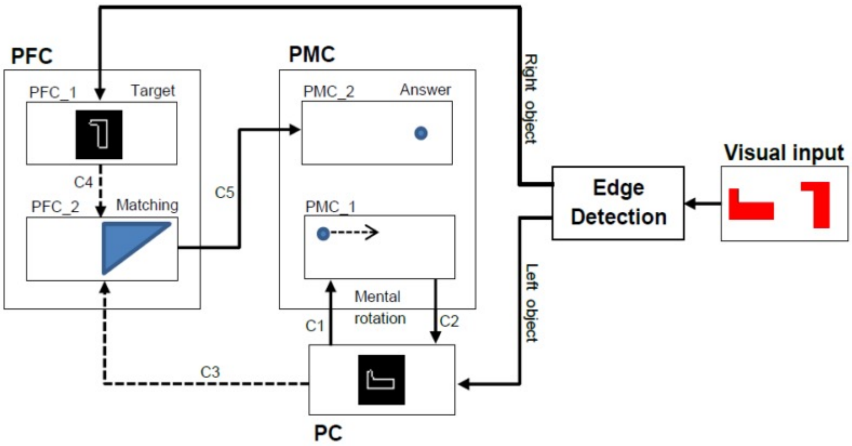


Figure 3.6.: The model of mental image rotation. Each box represents the model's components. The arrows defines the information flow between the different components [20].

4 Concepts

As described in the previous two chapters concepts of mental imagery have previously been applied to robotic tasks. Hereby, most existing approaches upon neural networks. Due to the fact that neural networks are inspired by neuropsychology to work like the visual cortex. This thesis introduces a different concept to transfer mental training to a mixture of experts setting. Eventhough theoretically the single experts could still be neural networks, here we deploy movement primitives, as they are a common skill representation in motor skill learning.

4.1 Concept for Neuropsychological Movement Creation

Concepts of neuroscience are needed, before developing concepts for robotics. The following concept is an approach which tries describes how motor imagery could operate in the human brain for movement learning.

This subsection summarizes the insights gained from literature review in Section 2 and presents a possible concept for biological motor skill learning. In particular, this concept distinguishes between activation of different components during mental and physical training. This concept provides the first step towards the transfer of mental training to robot motor skill learning.

4.1.1 Brain Regions used for Mental Imagery

In order to be able to create mental images a cooperation of several brain regions is needed. Identifying of the used brain regions is possible by using the functional magnetic resonance imaging (fMRI) technique. During mental imagery the in Figure 4.1 described regions show clear activation's. The functions of the brain regions for mental imagery can be determinated, by identifying problems in the usage of mental imagery, if a possible related brain region is injured. This can be seen in bad mental imagery performances. For example motor imagery prediction relays on computations of the posterior parietal cortex [23] Clinical observation of patients with posterior parietal cortex damage, indicate that this brain region is critical for sensorimotor integration and prediction [24].

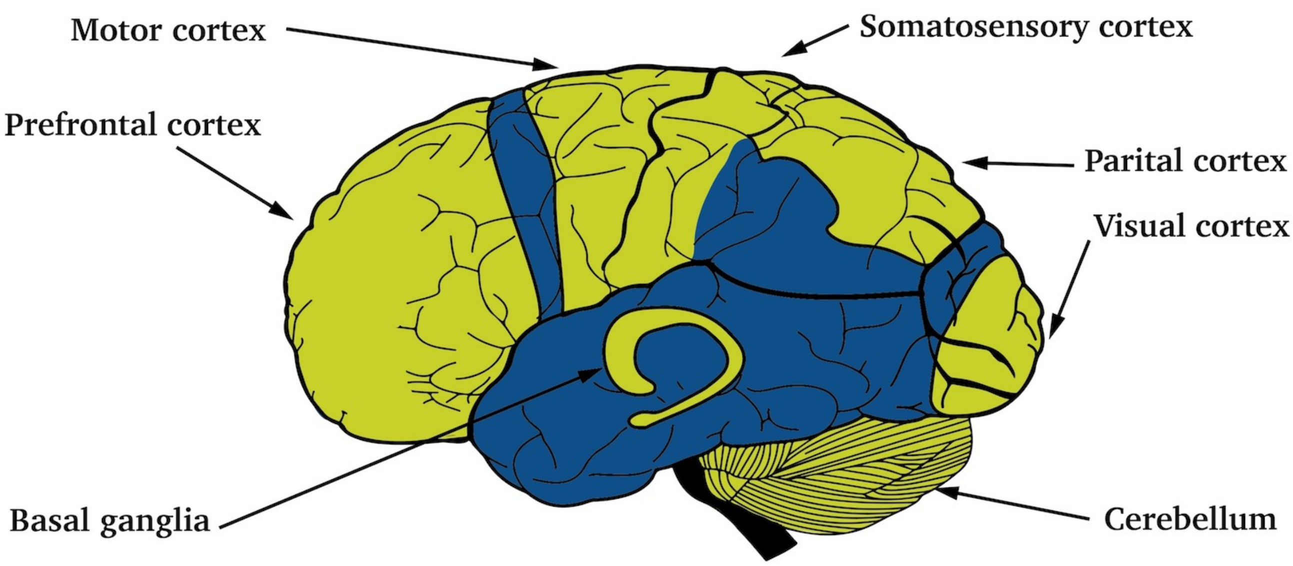


Figure 4.1.: Graphical visualisation of the different brain regions used for motor imagery (in green) and physical Training. Including Frontal lobe, Parital lobe, Occipital lobe, Temporal lobe, Cerebellum.

The knowledge how a specific movement is generated and learned, and how different brain regions influence this process is in the actual state of research not defined. In contrast to this, the specific function of each brain region related motor skill is well known. Hypothetical conclusions are possible, based on the specific knowledge of these different regions.
The following concept builds upon the literature review in Section 4.2. The real cause of movement creation, is determinated in the a high-level planning area, which consists of the basal ganglia and cerebellum. By performing an interplay between these two regions the basal ganglia creates trajectory information, some of these trajectories can be defined as meaningful, others as senseless. To select the meaningful movements the cerebellum performs a kind of sanity check, this process runs in a loop between this two regions, until a usable trajectory is generated. Subsequent this information will processed by the Motor area. By preprocessing the given trajectory in the premotor cortex, the information are conditioned, to be processed in primary motor cortex (PMC) and supplementary motor cortex (SMA). Those two regions, perform, as described in Figure 2, a mapping process of the trajectory information to the body.
At this stage two different kinds of execution are possible, a physical performed movement or an mental performed movement.

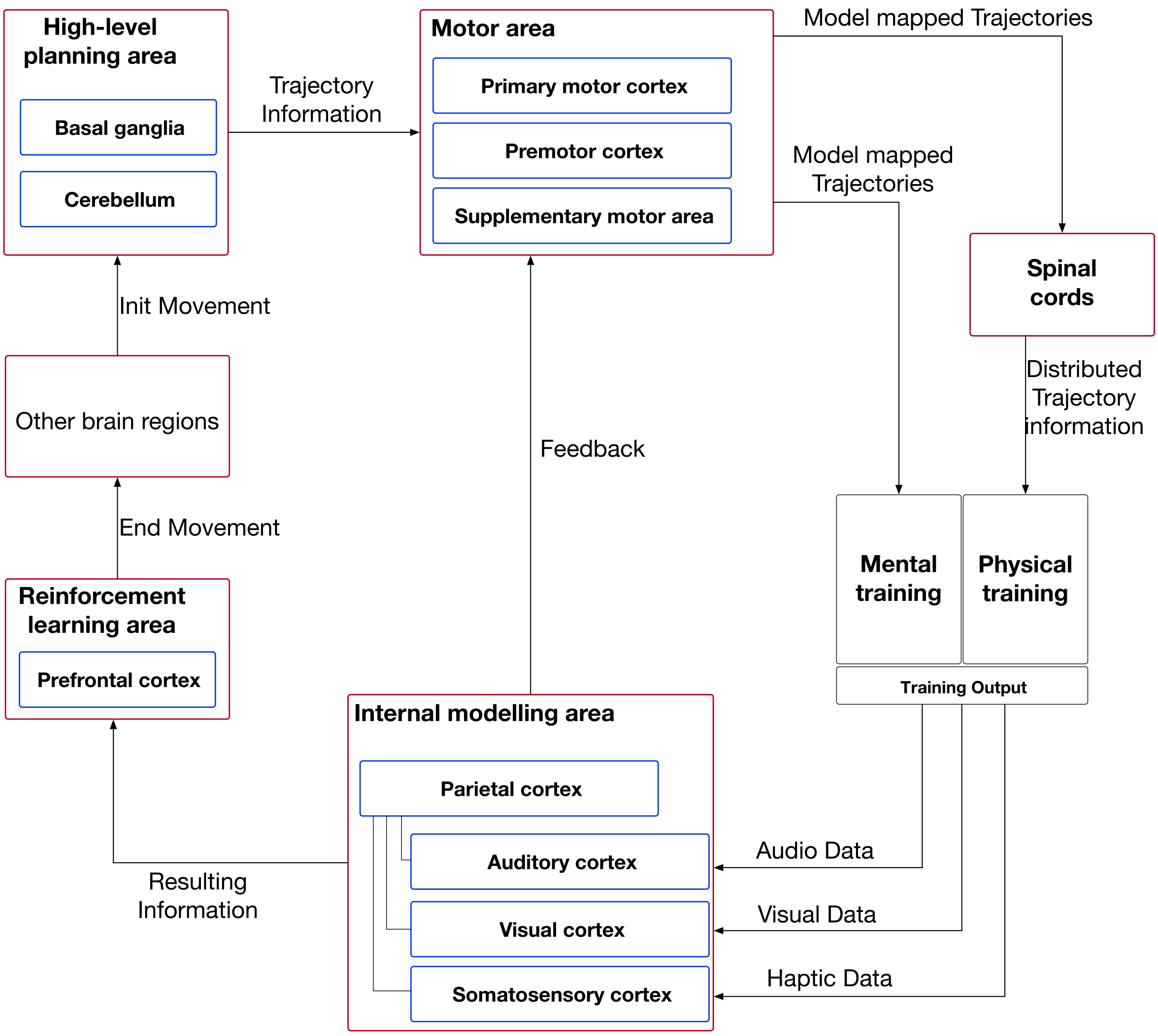


Figure 4.2.: Motor skill learning include many brain Regions. Basal ganglia, cerebellum, primary motor cortex, premotor cortex, supplementary motor are, parietal cortex, visual cortex, auditory cortex, somatosensory cortex and prefrontal cortex are the primary brain regions engaged in skill learning and development.

