Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

Distal teacher learning in robotic arm reaching with active exploration

Bachelor's thesis

Study Programme:	Applied Computer Science (Single degree study, bachelor I. deg., full time form)
Field of Study:	9.2.9. Applied Informatics
Supervisor:	prof. Ing. Igor Farkaš, PhD.

Bratislava 2015 **Karin Alexandra Vališová**

Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

THESIS ASSIGNMENT

Name and Surname:	Karin Alexandra Fališová
Study programme:	Applied Computer Science (Single degree study, bachelor I. deg., full time form)
Field of Study:	9.2.9. Applied Informatics
Type of Thesis:	Bachelor´s thesis
Language of Thesis:	English
Secondary language:	Slovak

Title: Distal teacher learning in robotic arm reaching with active exploration

Aim: 1. Study the literature related to inverse and forward models in robotics, and application of a distal teacher learning using artificial neural networks.
2. Implement the task of goal reaching with a robotic arm using the distal teacher and active goal exploration.
3. Using computational simulations, analyse the model performance.

Annotation: Supervised learning with a distal teacher is a well-known method (Jordan & Rumelhart, 1992) that uses a forward model for training the inverse model. An inverse model predicts an action needed to reach the goal state, whereas a forward model predicts sensory consequences of generated actions. Active exploration of goals or actions is a new approach in cognitive robotics aimed at improving the learning of both models.

Keywords: inverse and forward model, distal teacher learning, active goal exploration

Supervisor:	prof. Ing. Igor Farkaš, PhD.
Department:	FMFI.KAI - Department of Applied Informatics
Head of department:	prof. Ing. Igor Farkaš, PhD.
Assigned:	18.10.2013
Approved:	28.10.2013

doc. RNDr. Damas Gruska, PhD.
Guarantor of Study Programme

....
Student

....
Supervisor

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Karin Alexandra Vališová
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: 9.2.9. aplikovaná informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: anglický
Sekundárny jazyk: slovenský

Názov: Distal teacher learning in robotic arm reaching with active exploration
Učenie s dištančným učiteľom pri siahaní robotického ramena pomocou aktívnej explorácie

Cieľ: 1. Preštudujte literatúru týkajúcu sa inverzných a dopredných modelov v robotike, a aplikácie dištančného učiteľa s využitím umelých neurónových sietí.
2. Implementujte úlohu siahania na ciele pomocou robotického ramena s využitím dištančného učiteľa a aktívnej explorácie cieľov.
3. Pomocou výpočtových simulácií analyzujte správanie modelu.

Anotácia: Učenie pomocou dištančného učiteľa je známa metóda (Jordan & Rumelhart, 1992), ktorá využíva dopredný model na trénovanie inverzného modelu. Inverzný model predikuje akciu potrebnú na dosiahnutie cieľového stavu, zatiaľ čo dopredný model predikuje senzorické dôsledky generovaných akcií. Aktivna explorácia cieľov alebo akcií je novým prístupom v kognitívnej robotike, zameraným na zlepšenie učenia oboch modelov.

Kľúčové slová: inverzný a dopredný model, učenie pomocou dištančného učiteľa, aktívna explorácia cieľov

Vedúci: prof. Ing. Igor Farkaš, PhD.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, PhD.

Dátum zadania: 18.10.2013

Dátum schválenia: 28.10.2013

doc. RNDr. Damas Gruska, PhD.
garant študijného programu

.... študent

.... vedúci práce

I hereby declare that this thesis is my own work and that all sources I have used or quoted have been indicated and acknowledged as complete references.

Bratislava May 29, 2015 ....

signature

Acknowledgements

I woluld like to thank my supervisor prof. Ing. Igor Farkaš, PhD., for introducing me to the artificial neural networks and guidance through various topics in the field.

Abstract

This thesis is focused on an analysis of the distal teacher algorithm. It is an algorithm under the supervised learning paradigm that uses the forward model and the inverse model together in order to learn ill-posed tasks. A good example of such a non-unique mapping is the reaching problem - simulation of the 2D robotic arm with five degrees of freedom operating over the unit circle half-plane. The cognitive model using the distal teacher learning is implemented using the artificial neural networks, and the reaching problem is simulated and used for testing the capabilities of the algorithm. The system is enhanced by the active goal exploration and its contribution to the performance of the model is also analysed.

Keywords: inverse and forward model, distal teacher learning, active goal exploration

Abstrakt

Táto práca sa venuje analýze algoritmu učenia s dištančným učiteľom. Tento algoritmus využíva dopredný a inverzný model na naučenie problémov, ktoré spočívajú v nejednoznačnom mapovaní. Príkladom takéhoto problému je napríklad siahanie v priestore - v práci sa využíva počítačová simulácia 2D robotického ramena s piatimi stupňami voľnosti operujúceho nad polrovinou jednotkovej kružnice. Kognitívny model vykonávajúci siahanie využíva algoritmus dištančného učiteľa a je implementovaný pomocou neurónových sietí. Simulácia je využitá na testovanie možností a efektivity tohto algoritmu. Systém je vylepšený použitím algoritmu pre aktívnu exploráciu cieľov, ktorý je tiež analyzovaný.

Kľúčové slová: inverzný a dopredný model, učenie pomocou dištančného učiteľa, aktívna explorácia cieľov

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

One of the most basic evolutionary functions of our brain is the movement (Dawkins, 2006). Being absolutely necessary requirement for any physical interaction with the environment, the movement and sensorimotor mappings are widely studied problems in the robotics and cognitive science.

The problem of simulating natural and fairly easy activities for humans, such as reaching for an object in the visual field, is a very challenging task for the artificial intelligence. Proper navigation in space requires high degree of flexibility and learning of complex sensorimotor mappings. The problem can be approached from various angles, the choice of the appropriate cognitive model is usually based on the studies from fields such as neurology, psychology or cognitive science. The models differ in performance, generalisation abilities, precision, speed and even biological plausibility.

In this thesis, one particular cognitive model is chosen and analysed. The model consists of two modules, each dealing with different aspect of kinematics: forward and inverse model. The forward model has the ability to predict next sensory state, working as kind of imagination module. (Lalazar, 2008) In our case, however, the model serves only as a simple mapping from one coordinate system to another. The inverse model is the 'actor', choosing the action in order to reach for the given target. These two modules can help each other in learning using the distal teacher algorithm described in the following chapters.

1.2 Goal

The aim of this thesis is to grasp the concepts from the literature related to the inverse and forward models and implement the distal teacher learning algorithm using the artificial neural networks. The core lies in implementation of the computational simulation of a cognitive model for reaching in a space using a robotic arm with five degrees of freedom operating over the unit circle half-plane. The analysis of the algorithm performance will be provided. The implementation of the active goal exploration is used too. This computational concept enhances can shed even more light on the distal teacher algorithm, behaviour of its error landscape and overall performance.

