論文 / 著書情報
Article / Book Information

題目(和文)	認知ロボティクスのための脳型知能情報処理機構
Title(English)	Cortically-Inspired Mechanisms of Computational Intelligence and Learning for Developmental Robotics
著者(和文)	TarekNAJJAR
Author(English)	Tarek Najjar
出典(和文)	学位:博士（工学）, 学位授与機関:東京工業大学, 報告番号:甲第10229号, 授与年月日:2016年3月26日, 学位の種別:課程博士, 審査員:長谷川 修,新田 克己,渡邊 澄夫,宮下 英三,小野 功
Citation(English)	Degree:, Conferring organization: Tokyo Institute of Technology, Report number:甲第10229号, Conferred date:2016/3/26, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Tokyo Institute of Technology

Doctoral Thesis

Cortically-Inspired Mechanisms of Computational Intelligence and Learning for Developmental Robotics

Author:
Tarek Najjar

Supervisor:
Osamu Hasegawa

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Engineering

in

Computational Intelligence and Systems Science

March 2016

Declaration of Authorship

I, Tarek Najjar, declare that this thesis titled, 'Cortically-Inspired Mechanisms of Computational Intelligence and Learning for Developmental Robotics' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

TOKYO INSTITUTE OF TECHNOLOGY

Abstract

Interdisciplinary Graduate School of Science and Engineering
Computational Intelligence and Systems Science

Doctor of Engineering

Cortically-Inspired Mechanisms of Computational Intelligence and Learning for Developmental Robotics

by Tarek Najjar

The Cerebral Cortex is the most competent learning mechanism ever known. With its vast complexity it makes for an invaluable resource of insights that would takes us a step further toward decoding the essential key elements of learning systems. On the other hand, Developmental Robotics tries to benefit from the dynamic course of intelligence-formation in biological systems in order to achieve robotic systems that are more competent than traditional ones.
This study is about investigating Computational Neuroscience findings in studying the human brain, in order to build better Machine Learning Algorithms and robot control architectures that would realize a developmental robotic platform for achieving, as close as possible, an Incremental and continues path of learning, thus enabling robots to learn tasks that were not preprogrammed in advance.
A new robot control-architecture for learning through exploration is proposed together with a novel neural network that is inspired by cortical mechanisms of learning and data representation. The validity of the proposed mechanisms were tested and demonstrated through multiple experimental setups from both robotics and real-world regression problems. ...

Acknowledgements

I would like to express my sincere gratitude to my advisor Prof. Osamu Hasegawa for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.
I would also like to thank the rest of my thesis committee for their support.
I offer my enduring gratitude to the faculty, staff and my fellow students at Tokyo Institute of Technology, for their help, support and all the good times.
I would also like to thank my parents. They were always supporting me and encouraging me with their prayers and best wishes.

...

Contents

List of Figures

Abbreviations

SOINN	**S**elf-**O**rganizing **I**ncremental **N**eural **N**etwork
RF-SOINN	**R**eceptive **F**ield **S**OINN

Symbols

θ	angle
L	location
m_{cp}	mean error in close past
m_{fp}	mean error in far past
μ	expected value
σ	variance
η	learning rate
r_i	post-synaptic firing rate
r_j	pre-synaptic firing rate
E_k	learning experience
S_{input}	input data-space
C_{in_j}	input-space-component node
S_{output}	output data-space
C_{out_i}	output-space-component node
w	connection weight

Chapter 1

Introduction

1.1 Robot Intelligence

Since Asimov's three laws of robotics, and the question of defining the robot's cognitive and behavioral architecture, is still opened and unsolved. The reason behind this is the fact that we want from robots to be part of our world, to perform tasks that originally intended to be performed by us, and to do these tasks without much of our presence or participation. All these requirements made it very difficult to come up with the optimal approach to robot intelligence.

When Alan Turing in 1950 first talked about his "Imitation game"[1], which we call now "The Turing Test", he, at that time, made an early vision of machine intelligence and the features of its elements[2]. In fact the research on AI has been greatly influenced by this early vision[3]. but yet, no currently-known approach

has proved to be robust and reliable enough for every day situations. One of the reason behind this, is the fact that we constantly attempt to come up with schemes and concepts that represent human intelligence and problem-solving mechanisms and then design robots according to these mechanisms despite the fact that we haven't fully understood the way the human brain really operate[4].

With the introduction of the concept of embodiment, during the eighties, much of the features and characteristics of Turing's true intelligence became more dynamically emphasized. Behavior-based approach to robotics, introduced by Rodney Brooks[5], became one of the first practical formalization of this embodiment of intelligence[6], and this is how embodiment got its place and significance in AI research[2].