Physical Movement: For performing the task physical, the mapped trajectories are send to the spinal cords, which supply the body parts with the needed information. The movement will be performed in a physical training environment, while performing the subject collects data. These data is captured by auditory, visual and somatosensory cortex. This is done in an internal modeling area, which consists of these three cortexes and the parital cortex. Regulation is needed to perform the movement precise, this is done by the parital cortex, which works as an internal movement controller. A generated feedback, is processed by the motor area, to correct the movement, until the task is finished. Afterward a reward is generated based on the success of the planed movement. A reinforcement learning area, which consists of the prefrontal cortex is dependent to this task.

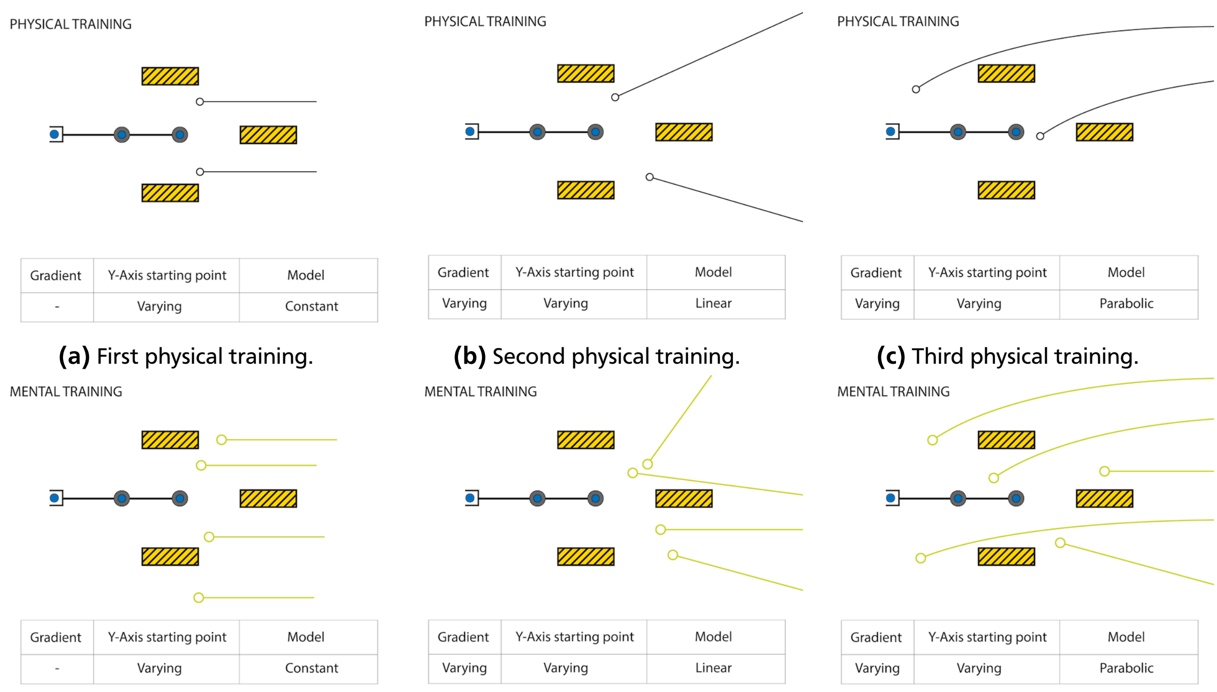
Mental Movement: A mental performed movement is spinal cord independent. The movement can only performed in a mental training environment, which is generated by the visual cortex suggests current research. For better understanding this "Mental training" is placed beside "Physical Training" in Figure 4.2 for better comparison between booth paths of movement execution. The mental performed movement underlies the same controller and data capturing cycle as the physical movement, only different is the capturing process. Due to the fact, that a mental training environment could not represent a real world feedback, the working memory is used to generated information about the task. The working memory is related to the prefrontal cortex, which organizes these information. Equally to the physical training a reward is calculated. After completing the reward generation the movement process ends.

4.2 Concepts for Bio-inspired Motor Skill Learning with Mental Training

The concept of mental imagery on robots is capable of faster and safer learning. The concept can easily transferred to other algorithms. Especially in situations in which a robot learns its own parameters and model. In this case the concept of mental imagery can be seen as an add-on framework, which is based on data caption and recreation. So the learned model can be used for data reconstruction and training.

Principle

Mental Imagery should be able to generate training data from learned models. This is illustrated in the example Figure 4.3. In this example a two degree of freedom robot should perform a ball catching task. To simplify this example, the incoming ball use very simple dynamics. The Training starts in a phase of physical training. At the beginning two training balls are incoming that share the same gradient behavior, gradient equals zero. So the approach captures this behavior, additional different Y-Axis starting points are captured. The following mental imagery phase, uses the model learned in physical training, so several new "mental" balls can be thrown to train the robot.
This gives the robot the possibility to optimize the gating network which controls the different DMP.



(a) First physical training. **(b)** Second physical training. **(c)** Third physical training.

(d) First mental training. **(e)** Second mental training **(f)** Third mental training.

Figure 4.3.: The dynamical model enhances over time, based on the captured dynamics of the physical training. First column (4.3a , 4.3d) represent the physical and mental training based on a linear model without gradient. Second column (4.3b , 4.3e) represent the physical and mental training based on a linear model with gradient. Third column (4.3c , 4.3f) represent the physical and mental training based on a parabolic model.

4.2.1 Training Pipeline for Experiments

The following pipeline will present a training scenario concept in which a motor imagery training is performed. Based on mental imagery many other training concepts are possible, this concept was chosen because of its strong connection to motor skill training.

4.2.2 Ball Catching Task with Motor Imagery

In this example the robot has to catch incoming balls, which are thrown in physical training. By doing this the robot has to optimize two different models: its own and the model of the ball prediction. To simplify the used concept of the mental inspired ball-catching experiment, several parts are grouped to sub-tasks.

Training pipeline

As shown in Figure 4.4, a training task starts with a start symbol, which sets the parameters $i, i = 0$. To learn an initial model a physical training needs to be performed. By throwing test balls, the robot generates a generalized model, which is comparable with the working memory of humans. This is done until the hyper-parameter η is reached by i, physical training ends. Based on the captured Information the mental imagery training starts to develop a simulation to train the robot in an mental environment. The motor imagery training is performed until j equals χ. A control cycle of several training tasks of this is useful, because the mental learned information need validation.

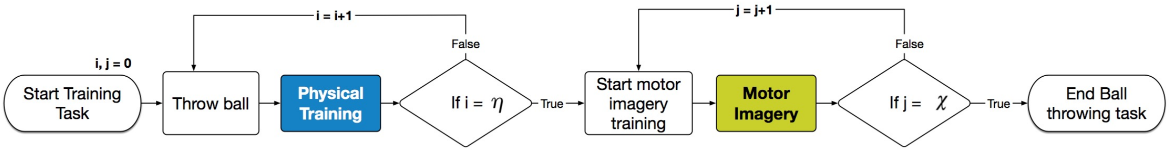


Figure 4.4.: Simple pipeline of physical training coupled with motor imagery training. η and χ are used for counting physical and mental training iterations. The control flow of physical training is defined in Figure 4.6 , motor imagery is defined in Figure 4.5.

Motor imagery training

The motor imagery training (Figure 4.5) starts with defining a new motor imagery task that should be performed. Secondly a imagined ball is generated, by using the collected information of the captured dynamics of the given physical samples. Then the Plan trajectory process begins. By using the MoMP a simulated catch action will be performed, additional the simulated catch action is used to update the weights of the used MoMP and to collect the reward.

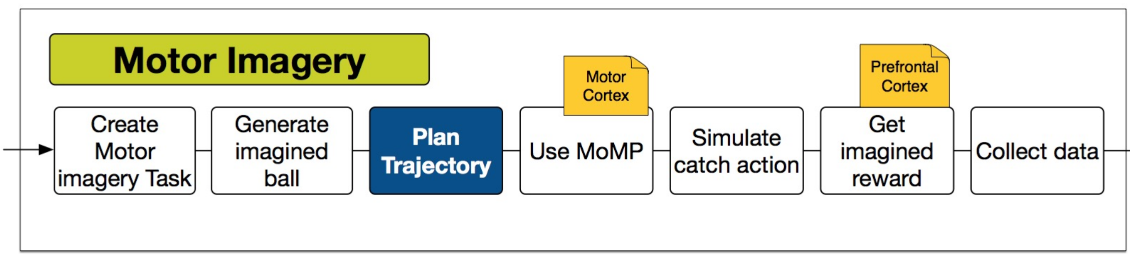


Figure 4.5.: Pipeline of a mental training iteration, the iteration could be handled as a "Task" in a black-box concept. A Task is based on previous collected information of the ball Trajectory, on which a new ball with modified information can be thrown to train the robot, using the existing ball model.

Physical Training

The physical training (Figure 4.6) is not very different to motor imagery training. First a new task is created, then a real ball is thrown. Afterwards the MoMP approach is used to perform a catch action. In this case a real reward is captured. A weighting between a real and imagined reward can be useful in this approach.

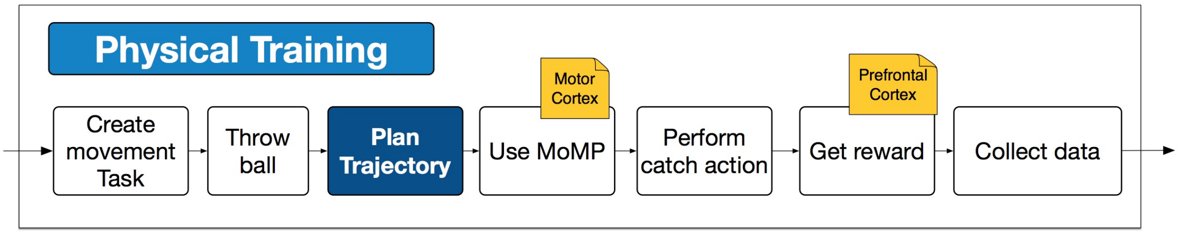


Figure 4.6.: Physical training is not very different from mental training the significant, difference is a new input of information from the real world, instead of generating new "pseudo" information of the model.

Trajectory planing

The trajectory planning process (Figure 4.7) begins with the perception of the ball position, by capturing the next position a model can be learned. This process runs in a loop until the model is nearly equal to the ball trajectory. Secondly this model is used to estimate the position on which the robot should catch the ball, if this position is predicted, the catch-trajectory can be computed.