Chapter 2

Theoretical background

2.1 Cognitive Systems

The movement plays the key role in our interaction with the environment. The seemingly easy task of reaching for an object placed in the visual field of the actor is in fact a great challenge for artificial intelligence with computational approach to problematics. (Wolpert, 2001) If the system is sufficiently simplified, analytical solution to the inverse kinematics can be obtained. However, for models with high number of degrees of freedom, this problem is almost unsolvable in real time and requires perfect model of the arm and environment in order to make the calculations. (McKerrow, 1991) Therefore, different approaches than purely analytical solutions are used and the inspiration is taken from the nature. (Wolpert, 2001)

The mathematical model of the environment and the arm itself is luxury that is not usually present in the real-life situations. Let us analyse first, what component of the problems play a vital role in the architecture of the cognitive models.

2.1.1 Inverse Model

The inverse model in Figure **??** on the right side, sometimes referenced as the actor, has the ability to determine which action should be taken in order to reach the target state. It provides anti-causal relationship and this problem may not have a unique solution, i.e. it may be an ill-posed problem. (Nguyen-Tuong and Peters, 2011) In the case of the robotic arm, it is a mapping from joint angles to the changes of the angles (actions) or the absolute joint angles position.

2.1.2 Forward model

The forward model in Figure 2.1 on the left side is able to predict the next state of a system given the current action and the current state. Thus it represents the causal relationship between the actions and states. (Lalazar, 2008) In a simplified version used in this thesis, the forward model acts as a mapping between arm's joint angles and the Cartesian coordinates of the effector over the operating plane. This mapping is unique and provides causal relationship. (Nguyen-Tuong and Peters, 2011)

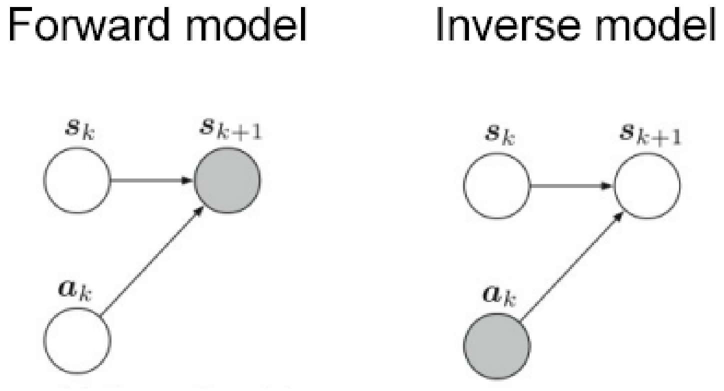


Figure 2.1: Schematic diagram of the inverse and forward model (Nguyen-Tuong and Peters, 2011, Fig 2) The white nodes denote the observed quantities, while the grey nodes represent the quantities to be inferred.

2.2 Motor Learning

The human brain possesses a great deal of plasticity and ability to learn new motor skills. The ability to improve the performance according to error correction is vital for mastering new skills. (Wolpert, 2001) According to the computational structure of the learning process, three distinct paradigms are present: Supervised learning, Reinforcement learning and Unsupervised learning.

All of these three approaches to acquiring new skills can be found in the nature.

2.2.1 Supervised learning

The supervised learning paradigm requires the presence of the external teacher or the feedback learning signal, for example during learning by imitation (Haykin, 1998) (Wolpert, 2001). In a classical approach, the cognitive system produces output, set of actions for example, that is performed and thus evaluated by the environment. The teacher provides the correction based on discrepancy between the desired state and the feedback from the environment and the system is adapted accordingly. This approach is very limited, let us illustrate it on the following example: consider a problem or reaching a certain position in the space. The actor's outcome (inverse model) is the muscle contraction pattern required to move the hand to the desired position. If any error is observed, the teacher must be able to provide the correct muscle contraction pattern that would have resulted in the goal position. It is almost impossible to imagine a system, which would allow the access to such low-level information as neuronal patterns of muscle contraction.

The supervised approach is usually not applicable for the motor learning, as it requires the omniscient teacher, the perfect mathematical model of the system, which is typically not available (in that case the whole concept of creating the cognitive model would be quite useless as all necessary motor commands can be obtained by the teacher alone). However, the idea of supervised learning can be extended to self-supervised learning which overcomes these problems.

2.2.2 Self-supervised learning by the distal teacher

The concept of distal teacher was first introduced by Jordan (Jordan and Rumelhart, 1992) in an attempt to expand the range of problems, where the supervised learning could be applied. He argues that the importance of the presence of the teacher signal can be weakened and substituted by properly chosen forward model.

The components present in distal teacher learning task are shown in Figure 2.2, consisting of *the intentions*, i.e. the task (for example reaching a certain position) given to learning systems. These intentions are by the inverse model transformed into *actions*. When confronted with the environment, the performed actions result in *outcomes*. The actions are considered as proximal variables, meaning that the target values are not provided as opposed to the distal variables, such as outcome with the defined target values. The learning of the system can be considered as a mapping from the intention to the desired outcomes and the general approach to solve the distal teacher learning problems consists of combining the forward and inverse model in order to minimise the distal error. All necessary technical and conceptual details are provided in the later chapter.

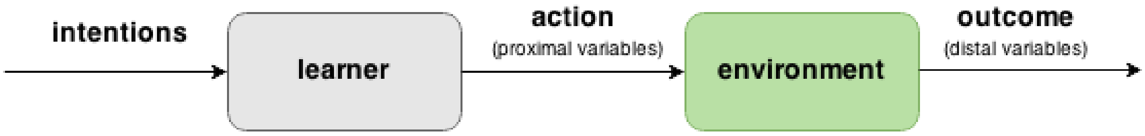


Figure 2.2: The distal teacher supervised problem (Jordan and Rumelhart, 1992, p. 3)

Chapter 3

Methods

3.1 Multilayer perceptron

Artificial neural networks are widely used models of the connectionist paradigm. This approach is based on a phenomenon of the emergent properties that arise from the connection of simple units, which is very commonly observable in the nature. The brain itself is considered to be highly complex, nonlinear and parallel computer consisting of millions of simple units, neurons. (Haykin, 1998)

The most important feature of the artificial neural network for this thesis is its ability to learn nonlinear function of the input data, serving as a universal function approximator. For this purpose the architecture of multilayer perceptron is commonly used. (Haykin, 1998)

Multilayer perceptron (MLP) is a feedforward multi-layered neural network consisting of input neurons that receive the input signal, which is propagated forward through the network to produce the activations of the output layer. The hidden neurons are organised in disjoints layers that are propagated one at a time.

Multilayer perceptrons are commonly used with the error backpropagation algorithm introduced (Rumelhart et al., 1988). It is based on the concept of propagation of the signal through the network in order to find the activation of the output layer. The output is then compared with the target values (thus it is a type of supervised learning) and adaptation of the weights in the backward pass follows. The weight matrix is adapted in order to perform gradient descent on the error surface.

3.2 Distal teacher learning

The distal teacher learning problem requires a few assumptions about the environment and the learner. The following definitions are taken from Jordan's paper that coined the term and algorithm in 1992 Jordan and Rumelhart (1992).

We can characterise the environment as a next-state function f and output function g. The action $u(t)$ is produced by the inverse model as a response to the desired state $d(t+1)$ and current (sensorimotor) state of the system $s(t)$. Note that for the sake of simplicity, the inverse model implemented in this thesis outputs not the change in angles - action, but rather the absolute joint position of the final position $s(t+1)$.