1.2 Classic Robotics

Approaches and attempts toward Intelligent Robot realization could be divided, roughly, into the following groups[7, 8]:

- Direct Programming Approach
- Behavior-based Approach
- Supervised Learning Approach

1.2.1 Direct Programming Approach

In this approach a human engineer must envision and imagine the task that the robot will execute. This task must be fully understood and defined by the human engineer. After that, the engineer must define concepts, representations, and mechanisms for solving the given task in hand. Only then this task, together with its representation and algorithm of solution, will be "spoon-fed" into the robot. It is obvious here that the robot actually does not show any intelligent policy in reacting to the dynamic world around, we notice that it is not the robot's sensors that formulate representations for world objects and tasks but rather the engineer, the robot is just executing what the engineer ordered it to do, in a blind manner. So could we claim this as machine intelligence? beside there is one more problem to this approach: when we are providing the robot with explicit concepts and representations, and when we define manually the entities that the robot will have to encounter and deal with in its operational environment, can we claim that we are able to provide the robot with everything he need to know? Let's put in other way: are we able to write a code that covers and define every concept the robot might encounter if it will operate in the complex world that we live in? The obvious answer is simply No. What if the robot will encounter a situation that the human-written software does not cover? The robot will simply fail.

1.2.2 Behavior-based Approach

In this approach, as envisioned by Brooks "The Subsumption Architecture "[5], instead of modeling the Task, we model the behavior and organize it in layers. This would, of course, lead to more complex and adaptive behavior, but it actually has never been practically proven to be capable of general and reliable Intelligence.

1.2.3 Supervised Learning Approach

This approach is considered as a significant improvement over direct programming, but again we notice a strong presence of the human engineer in the picture, because in this approach the problem to be solved is represented to the robot as correlation and mapping between given sensory data and desired behavior on the robot side, after that the robot is supposed to be able to go a step further beyond what has been given to him and generalize, as much as possible, for situations not exactly identical to the ones provided by training samples.

The problem here is that the amount of work that is required on the engineer side is considerably enormous. Beside, again we notice that it is the engineer who decides how to solve the problem and not the robot, and again the whole problem is seen by the engineers's eye and from his point of view and prospective, which actually limit the ability of the machine to adapt and does not allow for continues learning.

1.2.4 Does it Really work ?

In all of the approaches mentioned above we encounter a common problem that limits the ability of these methods[9]; We notice a strong presence of the human engineer during the problem-formalization phase. It is clear that it is always the human who define and shape concepts to be adopted by the robot during its operation. What if these concepts fail, as they usually do, to represent a realistic component of the actual complex world?

What if these concepts don't really reflect the way the robot experiences the world through its own sensors and motors?

Well, eventually the robot will fail to perform in a way that is considered rational in the current situation. The reason behind this is that the robot wasn't given the chance to generate these concepts the way it experiences them in the environment. In other words, the engineer made these concepts and force them to the robot but from the engineer prospective and point of view, and in a way that reflect how he understands and perceives these concepts, but what if this is not consistent with how the robot experience them through its own body and perceptual ability?

A nice example of this problem is found in[9]; Let's imagine a mobile robot that is designed to navigate and avoid obstacles. If we define the concept of "obstacle" as a short reading on a laser range sensor, and then wire this concepts into the robot controller without enabling it to really understand and experience it or to understand why an obstacle must be avoided, then the result is that when this human-made concept fail to reflect a real-world situation, the robot won't be able

to perform rationally, and the robot would treat a grass as a possible obstacle that must be avoided as if it was steel rods.

The reason behind this is that the robot didn't really generate and learn these concepts on its own through its experience. Thus it didn't really test these concepts.

This problem is formally defined as "The Verification Principle", that was first introduced by Richard Sutton[10]: In short, this principle states that an agent cannot learn a concept unless it is able to test and verify this concepts on its own.

1.3 Muddy Tasks

Today's computers are characterized by advanced computational power so that complex mathematical calculations, that would take a very long time by human to solve, are done in a split second.