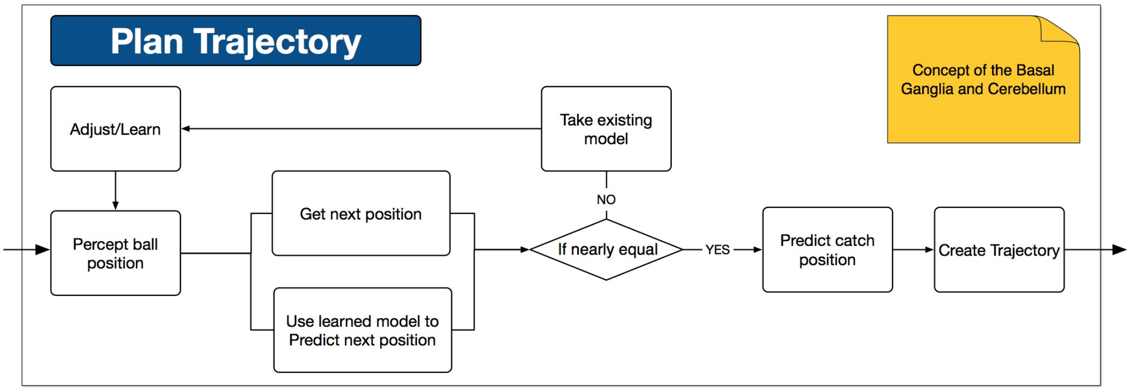


Figure 4.7.: The concept of trajectory planning during motor imagery.

Parameter Updates in the Motor Imagery Concept

During motor imagery only the parameters of the used robot can be updated. In difference to physical training, mental created ball trajectories are not designed to give new model information. By defining variances or specific noise an artificial creation of new data for ball trajectories is possible. This means, that the model of the ball is, in this concept, defined as fixed during mental training. So only the parameters of the MoMP can be updated, during a motor imagery training session.

4.2.3 Simulation in Comparison with Motor Imagery

To define the differences between a simulation and motor imagery, the concrete definition of booth should be clear. First, a simulation does not follow a clear definition, a common definition is, that a simulation is the imitation of the operation of a real-world process or system over time. Simulation involves the generation of an artificial history of the system, and

the observation of that artificial history to draw inferences concerning the operating characteristics of the real system that is represented [25]. Which sounds very similar to motor imagery , due to the fact that both imitate the real world, generate an artificial history, and follow the characteristics of the real world. But in detail, there are some differences. Motor imagery is does also not follow a concrete definition in this case, we define it as: "A simulation processes based on an initial model, which is filled over time with data, via a data capturing and data recreation process, to perform and optimize actions and the model over runtime". This makes clear that a simulation is different to motor imagery. Especially the development of the model and the concept of data capturing and recreation, show huge differences. In further sense motor motor imagery enhances the concept of an simulation, due to its adaption capabilities to data.
Differences can be seen in the in the pipeline which is matchable for booth concepts 4.8. Usually a the concept of a simulation is defined by. Build in a model, which is an abstraction of the real world; make a discretization of the model; Implement the algorithm for this discrete model; visualise the results; Validate the results with the real world or test data.

Term	Simulation	Motor imagery
Modelling	Static model, fixed during simulation	Initial dynamic model, that develops over time
Discretization	Fixed on previous knowledge	Fixed on previous knowledge
Implementation	Fixed	Fixed
Visualization	Fixed	Not needed, only for updating parameters
Validation	Comparison with real world or knowledge	Comparison with physical training or existing knowledge

Figure 4.8.: The pipeline of simulation defined in [26] can be used to describe differences between Simulation and motor imagery.

5 Experiments

Possibly motor imagery and mental training concepts can enhance existing skill learning approaches for robots. In this Section we show how to apply the single components of the concepts introduced in section 4 for a robot ball catching task. First, Section 5.1 shows results for learning a model of a ball out of real trajectories by using linear regression. Next, Section 5.2 presents results on function approximation with the Mixture of Experts approach. Finally, Section 5.3 evaluates dynamic movement primitives as motion generators for a two link robot arm and discusses possible future extensions to mixture of movement primitives.

5.1 Learning and Generating Trajectories

To implement the developed concept of data capturing and data recreation two key features are important: Learn ball trajectories and generate ball trajectories for mental training. This section present regression as solution for booth problems.

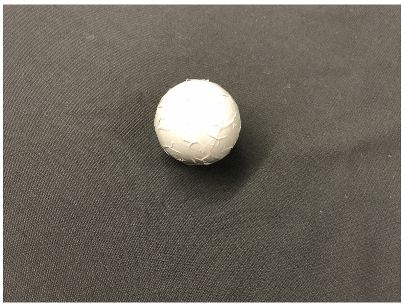
5.1.1 Regression for Ball Trajectories

For prediction, regression can be a very powerful tool to learn data easily and fast, especially if model knowledge exists. In this case regression is used to estimate the position on which the robot should catch the thrown ball. Hereby, we take advantage of model knowledge of the parabola.

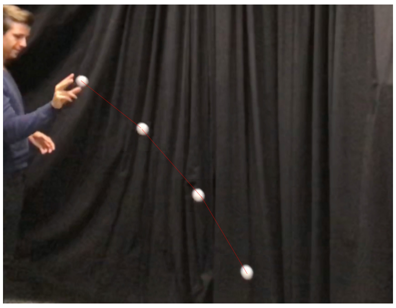
Experimental Set-Up

To test the regression capabilities real data of a ball trajectory is needed. So the data of the ball is captured by the cameras of the OptiTrack Motive software (Figure 5.1a), a ball primed with special OptiTrack markers (Figure 5.1b) is thrown in a 3D cage. The software captures the x-y-z coordinates of each time step. The illustration of the captured trajectory is given by (Figure 5.1c).

(a) Cameras of the OptiTrack system.



(b) Ball with OptiTrack markers.

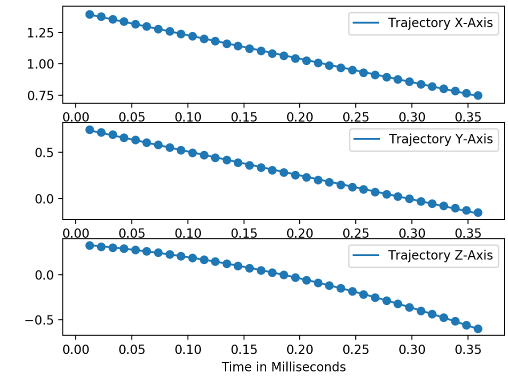


(c) An example for a physical thrown ball trajectory .

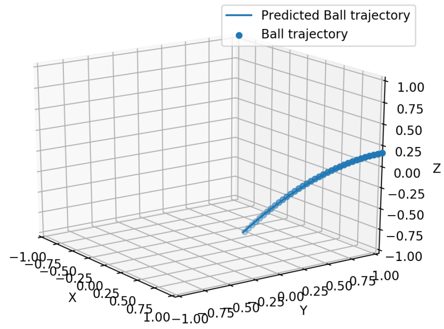
Figure 5.1.: Setup of the experiment using the OptiTrack system and OptiTrack markers.

Setting A

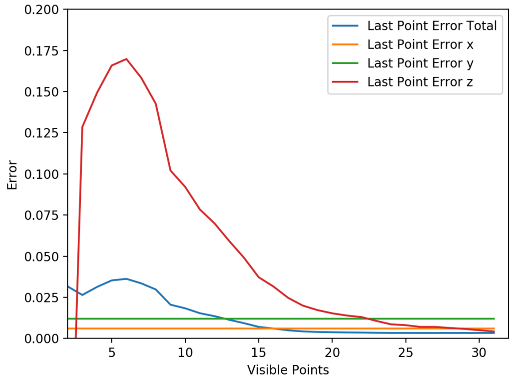
The Figures 5.2c 5.2d show that linear regression on the X-Y-Axis and polynomial regression in the Z-Axis show good results. The noise in this case is manually added, because Especially Figure 5.2b shows, that knowledge of only 15 % show precise and accurate results, to predict the end position Figure 5.2a .In which the variance of x, y, z is: 7.2518e-06, 2.775e-05, 2.538e-06. As the last point error of Figure 5.2c shows that regression can predict an accurate catch position with ≈ 55 % of the data. Due to the fact, that the movement in X-Y coordinates is defined as linear this model knowledge, seems to improve the results of regression the polynomial regressions produces in comparison more errors



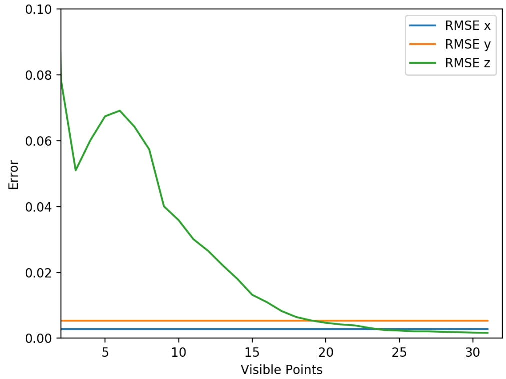
(a) Shows the graphical regression of each Axis.



(b) Shows the regression in 3 Dimensions.



(c) Last point error, from for visible points of the trajectory.

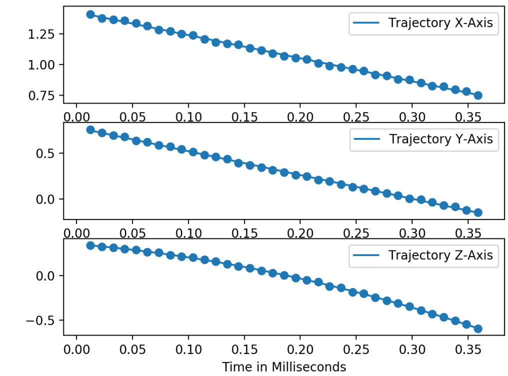


(d) Shows the RMSE for a number of visible points.