The following formulas are adapted from the original Jordan's work to our simplified model.

3.2.1 Problem definition

Each state $s(t)$ has a corresponding sensation $y(t)$ given by the function g.

$$y(t) = g(s(t))$$

So the relationship between sensation and action performed is determined by the model of environment, given as:

$$y(t+1) = g(s(t+1))$$

We assume that the agent has the access to the proximal variables, e.g. the state of the environment - in this case its own body and its joint angles, $s(t)$ at any given time. The distal variables - effector's position in the space $y(t)$ have to be obtained from the observation of the environment.

So given the state $s(t)$ and desired goal $d(t)$, the learner produces the state $s(t+1)$.

$$s(t+1) = inv(s(t), d(t))$$

The goal of the learning process is to minimise the difference between $d(t)$ and $g(s(t+1)) = y(t+1)$, the desired sensation (goal) and sensation observed when the action produced by the inverse model is performed. The inverse function is essentially an inverse model and the associated error $|d(t) - y(t+1)|$ is called *the performance error.* It is considered to be the objective measure of performance of the system throughout the thesis.

Forward model is the internal model providing a mapping between two spaces, one of sensorimotor states $s(t)$ and the other the sensation space (also referenced as Cartesian coordinates space). It is given by the relationship:

$$\hat{y}(t+1) = fwd(s(t+1))$$

The system gives the predicted sensation $\hat{y}(t+1)$, which is state predicted after the joint angles are moved to $s(t+1)$. This system can be trained by minimising the difference between $\hat{y}(t+1)$ and $y(t+1)$ and the error $|\hat{y}(t+1) - y(t+1)|$ is referenced to as *the prediction error.* Note that the forward model can be learnt in advance without the presence of the inverse model. This is usually a unique mapping, so the multilayer perceptron is capable of learning it. (Vijayakumar, 2007)

3.2.2 The algorithm

The problem consisting of above mentioned components can be considered as a composite learning system, with inverse and forward model linked together. The detailed interaction between the components is shown on the Figure 3.1

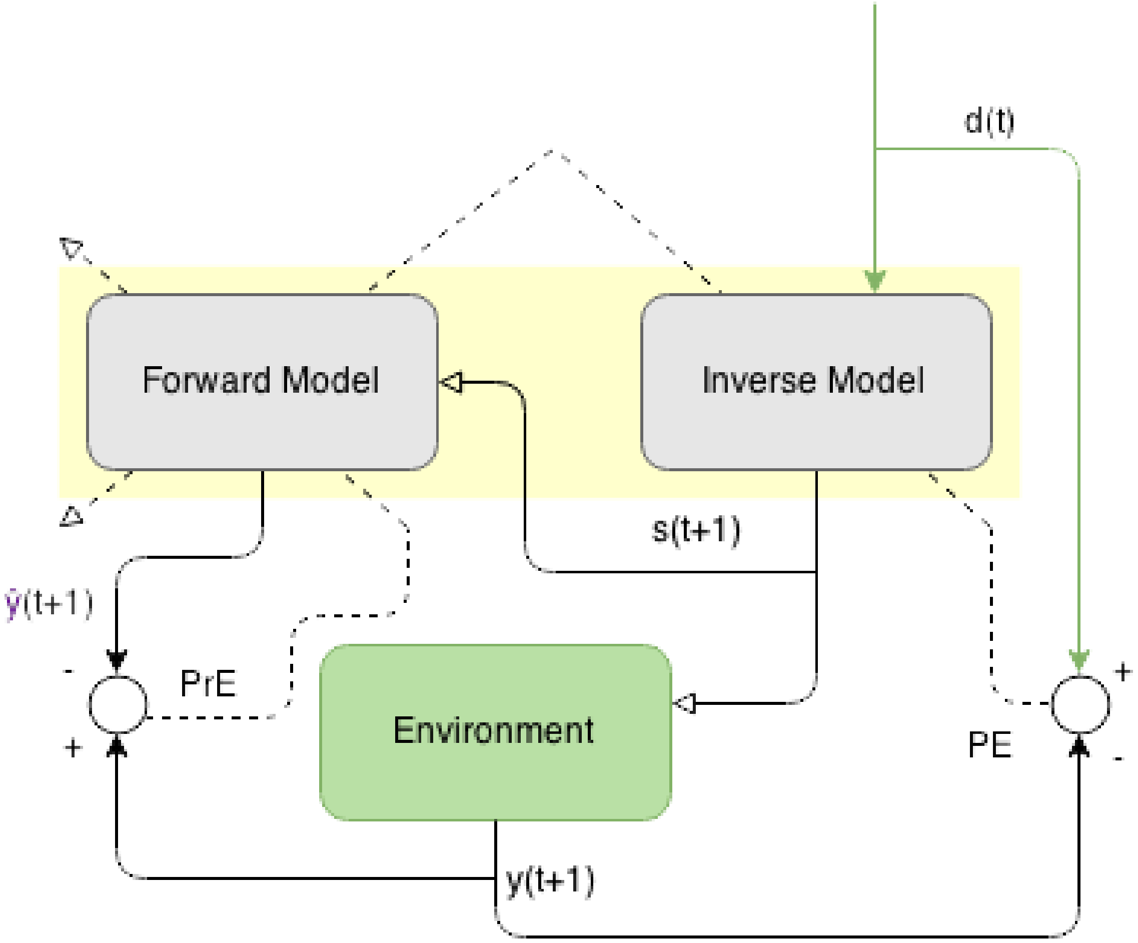


Figure 3.1: The distal teacher model architecture depicts the interaction between components of the system. Solid lines carry the signal, dashed lines provide the error.

The composite system is provided with the input consisting of the desired state $d(t)$ and the current position $s(t)$. The inverse model produces the output $s(t+1)$. The forward model uses the output $s(t+1)$ to make a prediction about the future state $\hat{y}(t+1)$. The real outcome $y(t+1)$ can be obtained from the environment by performing the action, i.e. adjusting the arm to $s(t+1)$. We can calculate the following errors:

The essence of the distal teacher learning is in the learning the composite model using the prediction error. All formulas for the application of this approach to the backpropagation are derived in the following section.