Database search and retrieve operations are one of the most trivial tasks that computers can do with an absolute efficiency. Data analysis and processing would have never reached its current level today without the use of computers. These are just a small amount of the tasks that computers can do efficiently and easily and that are considered to be very difficult and tedious for human. On the other hand. we ,surprisingly , find tasks such as vision, natural language understanding ,emotional inference and understanding ,that are very easy and intuitive for human, to be extremely difficult to be performed by computers and robots[2]. These tasks are unstructured, undetermined with a solid pragmatic approach to solve

and don't have a clear and consistent mathematical and computational definition. These tasks are what we might call "Muddy Tasks"[11] because they are muddy in both algorithmic and computational sense, where we cannot define a predetermined and coherent steps and mechanism for solving and performing them, and we cannot provide a clear and structured approach to solving them. These tasks are characterized by rawness of sensory input, richness of input space, and the lack of ability to be described in mathematical terminology[2]. Actually such muddy tasks represent both a challenge for engineers and robotists, besides it makes an obstacle on the way of robot fusion in everyday life since they are crucial for robot integration in our environment and essential for the realization of efficient robot utilization in the real world.

Unfortunately current AI techniques proved to be unable to make robots perform these tasks successfully.

Most of conventional AI methods are based on symbolic representation and manipulation which, of course, require providing the machine with predetermined and hand-made concepts that don't reflect the machine perspective, as mentioned in earlier section. These symbol based methods couldn't so far produce robots that can learn on their own or understand English for instance. That is why a different approach must be adopted for robot intelligence. An approach that eliminate the need for human-made symbols and concepts to be part of the internal and mental state of the robot. An approach that enable the agent to experience the environment on its own using its own sensors and actuators[12] in order to learn and

adapt in its environment where it can generate its own experience and learn from it through interaction with the world around, and where it could autonomously shape its strategies and internal state in continuous manner[13].

This leads us to the topic of Developmental Robotics.

Chapter 2

Developmental Robotics

2.1 The Theory of Cognitive Developmental Robotics

Both inspired by developmental psychology and cognitive neuroscience, a new field in robotics has emerged which is the field of developmental robotics[13, 14].

This multidisciplinary field combines developmental child psychology, cognitive neuroscience , artificial intelligence and robotics[8, 15].

The basic concern in this discipline is to formulate embodied artificial agents with features of semi-continues cognitive development[16],which is the ability of the agent to adapt and grow "mentally" in the way it perceive, represent and process its experiences and the way it acts in the world around.

This development must take place through interaction with the environment, using the agent's own sensors and actuators, in a continuous manner[17].

Initially the system is equipped only with a core developmental program[18] which is simply an engine for attention, exploration and interaction, and then during this interaction with the environment more complex cognitive structures would emerge[9].

Influenced by the agent's experiences, these cognitive structures encompass the agent's concepts, representations and strategies for its behavior. It is essential for this type of agents to be characterized by embodiment, situatedness and an extended process of development[19].