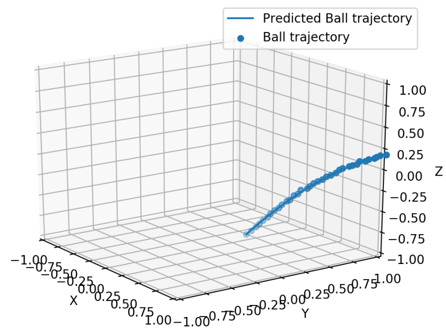
Figure 5.2.: Experiment of Setting A, with capturing noise

Setting B

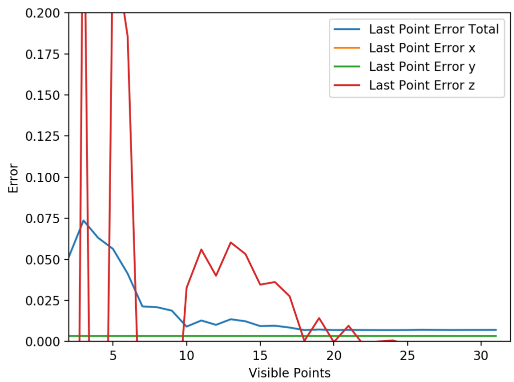
To test how good regression predict the true end position to the given trajectory a noise of factor 0,02 was added. IN comparison to the previous experiment with Setting A the error level drops at the same number of visible points ≈ 18 Points 5.2c 5.7c. As Figure 5.7a and 5.7b show, this noise is very strong and represents an upper bound of noise in this test scenario. The variance is in this case increased, the variance of x,y,z is: 3.6989e-05, 4.8476e-05, 3.9134e-05.



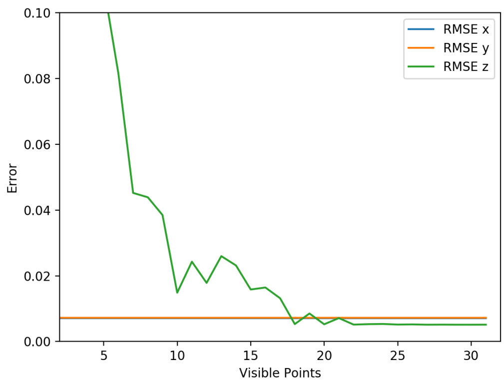
(a) Shows the graphical regression of each Axis.



(b) Shows the regression in 3 Dimensions.



(c) Last point error, from for visible points of the trajectory.

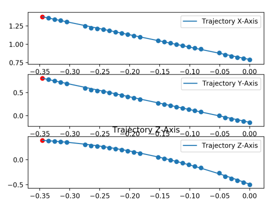


(d) Shows the RMSE for a number of visible points.

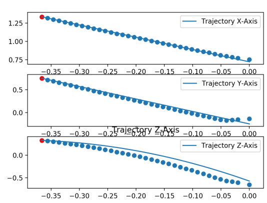
Figure 5.3.: Experiment of Setting B, with capturing noise and additional Noise = 0,02.

Setting C

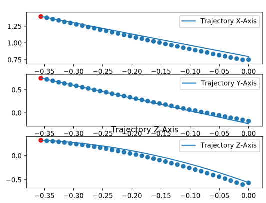
For the prediction of different ball trajectories, the learned model can be used. Figure 5.4 shows the same learned regression model predicting different trajectories.



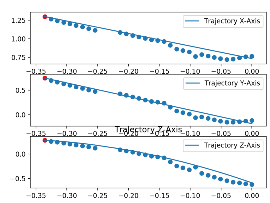
(a) Prediction of ball trajectory 1.



(b) Prediction of ball trajectory 2.



(c) Prediction of ball trajectory 3.



(d) Prediction of ball trajectory 4.

Figure 5.4.: Prediction for different trajectories with the same learned model

5.1.2 Regression in Motor Imagery

Classical regression is a tool, which can used to perform mental imagery. In this example regression is used to capture the dynamics of a physical training, during a physical training session, then the capture data can be used to recreate new data, to throw imagined new balls, based on the model of captured information.

Workflow

In a given tasks the a robot learns its own motor skills using balls that should be captured. The training is divided into physical training, followed by a phase of motor imagery, afterwards a new training can begin.
During physical training regression learns the parameters of the ball parabola, so coefficients and starting point are captured and saved in the model. After the training end the process of data recreation begins. A small modification of the adapted parameters can be used to create new imagined ball trajectories, that are able to train the robot (Figure 5.5).

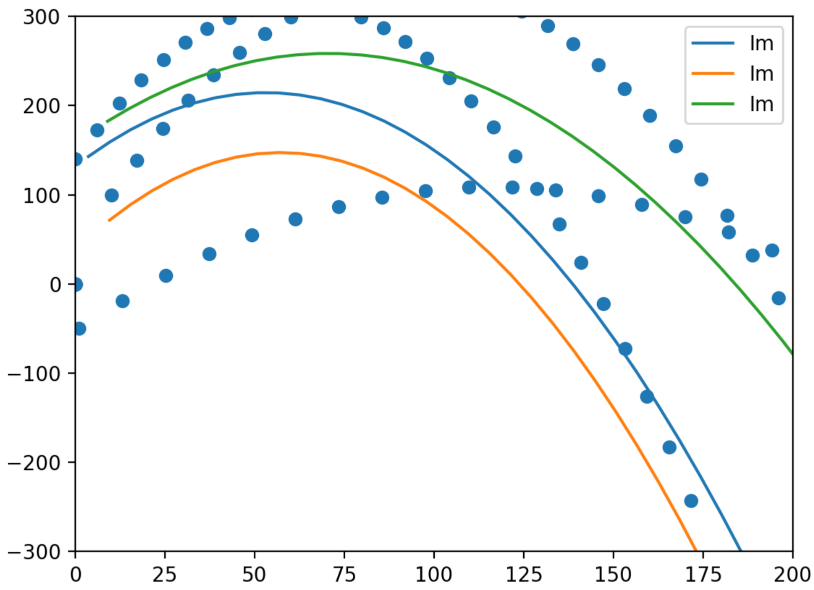


Figure 5.5.: Imagined ball trajectories based on the captured dynamics during physical training.

5.2 Mixture of Experts

Mixture of Experts is a Machine Learning technique that basically follows the divide and conquer principle. A fixed number of, in this case linear, features, which are called experts, adapt to the given data, for creating a Gaussian mixture which approximate the given data. Knowledge of the data can very important, due to the fact that useful initialization improve the algorithm very effective.

5.2.1 Naive Initialization for Variable Number of Experts

By using a variable number of experts several initialization adjustments are important, such as the correct initialization of prior and features. As described in 2.13 the sum of prior should be equal to one, as approach one was simply divided by the number of experts k

$$\alpha = \frac{1}{k} \tag{5.1}$$

To calculate the feature Matrix $\mathbf{\Omega}$ with linear features of $\varphi = [1, x]$ the input sequence was divided into $k+2$, sections to place a bias from $1, 2, ..k-1$. The gradient m is calculated by $\frac{y_i + y_{i+1}}{x_i + x_{i+1}}$ for each value between the sections. For approximating the intercept b the approach uses the linear equation $y = m \cdot x + b$ in which x is the y value of the section. By calculating

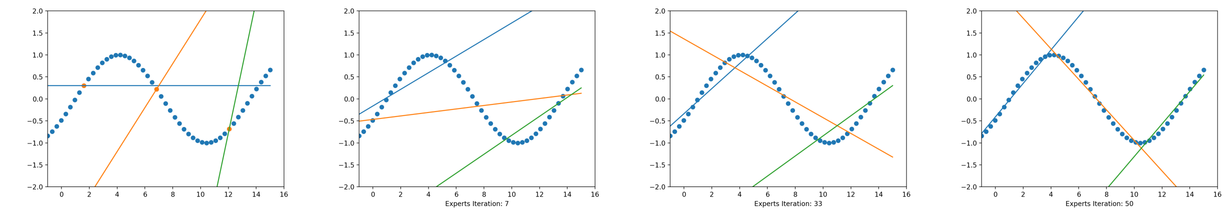
$$b = y - m \cdot x \tag{5.2}$$

By repeating this for each expert a useful expert initialization will be generated.

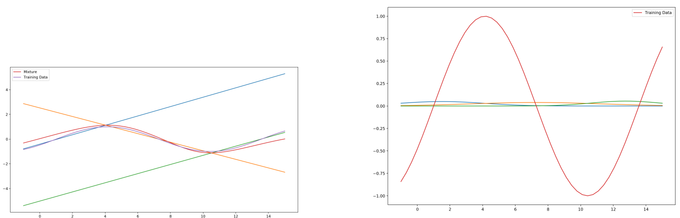
5.2.2 Experiments with a Trigonometric Functions and Linear Experts

In this given case the data is generated by a trigonometric function. For learning data, which is based on trigonometric functions like sinus and co-sinus, the number of experts can be easily chosen by counting the number of increasing and decreasing parts of the data, if data follows this as known principle. So three linear experts will adapt to this data for creating the Gaussian mixture. The initial situation is given by Figure 5.6a after 7 iterations (Figure 5.6b) 2 experts (blue and green) begin to adapt to proper parts of the data, the orange experts explore its environment to enhance its likelihood. after 33 iterations all experts begin to converge to meaningful part of the data (Figure 5.6c). After 50 Iteration the experts reach their optimal position, which is here bounded by a fixed variance of 0.34 (Figure5.6d). The resulting mixture is able approximate the given data (Figure5.6e), the experts are aligned and developed clear dependencies to the provided part of the data (Figure 5.6f).
The error function which is represented by equation 5.3, by using the error function the reduction of the error in the mixture over time can be seen (Figure5.6h). Especially the dump of the error at the 5 to 6 Iteration, make clear that the experts can show an exploration behavior, although the algorithm is designed to converge.

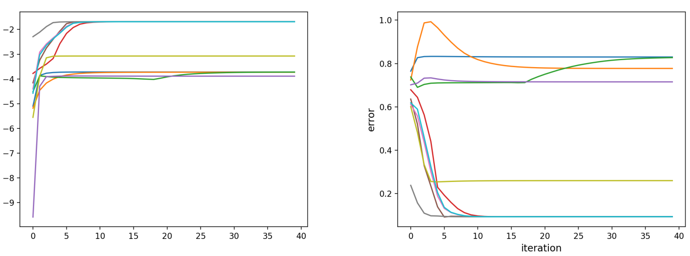
$$\epsilon = \sqrt{\frac{1}{n}\sum_{1}^{n}(y_{\mathrm{GM}} - y_{\mathrm{des}})^2} \tag{5.3}$$



(a) Initial situation for mixture of experts. **(b)** MoE after 7 Iterations. **(c)** MoE after 33 Iterations. **(d)** MoE after 50 Iterations.



(e) Resulting conditional mixture which approximates the training data. **(f)** Single Gaussians, which learned their dependencies.

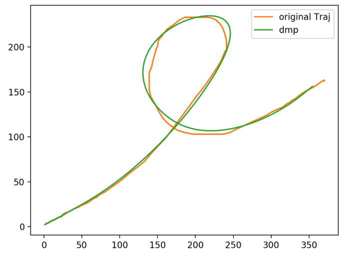


(g) Log-likelihood. **(h)** RMSE for each iteration.