Performance error	PrE	$\lvert d(t) - y(t+1) \rvert$
Prediction error	PE	$\lvert y(t+1) - \hat{y}(t+1) \rvert$
Predicted performance error	PPE	$\lvert d(t) - \hat{y}(t+1) \rvert$

Table 3.1: The list of the errors and their abbreviations

3.3 Distal teacher and backpropagation

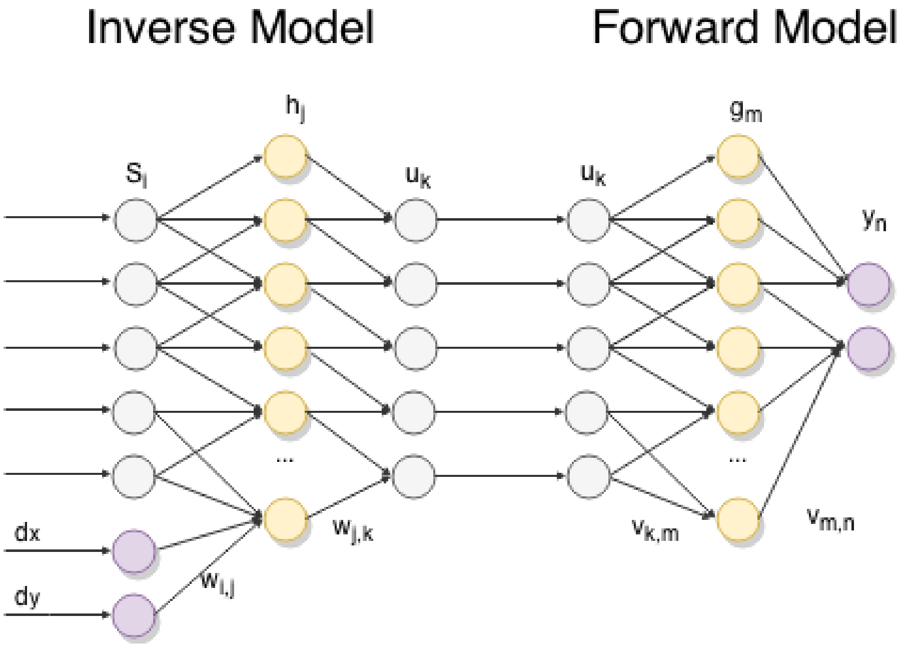


Figure 3.2: The structure of the neural network architecture with notation used in the formula derivation

In order to implement the distal teacher learning by the multilayer perceptrons, the weight-adaptation rule has to be derived. Let us assume two artificial neural networks, as depicted in the Figure 3.2, both with one hidden layer of neurons. The inverse model is mapping $s(t+1) = inv(s(t), d(t))$ and the forward model is $\hat{y}(t) = fwd(s(t))$.

Note: In the following section, for the sake of comprehensibility, the time indices in brackets (t) are omitted and the notation form the 3.2 is adapted. The notation also differs from Jordan's, because it uses indices rather than vectors, which is more relevant for the actual implementation of the neural networks as the multidimensional vectors.

The weights are changed based on the difference between the target d and the inverse model output y. (Vijayakumar, 2007) The gradient descent is in the direction of error minimisation measured as the sum of square errors of the output units of the network. We want to adjust the weights of the inverse model to minimise the function

J, Jacobian matrix, by rules of classical back-propagation. We know this formula from the least squares learning.

$$\frac{\partial J}{\partial w_{j,k}} = \frac{\partial J}{\partial y_n} \cdot \frac{\partial y_n}{\partial u_k} \cdot \frac{\partial u_k}{\partial w_{j,k}}$$

This is the expanded by the chain rule, each term of the equation can be evaluated. First term, differentiated Jacobian by output:

$$J = \sum_n (d_n - y_n)^2$$

$$\frac{\partial J}{\partial y_n} = -\frac{1}{2}(d_n - y_n)$$

The second term in the equation shows the power of the distal teacher learning. As the differentiation of the actual output y by the inverse model output u is impossible without mathematical description of the environment, we can replace the term with the forward model. This way the term can be calculated and gradient descent of the composite model can be achieved. So we replace the actual output y by the predicted output $\hat{y}$. Then the forward pass in the forward model is calculated.

$$y_n = f1(\sum_m v1_{m,n} \cdot g_m)$$

$$g_m = f2(\overbrace{\sum_k v_{k,m} \cdot u_k)}^{net_{gm}}$$

Where f_1 and f_2 are the activation functions of the output and hidden layer neurons respectively.

$$y_n = f_1(\overbrace{\sum_m v_{m,n} \cdot f_2(\sum_k v_{k,m} \cdot u_k))}^{net_{yn}}$$

$$\begin{aligned}
\frac{\partial y_n}{\partial u_k} &= f_1'(net_{y_n}) \cdot \frac{\partial}{\partial u_k} \sum_m v_{m,n} \cdot f_2(\sum_k v_{k,m} \cdot u_k) \\
&= f_1'(net_{y_n}) \cdot \sum_m v_{m,n} \cdot \frac{\partial}{\partial u_k} f_2(\sum_k v_{k,m} \cdot u_k) \\
&= f_1'(net_{y_n}) \cdot \sum_m v_{m,n} \cdot f_2'(net_{g_m}) \frac{\partial}{\partial u_k}(\sum_k v_{k,m} \cdot u_k) \\
&= f_1'(net_{y_n}) \cdot \sum_m v_{m,n} \cdot f_2'(net_{g_m}) \cdot v_{k,m}
\end{aligned}$$

The third term is typical backpropagation term, detailed derivation can be found in the textbooks (Haykin, 1998).

Hidden-output layer weights are adapted as:

$$\frac{\partial u_k}{\partial w_{j,k}} = f_1'(net_{u_k}) \cdot h_j$$

Input-hidden layer weights are adapted as:

$$\frac{\partial u_k}{\partial w_{i,j}} = f_1'(net_{u_k}) \cdot \sum_k \sum_j w_{j,k} \cdot f_2'(net_{h_j}) \cdot s_i$$

This way we have all necessary formulas for learning the inverse model.

3.4 Exploration strategies

The learning of the complex sensorimotor mapping by the distal teacher algorithm consists of repeating loop of prediction and correction. The model is provided with random initial state of the angles and the target position - desired position of the arm effector. If these pairs are chosen by random sampling, the exploration strategy is called the random goal exploration (SAGG-RANDOM). (Moulin-Frier and Oudeyer, 2013)

The exploration can be enhanced using more clever technique. Active goal exploration (SAGG-RIAC) chooses the target position according to the interest based on the previous errors. The distribution function of the interest is based on the performance error of the system, i.e. is indirectly proportional to the competence measure of the model. The higher is the performance error, the greater chance is that next target would be generated in proximity to the problematic region. This way the network gets more input from the problematic parts of the target space and the learning should be enhanced. (Baranes and Oudeyer, 2013)

Chapter 4

Implementation

4.1 Problem definition

Inspired by Jordan's and Moulin-Frier's simulations (Moulin-Frier and Oudeyer, 2013), we focused on the problem of reaching the target position in the 2D Cartesian space by robotic arm with multiple degrees of freedom. According to the empirical observations, the Cartesian target space was set as half plane in front of the arm in the shape of a unit circle.

The arm has five degrees of freedom with each segment shorter by half than the previous, together summing to one. The possible angle ranges on the arm are constrained to $[-\pi, +\pi]$ (full range). Note that for the simplicity, the segments can 'cross' each other in the space. The whole arm is not required to be on the working half-plane neither during starting nor final position, e.g. parts of the arm can lap over non-valid positions.

4.1.1 Representation of the state space

Name	Abbreviation	Domain
Joint angle state	$s(t)$	Joint space
Target position	$d(t)$	Cartesian space
Effector position	$y(t)$	Cartesian space
Predicted effector position	$\hat{y}(t)$	Cartesian space

Table 4.1: The list of the state space variables

The index t is the time step. For referencing values of the specific neurons of the network, notation $s(t)[i]$ for the i-th neuron is used.

4.2 Model design

4.2.1 Distal teacher implementation

The distal teacher learning system consists of two modules, inverse and forward model, both implemented by the two layer perceptron (see Figure 3.1).