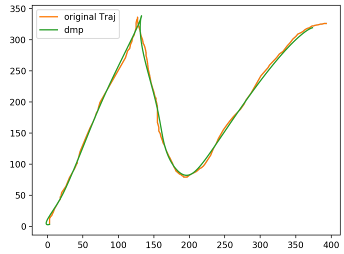
Figure 5.6.: Pictures a) to d) show the adaption procedure of the experts. Resulting is a conditional mixture e). The dependencies for each expert are seen in Figure f). The Log-likelihood for different initialization the iterations is shown in g). The error for different initialization can be seen in h)

5.3 Using Dynamical Movement Primitives with Referred Trajectories

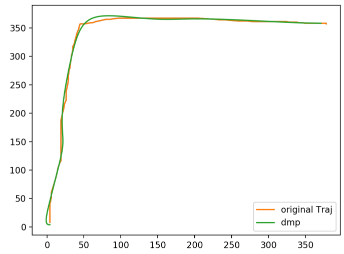
Dynamic Movement primitives are ideal for adapting different characteristics of data. For this the forcing term f is learned by a desired trajectory, which characteristics define the produced DMP.



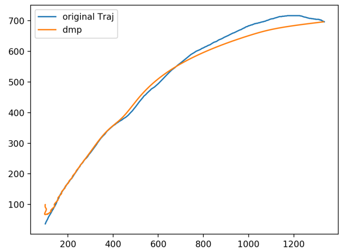
(a) Hand-drawn Example 1
RMSE = 15.7689



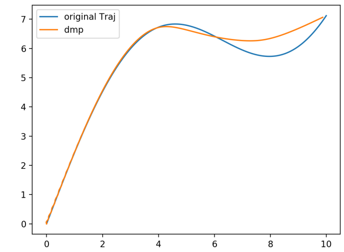
(b) Hand-drawn Example 2
RMSE = 21.5827



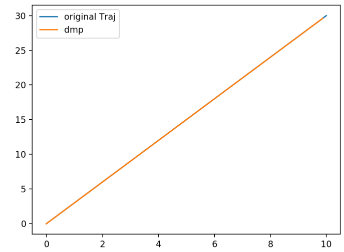
(c) Hand-drawn Example 3
RMSE = 16.1197



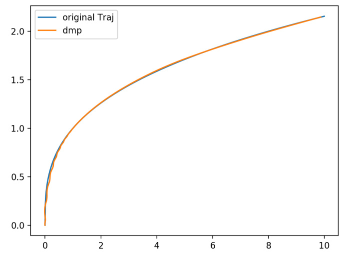
(d) Hand-drawn Example 4
RMSE = 16,4979



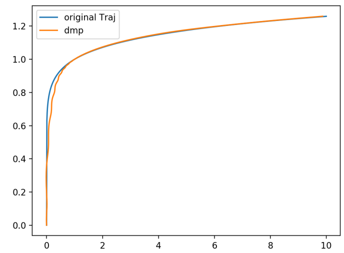
(e) DMP imitating $3\sin(\frac{t}{2}) + t$
RMSE = 0,4821



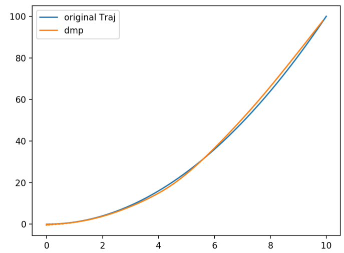
(f) DMP imitating $3t$
RMSE = 1,3012



(g) DMP imitating $\sqrt[3]{t}$
RMSE = 0,083



(h) DMP imitating $\sqrt[10]{t}$
RMSE = 0,0971



(i) DMP imitating t^2
RMSE = 7,6924

Figure 5.7.: Examples of DMP which imitates a desired path.5.7a, 5.7b, 5.7c, 5.7d learned data from a hand-drawn demonstration. 5.7e, 5.7f, 5.7g, 5.7h, 5.7i, imitate given functions.

5.3.1 Feature Scaling

DMP are designed to work with demonstrated trajectories, so these given trajectories are used as features. The DMP learn these features, by adapting its characteristics. A blending behavior between the characteristics and the dynamical systems provides reaching the goal in common situation. In several cases this can fail, in the used experiment the robot should not collide with obstacles, the blending mode can lead here to collisions. Because of the goal attraction an blending the DMP don't follow the characteristics at the end of the DMP. So the feature must show the same adaption behavior as the DMP. To guarantee this the feature will be scaled to enhance the characteristic behavior, while goal reaching is still possible.

Example

In the given scenario the DMP has adapted a sine like behavior (Figure 5.8). Due to the influence of the canonical system, the DMP lose its characteristics at the end to reach the attracted goal point. With feature scaling this does not happen, the used demonstration gets internal scaled to reach the goal point.
For goal points with larger distance to the demonstration goal point, DMP can become unstable. This can be seen in Figure 5.9. To avoid this feature scaling provides a scaled demonstration to avoid this, which results in smooth converging to the attracted point.

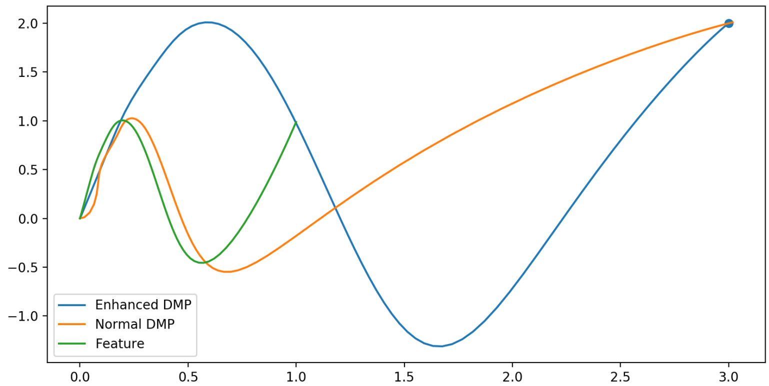


Figure 5.8.: The green graph is the used demonstration for DMP, the Normal DMP (orange) reaches the goal point with a descent lose of the given characteristic. The enhance DMP reaches its goal point, while the shape of the demonstration is complete recognizable.

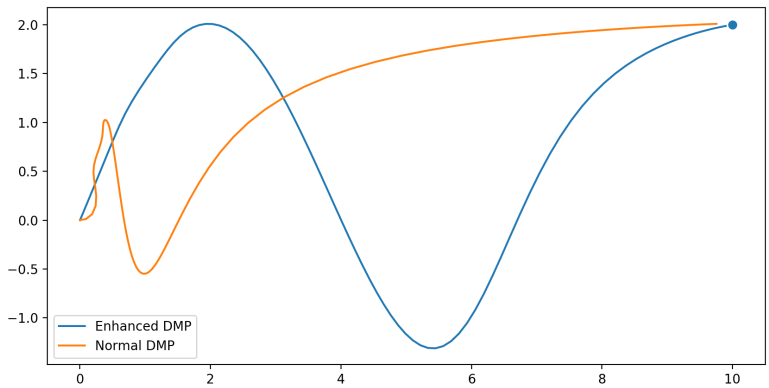


Figure 5.9.: For a goal point far from the goal of the feature, DMP can become unstable. The enhanced DMP with feature scaling, reaches this point, while the characteristic of the demonstration is recognizable.

5.3.2 Influence of Different Parameters

Dynamical movement primitives use several parameters, as shown in 2. These parameters influence, the precision, speed and factor of learning of a given DMP. To use DMP in an optimal way a fine adjustment of theses parameters is necessary.

Dynamical Movement Primitives Compared with PD-Controllers

DMP show similarities to classical PD-Controllers, for further experiments which describe the influence of DMP parameters, basic knowledge is needed for better understanding. The DMP equation is given by the first equation of 5.8, by several simple transformation the equation can represented similar as a PD-Controller 5.4

$$\ddot{y} = K_P(y_{\text{des}} - y) + K_D(\dot{y}^{\text{des}} - \dot{y}) + f(t) \tag{5.4}$$

$$\frac{\ddot{y}}{\tau^2} = \alpha(\beta(g-y) - \frac{\dot{y}}{\tau}) + f(t) \tag{5.5}$$
$$\frac{\ddot{y}}{\tau^2} = \alpha\beta(g-y) - \alpha\frac{\dot{y}}{\tau} + f(t) \tag{5.6}$$
$$\ddot{y} = \alpha\tau^2\beta(g-y) - \alpha\tau\dot{y} + f(t)\tau^2 \tag{5.7}$$
$$\tag{5.8}$$

Influence of the Canonical System

The parameter α_z defines the scale of blending between a given feature and the dynamical system of the DMP. By increasing the size of parameter α_z the created DMP will begin earlier with blending to the dynamical system, while the feature will be earlier neglected.
This is shown by 5.10. DMP 1 represents a DMP which follows exactly the path of the given demonstration. For this, α_z is defined as 8.8 which is needed to keep the systems stable. The given α_z represents a very close fit to the original demonstration used in this example. An adjustment of α_z by increasing it up to $\alpha_z = 12$, shows that the DMP begin earlier the neglecting of the original demonstration. This behavior continues with higher values, which influence DMP 3 and DMP 4. The influence of α_z effects for to high or to low values the stability of the DMP, this can bee seen immediately after the beginning of the last two DMP, as an overshooting behavior.

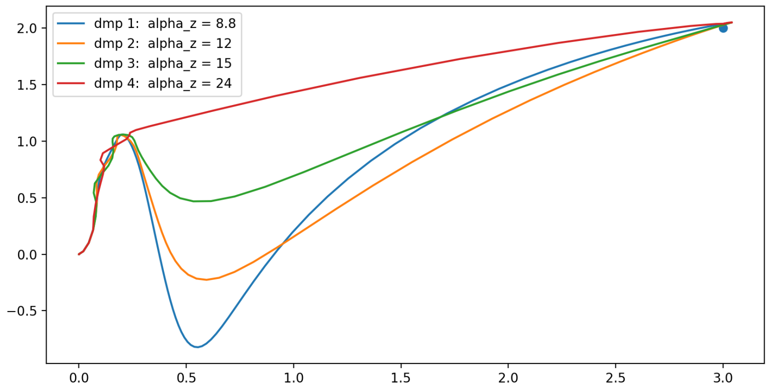


Figure 5.10.: Different behaviors by different α_z.

Influence of Basis Functions

Dynamical movement primitives use basis functions as non-linear approximator of a referred demonstration. The used number of basis functions does not necessary effects the convergence of the DMP, but the level of quality of the feature representation. A small number of basis functions lead to low approximation, overshooting and other unwanted behavior. The number of necessary basis functions depend mainly on the given feature. A detailed feature should use more basis functions as a low detailed. This can be seen, in 5.11. The first two DMP, with only 30 and 50 basis functions lead to unwanted overshooting, while a number of 100 basis functions is enough to provide a good fitting behavior.

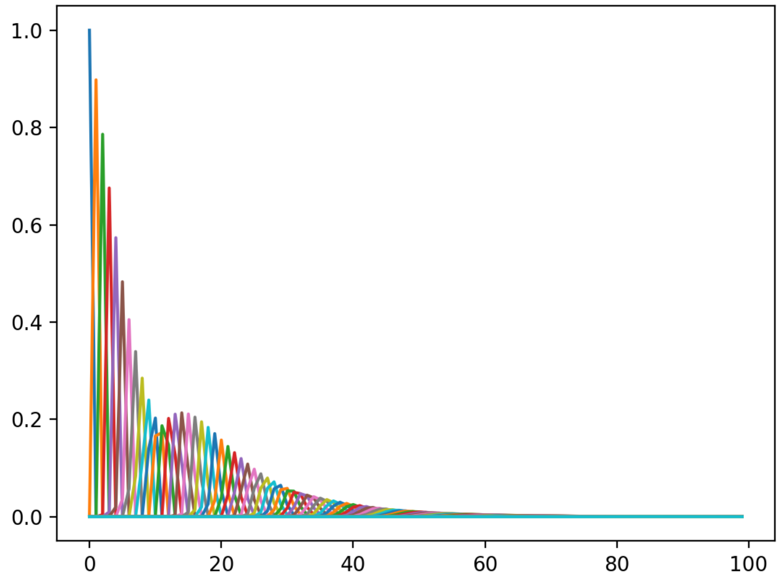


Figure 5.11.: Basis functions which approximates a referred trajectory of a sinus like demonstration.

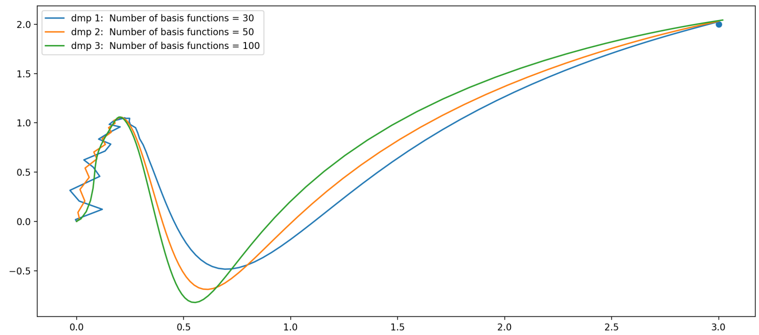


Figure 5.12.: Different behaviors by a different number of basis functions.

Influence of Parameter Beta

The formula of dynamical movement primitives is comparable with the equation of an spring-damper system. The product of α and β effect the gain and damping behavior of a given DMP (Section 5.3.2) So the value of proportion between α and β must be chosen carefully.

5.4 Application to a Restricted Workspace in Robot Ball Catching

The proposed experiment uses obstacles to, which force the robot to choose between different DMP. Aim is to train the robot choose between different methods for reaching the predicted position on which the robot should catch the ball. This implies that the robot should decide between an ideal linear behavior, for reaching the end position, which is faster due to the shorter distance. If an obstacle blocks the ideal linear path, the robot must choose a trajectory, from a set of given trajectories, that are optimal in this case. Based on the adaptivity of DMP via demonstrations a set of trajectory characteristics can be learned, due to the fact that based on this we are able to generate different DMP, with different behaviors to reach a goal point. A simple collision detection script allows to predict which DMP is the best to solve the task.

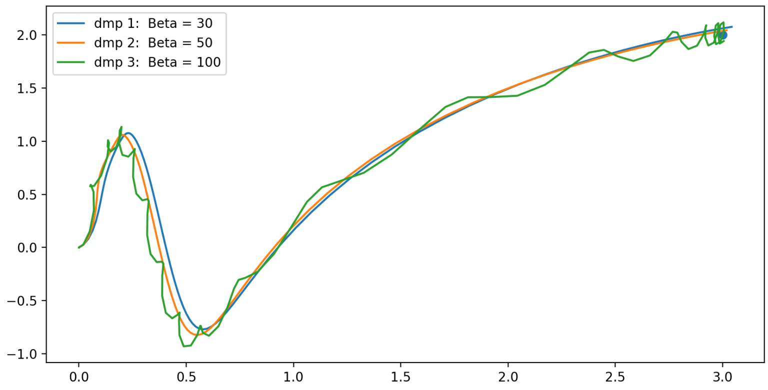


Figure 5.13.: Different behaviors by different β -Parameters.

5.4.1 Example

In this given example 5.14 an obstacle blocks the linear DMP, which should reach the point directly. In case some collisions occur during execution, an other DMP behavior must be chosen, in this case an parabolic or sine behavior solves the problem. This fact can be used for optimization, by planning different trajectories in the robots "mind", collisions can be estimated before execution. In this example the linear DMP takes the shortest path (length = 4.2331), while the parabolic DMP (length= 4.4111) and sine DMP (length= 13.8639) are longer. By taking the shortest DMP which does not collide, motor skills can be improved.

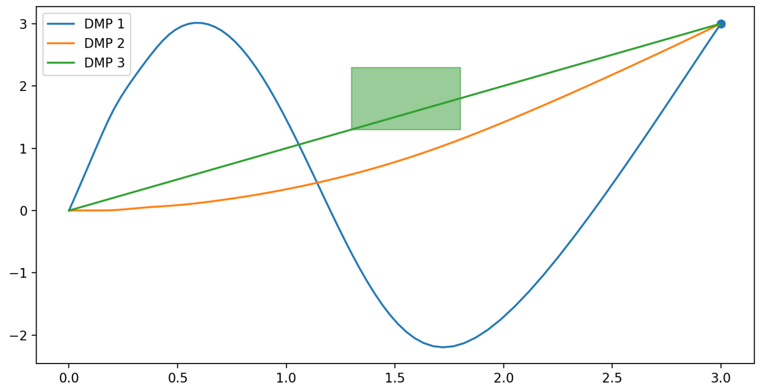


Figure 5.14.: Different characteristics of DMP to reach the same point. DMP 1 uses a sine demonstration, DMP 2 uses a parabolic demonstration and DMP 3 is linear.

5.4.2 Simulated Movement Primitives for Collision Detection

Mental imagery can be used for prediction, as introduced mental imagery can be used for forecasting, especially if a base knowledge does exist. In this given example the robot own a base knowledge of the environment. Before a movement will be performed the robot is able to test the execution of different goal points (Figure 5.15). This enable the robot to pre-select DMP and determinate its possible range.
By analyzing the length of the DMP, a selection of DMP is possible, to identify the shortest possible solution to reach the attracted goal point. This technique can be used to test the variance of a given DMP, by perception of the real world or a given model.

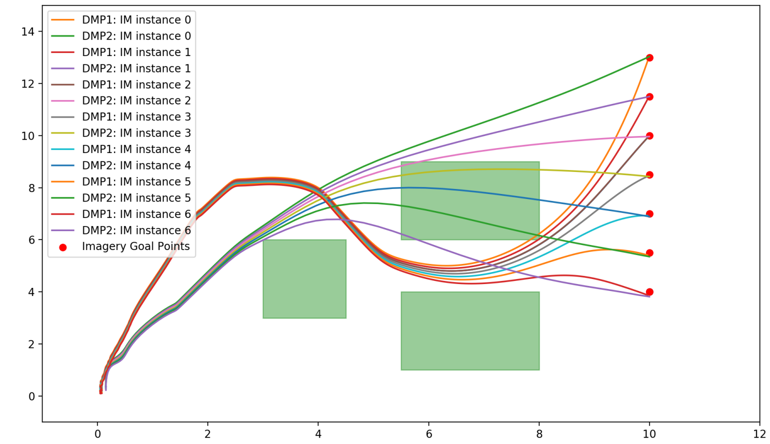


Figure 5.15.: 7 "imagined" Trajectories, for 2 different DMP.

5.4.3 Dynamical Movement Primitives Performed by a 2 Degree of Freedom Robot

To perform the DMP a 2 Degree of Freedom was used 5.16. The kinematics of the robot are given by (5.9, 5.10) for the forward kinematics and (5.11, 5.12, 5.13, 5.14, 5.15) for the inverse kinematics. The used variables in the kinematic equations are described in 5.17.

Forward kinematics

The forward kinematics describe the defined position of the end effector for given angulars θ_1, θ_2.

$$x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \tag{5.9}$$

$$y = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \tag{5.10}$$

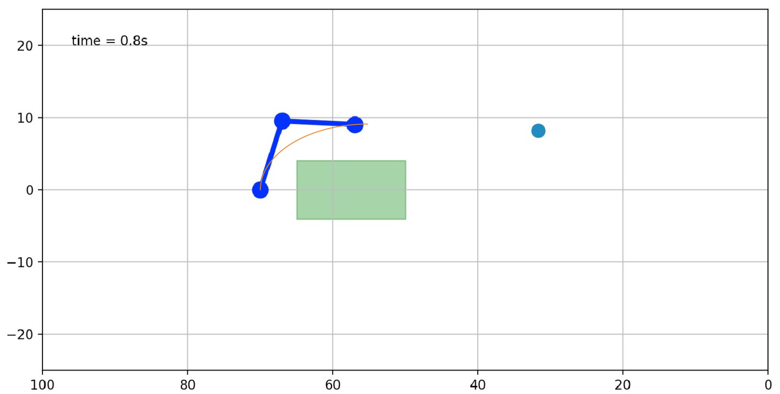


Figure 5.16.: The robot (blue) performs a catch action for the demonstration of an incoming ball. An obstacle restricts the movement. The desired trajectory is represented by the orange line.

l_1, l_2	Length of the first and second link
θ_1, θ_2	Angular of the first and second joint
x, y	Position of the robot end effector

Figure 5.17.: Used variables for the 2-DoF robot.