Forward Model

The forward model creates a mapping from the joint angle state to Cartesian space, transforming current state of the arm's angles to Cartesian position of the effector.

Input

The network expects joint angle position in the range $[-\pi, +\pi]$ on the input neurons. The values are processed with population coding with 12 peak values uniformly distributed over the input range.

Parameters

The activation function on the hidden layer is bipolar sigmoid with range $[-1, 1]$. The output layer is scaled bipolar sigmoid on the range over $[-1.02, 1.02]$. The value is empirically obtained, based on the observation that the network had poor prediction on the edges of reaching (where one of the coordinates was approaching 1). It was improved by the scaling. The learning ability of the network is enhanced by the momentum (Haykin, 1998). Note that all the values were inspired by Jordan's simulation (Jordan and Rumelhart, 1992, p. 20) and enhanced by random sampling over the parameter space. They are fixed throughout all the simulations.

Parameter	Value
Hidden layers	1
Hidden neurons	25
Momentum	0.1
Learning rate (α)	0.1 (decreasing)
Hidden activation function	Bipolar sigmoid
Output activation function	Bipolar sigmoid

Table 4.2: Parameters of the forward model

Output

The output neurons encodes the Cartesian position of the effector.

Learning

The mapping can be trained under classical supervised learning algorithm. The forward model is provided by the learning pairs consisting of the J=joint angle state $s(t)$ and corresponding effector position $d(t)$. The data are generated using the random motor exploration paradigm. It means that the angles of joints are generated at random and the pair is accepted (provided to the network as a learning pair) if the effector position lies within the operating half-plane. The difference between the output of the network and target position from the learning pair is then calculated and used as a error for classical backpropagation learning algorithm. All details can be seen on the diagram. Note that there is no training set in a typical sense, as all training pairs are generated at random and have real number values, so the network is never trained on the same pair twice. The testing set used for error calculation is generated in the same way.

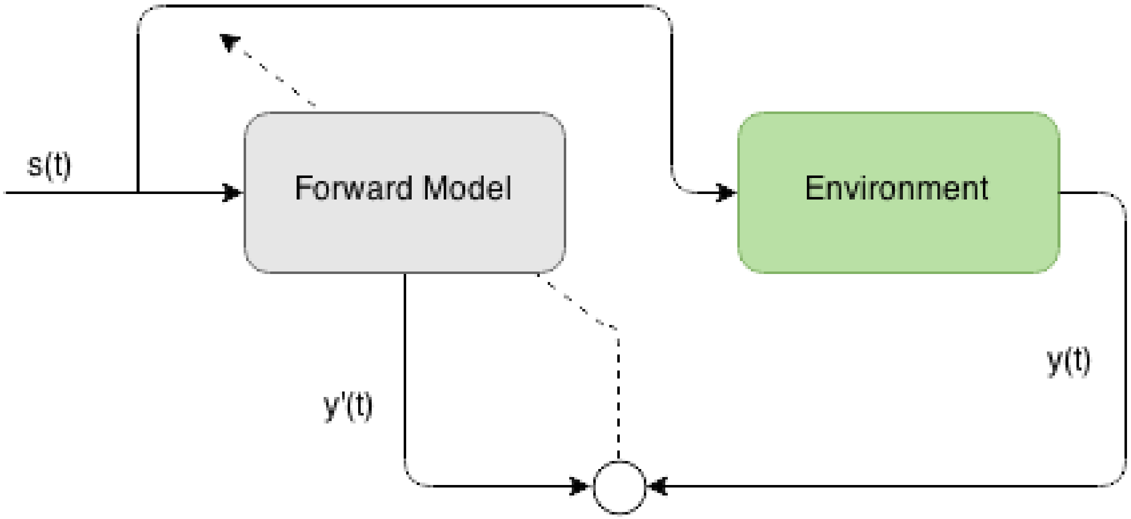


Figure 4.1: The forward model learning scheme

Algorithm 1 Pseudocode for forward model training

```
procedure FWDTRAIN
    state ← init
    while traning do
        environment ← state
        position ← environment.getCoordinate()
        model.load(state, position)
        model.propagate()
        error ← distance(position, model.out())
        model.learn(error)
        state ← randomAngles
    end while
end procedure
```

Inverse Model

The inverse model gets on the input position of the arm angles and desired target position and created mapping to joint position that leads the arm to the target position. Note that the output is an absolute angle position, not the change in angles.

Input

Each joint angle of the initial position is encoded by one neuron, the values are in range $(-\pi, +\pi)$. The target position is encoded by two neurons, each with value within the range $(-1, +1)$.

Parameters
The network has similar architecture to the forward model, so all the details are provided in the table.

Parameter	Value
Hidden layers	1
Hidden neurons	20
Momentum	0.1
Learning rate (α)	0.01
Hidden activation function	Bipolar sigmoid
Output activation function	Bipolar sigmoid

Table 4.3: Parameters of the inverse model

Output
The output neurons provides the absolute joint positions within the possible angles range.

Learning
The data samples used for learning are subject to the exploration strategy used. The strategy in fact determines only the target position. The system is ready and designed for learning reaching from variable starting position, but for the sake of simplicity, the starting position of the arm is fixed throughout all simulations. Various fixed starting positions were compared and all were showing the same pattern of the error landscape and error convergence, so arbitrary symmetrical position with the effector in point (0,1) - the arm stretched in the middle of unit circle was used, corresponding to joint angle position $(\frac{\pi}{2}, 0, 0, 0, 0)$. The pseudocode shows the variation of the algorithm, where the inverse model is trained with fixed pre-trained forward model. In fact, two approaches were examined and are analysed in the results. First one uses pre-trained fixed forward model (as Jordan suggested in his paper (Jordan and Rumelhart, 1992)) . The second one has the forward model that was pre-trained and then is fine-tuned simultaneously with the inverse model training.

4.2.2 Active Goal Exploration

The interest model is a distribution function over the Cartesian space of the arm effector positions. It is an inverted function of the competence or performance in a certain area, so the error function can be used as a measure or interest. Various approaches for modelling such a function were used, but the final network uses random sampling. Certain number of random targets are propagated through the network, the error is calculated and one with the largest error (highest interest value) is used as the training sample. Of course this approach is not practical in a sense of overall network time performance. If three points are used as a sample size for choice of a new target,

Algorithm 2 Pseudocode for inverse model training

```
procedure INVTRAIN
    $state$ ← init
    while traning do
        $target$ ← chooseTarget()
        invModel.load($state$, $target$)
        invModel.propagate()
        $environment$ ← invModel.out()
        $position$ ← environment.getCoordinate()
        $error$ ← distance($target$, $position$)
        invModel.learn($error$)
    end while
end procedure
```

the network makes three additional passes. If all these three points were subject to learning, the overall performance would be better (approximately three times) than if only the most interesting value is chosen. The purpose of this experiment is not to improve the overall performance of the system, but inspecting the performance of the network error landscapes and the speed of learning using active goal exploration.

Chapter 5

Results

The results are collected for two variations of the model, both described in the previous section. The overview of the models is provided in table 5.1.

	Phase 1	Phase 2
Model A	Train forward model	Train inverse model
Model B	Train forward model	Train inverse model Fine-tune forward model

Table 5.1: The variations of the model used for data collection

5.1 Training the forward model

The first step in training the whole system is the adjustment of the weights of the forward model. This phase is identical for both models. Random sampling of the arm joint positions is used for generating the training data, as described in the previous chapter. This phase is referenced as the first phase, the distal teacher algorithm is not used yet. The weights are adjusted by typical backpropagation algorithm.

The process of learning shows typical curve with fast increase in precision during first 100 epochs then the learning rate slows down and the overall error converges slowly, as seen in the Figure 5.1. Mathematically the model converges towards zero - the neural network works as a universal function approximator (Haykin, 1998, p. 372). However, this value was never obtained as the search for optimal parameters of the network is outside the scope of this thesis and is not really necessary for comparison of the efficiency of the algorithms.

The average error or the forward model after 2000 epochs, each consisting of 250 random initial positions, is typically in a range of 0.053 to 0.074. In the Figure 5.1 the error reached 0.062 distance units. This error can be considered satisfactory, as the arm's operating space is a half of a unit circle.

However, it is important to note that this error is not strictly relevant as a measure of precision of the forward model in the distal teacher system. When the forward model is integrated into the distal teacher system and first training and testing trials are performed, the error is dramatically increased, even doubled. The explanation is relatively simple. The training pairs (initial position, effector position) used for

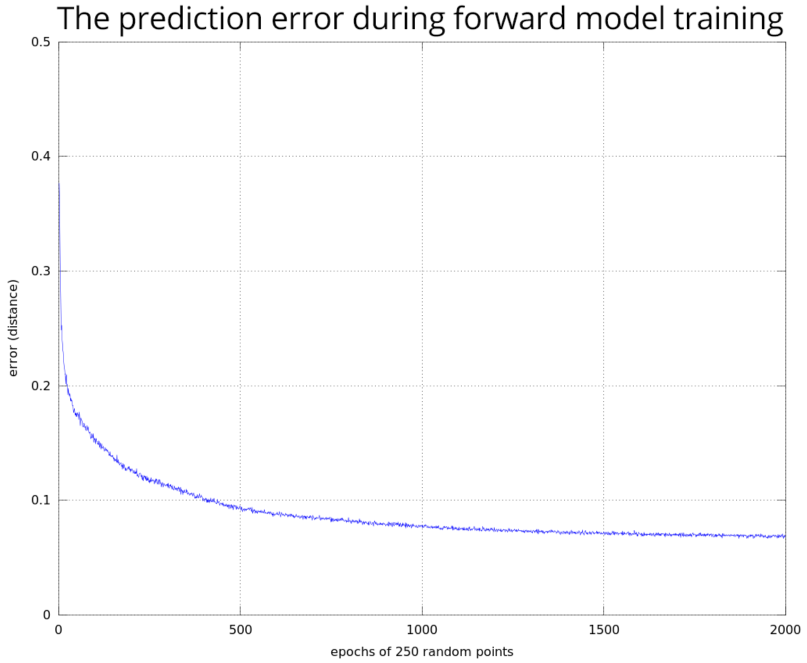


Figure 5.1: the Forward model error: the average values (with standard deviation $\sigma = 0.016$)

calculation of the error (difference between output position and real effector position) are sampled through the random generation of the joint angles, not effector points. This means that the effector positions are not randomly distributed over the unit half-circle but rather reflects the inner topology of the arm's most redundant positions - places that can be reached my most constellations of the joint angles. This distribution can be seen in the Figure 5.2. As these areas are more often used as a learning pair, the network performance in this areas should be better.

5.2 Training the inverse model

The inverse model starts learning after the first phase is finished and the forward model has a sufficiently low error. Two different variations of the algorithm are used - type A is the one with fixed forward model throughout whole training of the inverse model, the other, type B, uses pre-trained forward model too, but it is fine-tuned during the inverse model training on the same set of inputs.

The first set of simulations is run with a random goal selection. The target is selected from uniform probability function over the operating area. Each epoch consists of 250 targets.

The curve shows a typical pattern: fast decrease of the average error in the beginning and then slow fine tuning in the local error minima. This behaviour can be observed on the error landscape collected during the training. The grid is constructed

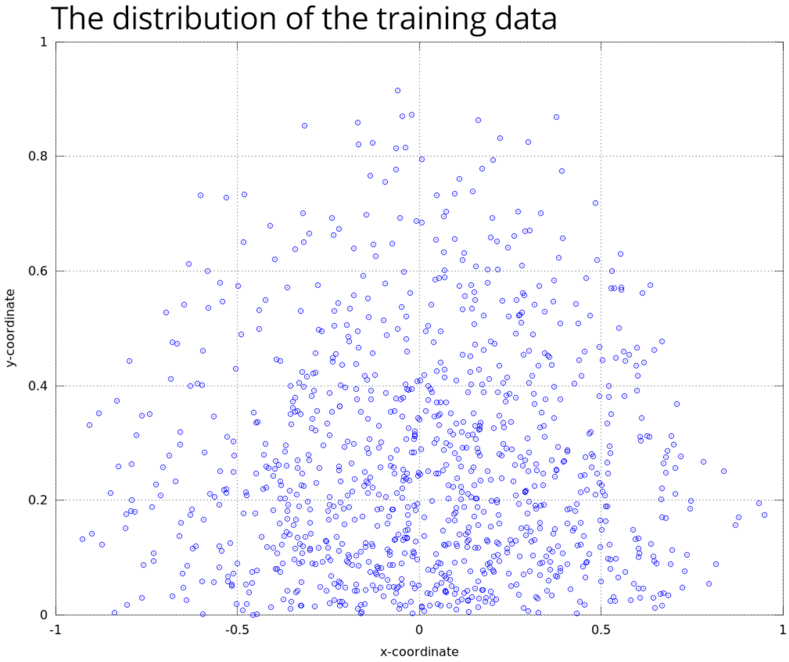
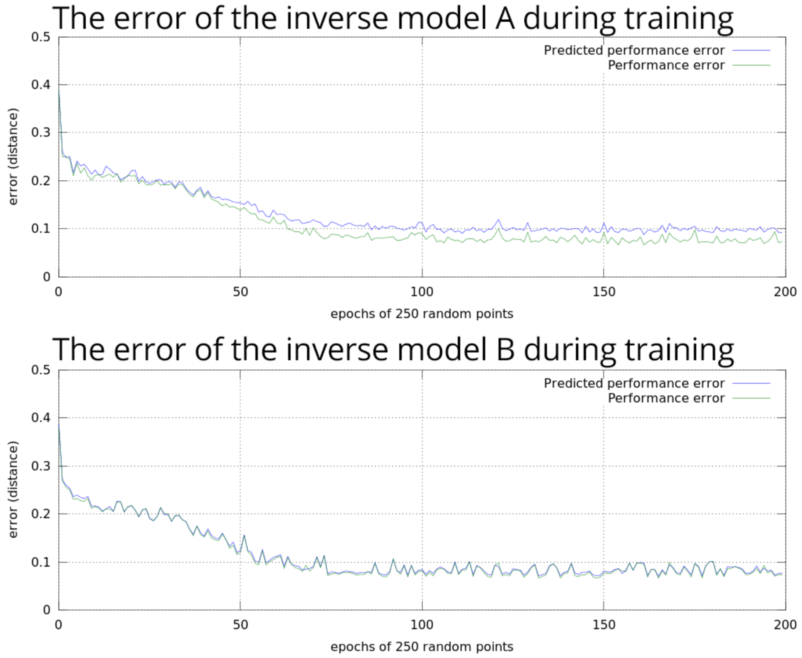


Figure 5.2: Density distribution of the generated points for the forward model in phase one

from values obtained by providing the network with fixed targets and interpolated. This gives a very good insight into the network's weak points. In the Figure 5.4 is used the type B's performance error. However, as the performance error is the same for both models, the error landscape for the performance error is also the same.