Inverse Kinematics

To calculate the joint angles for a given position inverse kinematics is needed. The output of the DMP trajectory is the input of this equation.

$$k_1 = l_1 + l_2 \cos(\theta_2) \tag{5.11}$$

$$k_2 = l_2 \sin(\theta_2) \tag{5.12}$$

$$\gamma = atan2(k_2, k_1) \tag{5.13}$$

$$\theta_2 = atan2\left(\sqrt{1 - (\frac{x^2 + y^2 - l_1^2 - l_2^2}{2 l_1 l_2})^2}, \frac{x^2 + y^2 - l_1^2 - l_2^2}{2 l_1 l_2}\right) \tag{5.14}$$

$$\theta_1 = atan2(y, x) - \gamma \tag{5.15}$$

6 Conclusion and Future Work

Humans can profit significantly from the use of mental training in motor skill learning. However, transferring insights from mental training to robot motor skill learning can be advantageous for different reasons. First, mental training can reproduce multiple training scenarios from a learned model of the environment. In particular, in situations where less physical training data is available this may accelerate the motor skill learning. For the case of ball catching the thesis showed experiments to generate more mental ball trajectories. This allows better adaption of the MoMP to the points where the robot should catch the ball. Second, a low information model, provides enough data to reproduce information. This situation appears in problems, in which a large number of physical training iterations is not possible, but each physical training sample is close to the other samples. Here motor imagery is capable of reproducing more training data. Third, if several different models can be learned independent to each other. For the particular application of robot ball catching a accurate model of the ball can be used in mental training to speed up learning to catch in different ways e.g. in a constrained workspace. Due to the fact, that the ball model is sufficient to reproduce the ball information, for training the MoMP of the robot. The ability to test DMP trajectories before physical execution is useful, to avoid damages. Additional this technique can be used to estimate which movement is more suitable for a given tasks, which is meaningful for the development motor skill learning.

The developed regression for motor imagery approach shows, that it is straightforward to reproduce captured data based from demonstrated data by using a dynamical model. Especially in situations in which prior knowledge exists, adaptions to the model can lead to improved motor imagery training. More results and comparisons between motor imagery training and physical training seem provide interesting points for future research. By implementing the described ball catching experiment on a real robot platform this comparison would be possible. Especially the obstacle avoidance of the complete robot arm via motor imagery can be tested.

For Future work several high level topics are interesting:

First, formalize the proportion between motor imagery training and physical training, to find the real limits behind this concept. Especially in situations in which humans interact with robots via demonstrations a large number of physical training data is expected. This amount of data can be passively learned via motor imagery. Additional it is interesting if motor imagery can downgrade the motor skills in some situations. For example in tasks in which physical training leads to wrong interpretations of models, or if motor imagery reproduces unwanted data for the task.

Second, use the collected knowledge, to transfer motor imagery inspired concepts to more complex robotics problems. Based on more data, motor imagery may show more benefits. For this new more generalized concepts can be created, to deal with more and different types of data.

Third, investigate how the dynamical models, which are used during the motor imagery concept can be saved in databases. The resulting models in the databases can be transferred between different robots. These databases can be used in connection with "lifelong learning". Based on these databases more data is available to create mental simulations. Because of this and the learned good initial model, the robot should be able to mentally train a task before it perform this task in physical training. For example in situations in which two different model databases are merged, motor imagery can be used to learn the connection between these two models. This can be done by simulating a problem which uses these two databases. A concrete example for this part of future work is: The connection of a model which comprises skills for moving a robot, with a learned ball catching model. The synthesis of both models can be simulated in motor imagery training. Then a physical training can be performed, in which the robot has to go to the location where the thrown ball is reachable and then catch the ball (Figure 6.1).

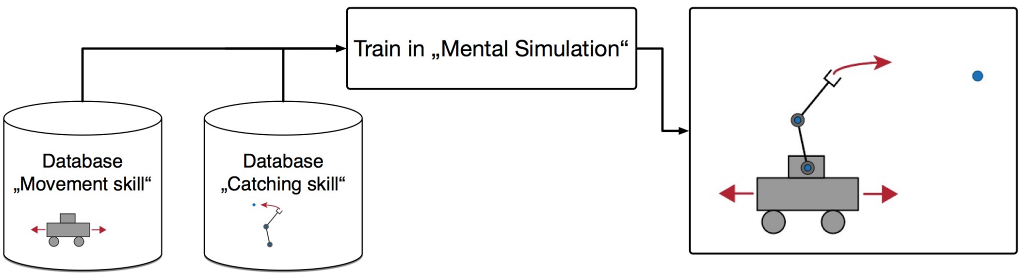


Figure 6.1.: Extending the concepts presented in this thesis to enable robots to test different combinations of motor skills by using mental training is a possible line for future work. Hereby, the robot could test combinations of skills for moving its base and catching a ball via mental training.

Bibliography

[1] K. Muelling, J. Kober, and J. Peters, "Learning table tennis with a mixture of motor primitives," *2010 10th IEEE-RAS International Conference on Humanoid Robots, Humanoids 2010*, pp. 411–416, 2010.

[2] B. Maloney and P. Newbold, "Forecasting Exchange Rates Using Feedforward and Recurrent," vol. 10, no. 4, pp. 347–364, 1994.

[3] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[4] M. Johnson and M. Schredl, "The Phenomenology of Lucid Dreaming : An Online Survey The Phenomenology of Lucid Dreaming : An Online Survey," vol. 127, no. May, pp. 191–204, 2014.

[5] M. Schädlich and D. Erlacher, "Applications of lucid dreams: An online study," vol. 5, 01 2012.

[6] C. A. Buneo and R. A. Andersen, "The posterior parietal cortex: Sensorimotor interface for the planning and online control of visually guided movements," 2006.

[7] M. Desmurget, C. M. Epstein, R. S. Turner, C. Prablanc, G. E. Alexander, and S. T. Grafton, "Role of the posterior parietal cortex in updating reaching movements to a visual target," *Nat Neurosci*, vol. 2, no. 6, pp. 563–567, 1999.

[8] S. J. Blakemore and A. Sirigu, "Action prediction in the cerebellum and in the parietal lobe," *Experimental Brain Research*, vol. 153, no. 2, pp. 239–245, 2003.

[9] J. Doyon, P. Bellec, R. Amsel, V. Penhune, O. Monchi, J. Carrier, S. Lehéricy, and H. Benali, "Contributions of the basal ganglia and functionally related brain structures to motor learning," *Behavioural Brain Research*, vol. 199, no. 1, pp. 61–75, 2009.

[10] H. J. Groenewegen, "The Basal Ganglia and Motor Control," *Neural Plasticity*, vol. 10, no. 1-2, pp. 107–120, 2003.

[11] Richard Morris and M. Fillenz, "Neuroscience: the Science of the Brain," p. 5, 2003.

[12] Richard Morris and M. Fillenz, "Neuroscience: the Science of the Brain," pp. 1–71, 2003.

[13] J. Pearson and S. M. Kosslyn, *Mental Imagery*. 2013.

[14] D. B. Willingham, "A Neuropsychological Theory of Motor Skill Learning," vol. 105, no. 3, pp. 558–584, 1998.

[15] K. Teffer and K. Semendeferi, *Human prefrontal cortex. Evolution, development, and pathology.*, vol. 195. Elsevier B.V., 1 ed., 2012.

[16] D. M. Wolpert, "Computational approaches to motor control," *Trends Cogn Sci*, vol. 1, no. 6, pp. 209–216, 1997.

[17] J. Decety, H. Sjöholm, E. Ryding, G. Stenberg, and D. H. Ingvar, "The cerebellum participates in mental activity: tomographic measurements of regional cerebral blood flow," *Brain Research*, vol. 535, no. 2, pp. 313–317, 1990.

[18] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Probabilistic Movement Primitives," *Neural Information Processing Systems*, pp. 1–9, 2013.

[19] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Control, Planning, Learning, and Imitation with Dynamic Movement Primitives," *Workshop on Bilateral Paradigms on Humans and Humanoids, 2003 IEEE International Conference on Intelligent Robots and Systems IROS*, pp. 1–21, 2003.

[20] K. Seepanomwan, "Mental imagery in humanoid robots," no. April, 2016.

[21] L. Xu, M. I. Jordan, and G. E. Hinton, "An alternative model for mixtures of experts," *Nips*, no. 7, pp. 633–640, 1994.

[22] P. Moerland, "Some Methods for Training Mixtures of Experts," pp. IDIAP–Com 97–05, 1997.

[23] M. Desmurget, K. Reilly, N. Richard, A. Szathmari, C. Mottolese, and A. Sirigu, "Movement intention after parietal cortex stimulation in humans.," *Science (New York, N.Y.)*, vol. 324, no. 5928, pp. 811–813, 2009.

[24] A. Sirigu, J.-R. Duhamel, L. Cohen, B. Pillon, B. Dubois, and Y. Agid, "The Mental Representation of Hand Movements After Parietal Cortex Damage," *Science*, vol. 273, no. 5281, pp. 1564–1568, 1996.

[25] J. Banks, "Discrete Event Simulation," *Winter Simulation Conference*, pp. 7–13, 1999.

[26] H.-J. Bungartz, S. Zimmer, M. Buchholz, and D. Pflger, *Modeling and Simulation: An Application-Oriented Introduction*. Springer Publishing Company, Incorporated, 2013.

[27] J. Randløv and P. Alstrøm, "Learning to Drive a Bicycle using Reinforcement Learning and Shaping," *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 463–471, 1998.

[28] H. H. Ehrsson, "Imagery of Voluntary Movement of Fingers, Toes, and Tongue Activates Corresponding Body-Part-Specific Motor Representations," *Journal of Neurophysiology*, vol. 90, no. 5, pp. 3304–3316, 2003.

[29] A. Jain, J. Mao, and K. M. Mohiuddin, “Artificial neural networks: A tutorial,” 1996.

[30] S. M. Kosslyn and S. P. Shwartz, “A simulation of visual imagery,” *Cognitive Science*, vol. 1, no. 3, pp. 265–295, 1977.

[31] C. Hall, W. Rodgers, and K. Barr, “The use of imagery by athletes in selected sports.,” *The Sport Psychologist*, no. FEBRUARY 1990, pp. 1–10, 1990.

[32] A. R. Barron, “Universal approximation bounds for superpositions of a sigmoidal function,” *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 930–945, 1993.

[33] A. Guillot, C. Collet, V. A. Nguyen, F. Malouin, C. Richards, and J. Doyon, “Functional neuroanatomical networks associated with expertise in motor imagery,” *NeuroImage*, vol. 41, no. 4, pp. 1471–1483, 2008.

[34] M. J. Farah, “The neural basis of mental imagery,” *Trends in Neurosciences*, vol. 12, no. 10, pp. 395–399, 1989.