The inverse model of type A has higher value of the predicted performance error. This is consistent with the hypothesis in Jordan's paper (Jordan and Rumelhart, 1992, p. 7), that system can reach zero performance error (in this case not zero, but with lower value) even though the forward model is not perfect. This is natural, as the forward model is not perfect and the inverse model is trained on the error obtained from the real environment. As the inverse model is performing a gradient descent towards minimising the performance error, the performance prediction error is not necessarily decreased as the error landscapes are not necessarily identical.

The model of type B has the values of PPE and PE very similar, in fact, the correlation coefficient between performance error vector and performance prediction error is $r = 0.989$ consistently throughout the learning process (the average correlation for x and y vector coordinates separately). This can be caused by the fact that the forward model undergoes rapid learning in first epochs of the phase 2. In fact the error of the forward model drops to 0.016 ± 0.006 sampled through uniform distribution (note that this value is not effectively comparable with the value in section about the forward model training). However, this value is obviously small enough to provide almost perfect prediction. On the other hand, this model is significantly less stable.



The error of the inverse model A during training

The error of the inverse model B during training

Figure 5.3: The difference between the performance and the performance prediction error

Approximately 33% of the training sessions are unsuccessful, meaning that the descent on the error landscape does not occur and the error remains fixed around the initial error value (approximately 0.4 - 0.6).

The conclusion is that the performance error, which can be considered as the objective measure of the performance, remains approximately the same in both models. The behaviour of the performance prediction error is obviously directly dependant on the error of the forward model. The model type A is more stable and reliable in the terms of percentage of the successful learning sessions.

5.2.1 Active goal exploration

Following sets of simulations were performed with type B model using the active goal exploration. Before choosing each target for the network, 10 random points were sampled, propagated through the network and the one with the greatest error was chosen to be the target. One epoch then consisted of 250 learning targets, as in previous case (even though 250 times 10 points were considered).

According to the graph in the Figure 5.5, the initial decrease of the error is more rapid with the active goal exploration and successive convergence reaches the same values after approximately 50 epochs. This means that the overall performance error after the training of the distal teacher model is not decreased, but the system is trained faster using the active goal exploration. When comparing the learning curve of the 3

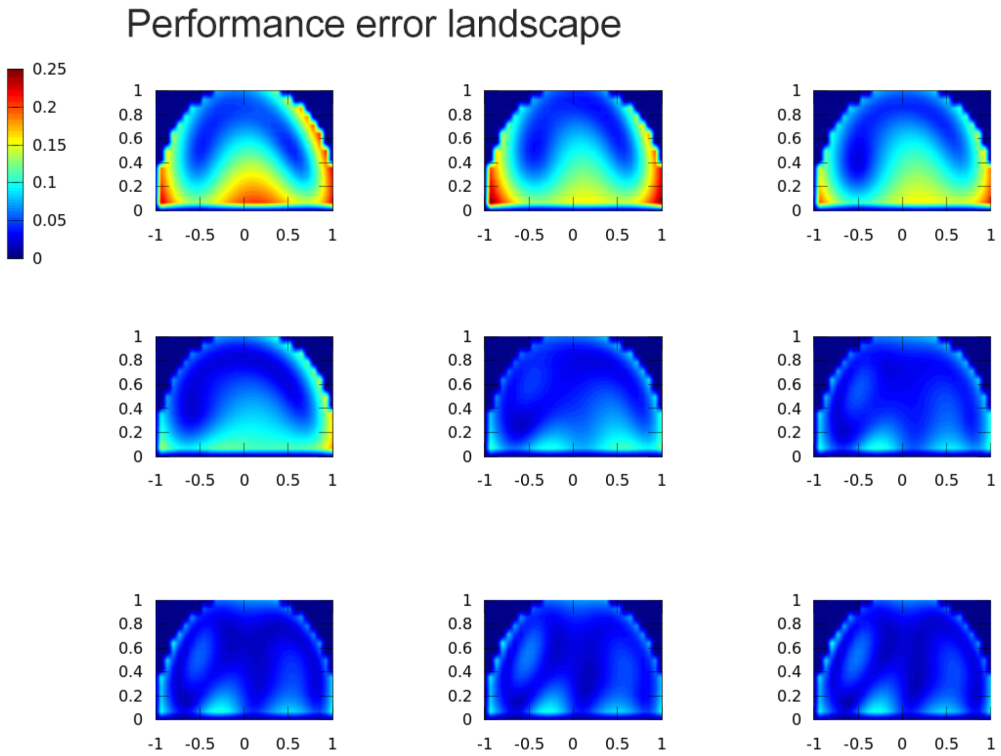


Figure 5.4: The performance error landscape of the first 90 epochs sampled by 10 of the type B model training

and 50 point sampling it is obvious that the better is the active goal exploration strategy, the faster is the learning. The error landscape is also different from the random exploration, as in the first critical epochs, when the rapid decrease occurs, the error in the most problematic parts (on the edge of the operating half-plane and right in the middle of a unit circle) is decreased faster. It is exactly as expected, as the inverse model was provided with more targets located in these problematic areas.

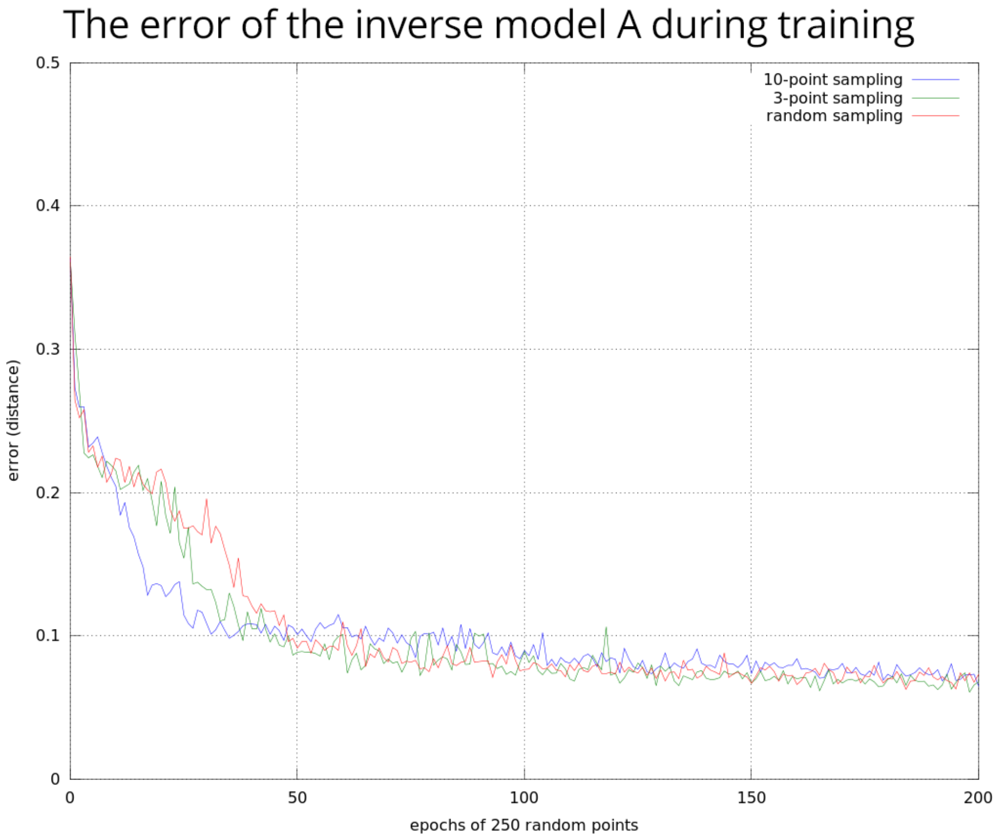


Figure 5.5: The performance error of the inverse model with active goal exploration using different number of sampled points

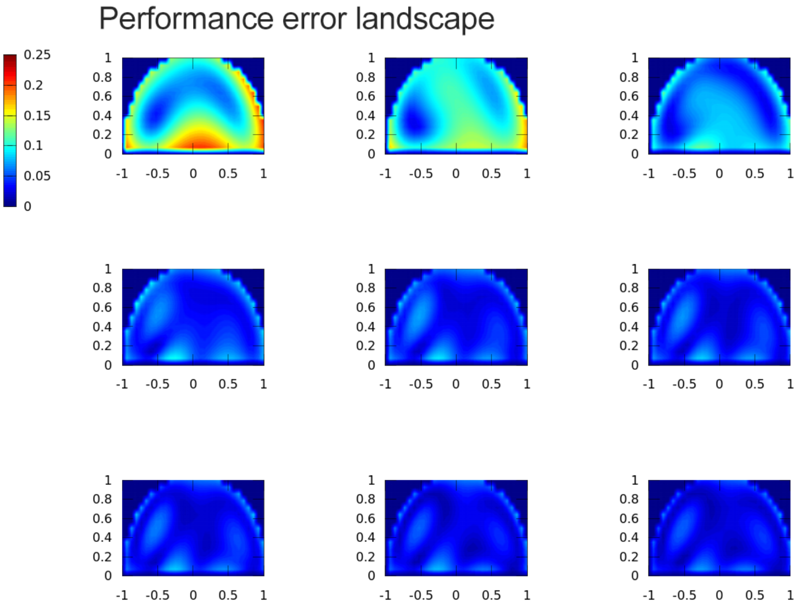


Figure 5.6: The performance error landscape of the first 90 epochs sampled by 10 of the type B model training using active goal exploration with ten sampled points

Chapter 6

Conclusion

This thesis is focused on analysis of the distal teacher algorithm. It is an algorithm under the supervised paradigm using the forward model and the inverse model together in order to learn non-unique mappings. The good example of such a mapping is the reaching problem - simulation of the 2D robotic arm with 5 degrees of freedom operating over the unit circle half-plane. The forward model in this case served as a model of the arm itself - providing mapping from the joint angles position (observable by the robot itself) to the Cartesian coordinates of the arm's effector in the plane (necessary to be observed from the environment). The inverse model was trained to output correct action (absolute joint angle position) in order to reach the input target position from input starting position. For the sake of simplicity, all the data collected were from the simulations, where the fixed starting position was used, but according to empirical experience, the model can be extended to operating from arbitrary starting position (even from flexible starting positions). The whole model is in fact designed for this purpose. All the formulas used and the architecture of neural networks are easily applicable to the extended model.

The distal teacher algorithm can be considered as capable of learning the reaching problem. After initial fast decrease in the performance error the system converges towards values with relatively high performance precision. The decrease in error is even faster when the active goal exploration is used.

Two variants of the algorithm were tested, one with fixed forward model, the other with simultaneous learning of the pre-trained forward model and the inverse model. The performance error was the same for both types. There are hints that the causality between the forward model error and performance prediction error exists.

The weakest point of the model is implementation of the active goal exploration. Instead of using sophisticated approaches as suggested by the paper of Moulin-Frier and Pierre-Yves Oudeyer (Moulin-Frier and Oudeyer, 2013), such as Gaussian Mixture Models for modelling the competence (inverse of error) distribution function over the operating space, very simple and inefficient random sampling was used. The initial attempt to use the self-organising maps for mapping such function (Haykin, 1998) failed on practical issues - even though mathematically the SOM should be able to create representation of the probability function, the optimal parameters for learning were never found and no significant results were obtained. The random sampling is ineffective, biologically non-plausible and increases overall time performance of the network (as additional passes through the inverse model are required in order to obtain the error). On the other hand, even this approach is sufficient to demonstrate the superiority of the active goal exploration in the distal teacher learning.

The further research in this problematics should lead to optimising the active goal exploration and extending the abilities of the inverse and forward model - ability to output action (change in joint position), not absolute joint position and ability to predict following Cartesian position given actual position and action, respectively for the models. The result should be examined for the variation of the training pairs, where the initial position of the arm is not fixed, but fully flexible. The capabilities of the distal teacher algorithm should be further tested on even more difficult mappings, such as multiple degree of freedom in 3D space.

Bibliography

Baranes, A. and Oudeyer, P.-Y. (2013). Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robot. Auton. Syst.*, 61(1):49–73.

Dawkins, R. (2006). *The Selfish Gene: 30th Anniversary Edition*. ISSR library. OUP Oxford.

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition.

Jordan, M. I. and Rumelhart, D. E. (1992). Forward models: Supervised learning with a distal teacher. *Cognitive Science*, 16:307–354.

Lalazar, V. (2008). Neural basis of sensorimotor learning: modifying internal models. *Current Opinion in Neurology*, 16:307–354.

McKerrow, P. (1991). *Introduction to robotics / Phillip John McKerrow*. Addison-Wesley Pub. Co Sydney ; Reading, Mass.

Moulin-Frier, C. and Oudeyer, P.-Y. (2013). Exploration strategies in developmental robotics: a unified probabilistic framework. In *ICDL-Epirob - International Conference on Development and Learning, Epirob*, Osaka, Japan.

Nguyen-Tuong, D. and Peters, J. (2011). Model learning in robotics: a survey. (4).

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1988). Neurocomputing: Foundations of research. chapter Learning Representations by Back-propagating Errors, pages 696–699. MIT Press, Cambridge, MA, USA.

Vijayakumar (2007). Lecture xi: Learning with distal teachers(forward and inverse models).

Wolpert, Ghahramani, F. (2001). Perspectives and problems in motor learning. *Trends in Cognitive Sciences*, 11:487–494.