[35] S. Lacey and R. Lawson, *Multisensory imagery*. 2014.

[36] Pylyshyn, “Mental Imagery - The Oxford Companion to the Mind,” *The Oxford Companion to the Mind*, pp. 1–4, 1980.

[37] R. Gentili, C. E. Han, N. Schweighofer, and C. Papaxanthis, “Motor Learning Without Doing : Trial-by-Trial Improvement in Motor Performance During Mental Training Motor Learning Without Doing : Trial-by-Trial Improvement in Motor Performance During Mental Training,” no. June 2010, p. 781, 2013.

[38] J. Pearson and S. M. Kosslyn, *Mental Imagery*.

[39] S. M. Kosslyn, “Mental images and the Brain,” *Cognitive Neuropsychology*, vol. 22, no. 3-4, pp. 333–347, 2005.

[40] R. Gentili, C. E. Han, N. Schweighofer, and C. Papaxanthis, “Motor Learning Without Doing : Trial-by-Trial Improvement in Motor Performance During Mental Training Motor Learning Without Doing : Trial-by-Trial Improvement in Motor Performance During Mental Training,” no. June 2010, pp. 774–783, 2013.

[41] P. Reviewed and F. Expression, “The emulation theory of representation: Motor control, imagery, and perception,” 2005.

[42] M. Jeannerod, “Mental imagery in the motor context,” *Neuropsychologia*, vol. 33, no. 11, pp. 1419–1432, 1995.

A Appendix

A.1 Neural Network for Mental Imagery

The common solution to solve mental imagery problems is to use a combination of two different kinds of neural networks. Lets say a robot should be able to use mental imagery to improve it's performance: The robot well be take use of two kinds o neural networks, a Feed forward neural network for controlling the actual task, and a second neural network a dual recurrent network for reproducing training data to learn from the previous tasks Neural networks are .

A.1.1 General Informations

Neural networks provide a class of nonlinear models which has been successfully applied to many different fields, especially in robotics. [2] Neural networks are often used in econometric applications in following two respects. multilayer neural networks can approximate a large class of functions, and are commonly used used for nonlinear time series models, additional neural networks are more parsimonious models than linear subspace methods such as polynomial, spline, and trigonometric series expansions by approximating unknown functions. [2]

A.1.2 Feed Forward Neural Networks

A neural network can be interpreted as a non-linear regression function characterizing the relationship between a target variable y and an n-vector of explanatory variables which entries are called inputs (x). Instead of postulating a specific non-linear functions, the neural network model uses a composition of many 'basic' nonlinear function via a multilayer structure. The feed forward neural network, works in several steps first explanatory variables activate simultaneous q hidden units in an intermediate layer, through the function of Ψ A.1, this results in the hidden-unit activation h_i, with $i = 1, ...q$. After that an output is generated through the function of Φ A.2.

$$h_{i,t} = \Psi\big(\gamma_{i,0} + \sum_{n}^{j=1} \gamma_{ij} x_{j,t}\big) \qquad i = 1, ..., q \tag{A.1}$$

$$o_t = \Phi\big(\beta_0 + \sum_{i=1}^{q} \beta_i h_{i,t}\big) \tag{A.2}$$

In the compact form $f_q(x_t, \theta)$. θ is assign compatible to A.3, in this example θ is a vector which contains the components of β and γ and the subscript of q of f signifies the number of hidden units that are in the network.
This is a flexible nonlinear functional form, in that Φ and ψ are arbitrary chosen function, except that Ψ is required to be a function that is bounded, for example the sigmoid function. This concept is able to approximate a large class of functions arbitrarily well, if they follow a suitable metric and the number of hidden units of q is sufficiently large enough. This is a property also known from nonparametric methods. As an example for this we consider the L_2 approximation property, in which a dependent variable y and explanatory variables x are given. During the L_2 approximation we are interested in the conditional mean, that is unknown $M(x) := E(y|x)$. If $M(x) \in L_2$, then for any $\epsilon > 0$, there is a q that behaves like $E|M(x) - f_q(x, \theta)|^2 < \epsilon$

$$o_t = \Phi\big(\beta_0 + \sum_{i=1}^{q} \beta_i \Psi\big(\gamma_{i,0} + \sum_{j=1}^{n}\big)\big) \tag{A.3}$$

The work of Barron [32] shows that neural networks are able to achieve an approximation rate of $O(\frac{1}{q})$ by using $O(qn)$ parameters that grow linearly in q. Traditional approaches, for example polynomial, spline or trigonometric expansions require an exponentially $O(q^n)$ terms to achieve the same approximation rate.
Due to the fact that neural networks are common used to implement the concept of mental imagery, this high approximation rate is a strong benefit.

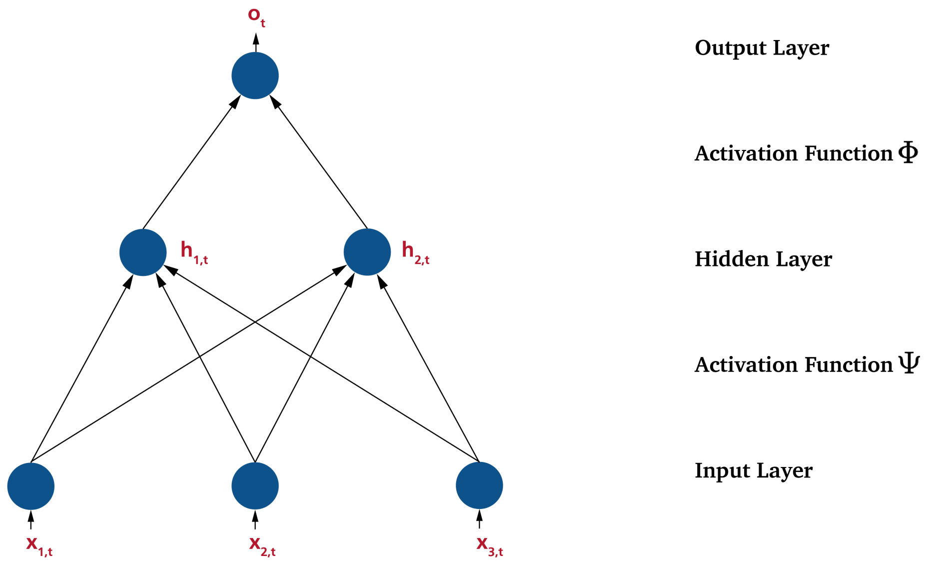


Figure A.1.: Model of a simple feed forward neural network, with input, and hidden output Layer. Using three Input unit, two hidden units and one output unit [2].

A.1.3 Recurrent Neural Network

Recurrent neural networks, work similar to feed forward neural networks, the primary difference is that recurrent networks work with feedbacks to characterize the behaviour of depended variable y. A recurrent network consists of a much richer dynamic structure and is more similar to a linear time series model with moving average terms. For a dynamic context, lagged dependent variables have to be included as explanatory variables to capture dynamics. For this approach a Drawback is needed, with a unknown number of of lags, which is analogous to problem to find the correct order in autoregression.[2]
The equations for recurrent network, look very similar to the equations of the feed forward neural network. The activation of Psi is defined by A.4 and the activation function of Φ is defined by A.6 The following neural network is was defined by Jeffrey L. Elman [3]

$$h_{i,t} = \Psi\big(\gamma_{i,0} + \sum_{j=1}^{n} \gamma_{i,j}\chi_{j,t} + \sum_{l=1}^{q} \delta_{i,l} h_{l,t-1}\big) \tag{A.4}$$

$$h_{i,t} = \psi(x_t, h_{t-1}, \theta) \tag{A.5}$$

$$o_t = \Phi\big(\beta_0 + \sum_{i=1}^{q} \beta_i \psi_i(x_t, h_{t-1}, \theta)\big) \tag{A.6}$$

$$o_t = \phi_q(x_t, h_{t-1}, \theta) \tag{A.7}$$

A.7 and A.5 denote the functions with θ as an parameter of β, γ, δ and the subscript q of ϕ again signifies the number of used hidden units.
The in A.2 is described how the feedback in the neural network works. Via hidden layer activations that send the feedback with some delay to the input layer, the networks 'memorize' the past information. This can be written as A.8

$$r_i(x^t, \theta) = h_{i,t} = \psi_i(x_t, \psi(x_{t-1}, h_{t-1}, ..., \theta), \theta) \qquad i = 1, ...q \tag{A.8}$$

where x^t and ψ are vector-valued with i-components. Due to the fact that $h_{i,t}$ depends on the variable x_t and its entire history, the equation A.9, shows that the recurrent network is able to capture more dynamic characteristics of y_t than a feed forward neural network.

$$o_t = \psi_q(x_t, h_{t-1}, \theta) \tag{A.9}$$

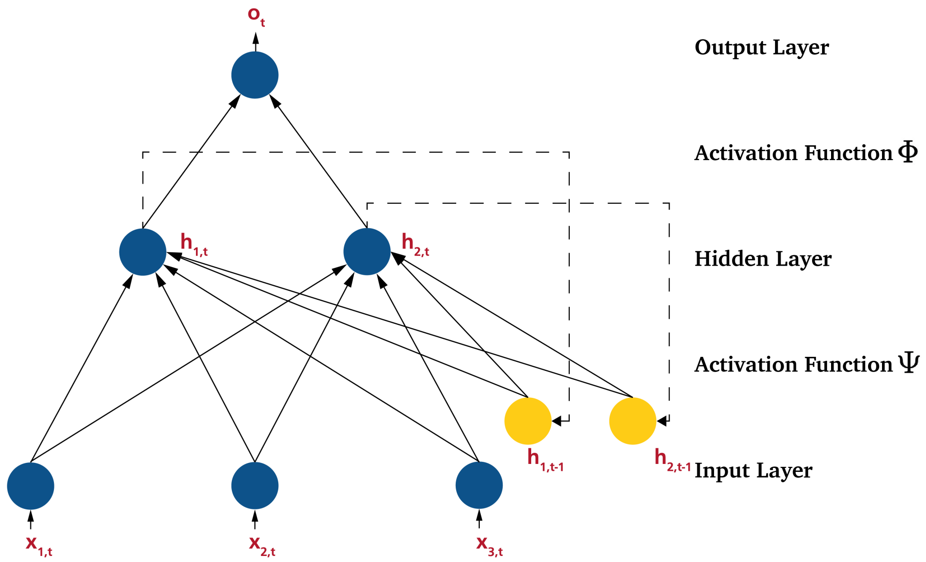


Figure A.2.: Model of a reccurent neural network by Elman, with hidden-unit activations feedback [3].