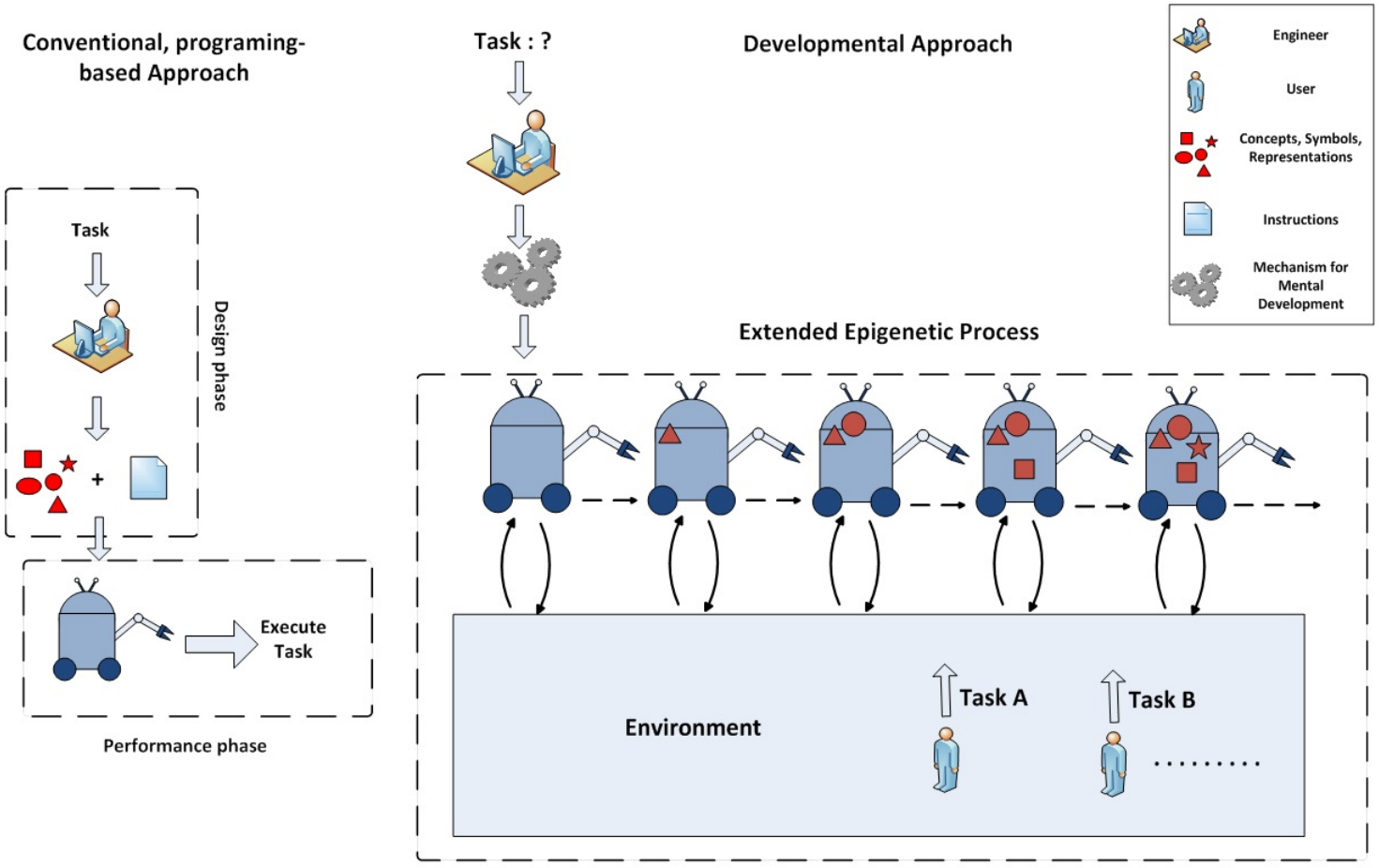

FIGURE 2.1: developmental robots learn incrementally through interaction with the environment[2].

So instead of manually providing the robot with human-made concepts and strategies, we let the robot learn on its own, driven by its developmental program, and that is by experiencing the world around and exploring its aspects inspired by the way a child learns by playing and trying things around him[20]. That is why this field tries to draw from the theories and finding of piagetian view of child development[21] which was founded by the psychologist J.Piaget during the fifties of the twenty century.

Unlike conventional robots, that are programmed with a task-specific controllers for performing a given task which is previously known and predetermined by the engineer, developmental robots supposed to be able to learn most tasks taught by a human through their operative service time[22], of course given that the robot is equipped with the sensors and actuators required for learning the task being taught.

In other words , when the engineer builds the robot he does not know previously the tasks that the robot will learn ,but rather provide the robot with a developmental program that would initially learn primitive tasks and representations and then incrementally the robot would build on these previously learned tasks in order to learn new, more complex ones[23], Figure **??**. So, for example, if the robot learns how to fixate its gaze on a given object, then using this skill it learns to visually follow objects in its visual field. After that if it learns to control its arm actuators, then eventually, and by reusing all these previously learned skills, it would be able to learn how to reach and grasp various objects in its environment. So it all

depends on, and relates to, the continuously-adapted and acquired concepts and representations that make up the experience and the internal state of the robot.

Remarkably, the field of developmental robotics does not only serve the goal of building better and smarter robots, by using biologically inspired algorithms and mechanism, but also it has the potential of being a possible test-bench for theories and modules of cognitive neuroscience and developmental psychology[24].
So both of engineers and scientists can have their prospective and use of this field; An engineer find in it a way for developing more intelligent robot controllers, while a psychologist could use it as a tool for testing possible modules of intelligence[25–29].

2.2 Related Work

Research in developmental robotics has been given a notable attention recently[13, 14]. Many robotists are trying to come up with an architectures that relate to developmental learning. Some researchers started by providing a robot with a motivation mechanism for exploration and then let the robot explore the action space driven only by infernal measures of motivation. The approach taken in[30] provides a sophisticated mechanism for intrinsic motivation for which the system is driven toward regions of sensory-motor space with the most possible learning progress. But the problem by basing a learning mechanism on motivational drives, is that the exploration path is unpredictable. In another research[31] a system

that used a knowledge gain measure of how "interesting" a given action is, was proposed. But this system did not count of open ended learning. In another research[32] we find that the error of anticipation of possible action consequences is used directly as a source of motivation.

Others tried to make the robot learn starting from hard-wired primitive tasks. For example a work done by the team in[33] used reinforcement learning for learning compound skills. The problem was that the system was built to learn a pre-specified skill by the programmer. The work of group [34] exploits reinforcement learning techniques in order to overcome some of the limitations posed by stochastic optimal control approaches to motor-learning in robotics. Other researchers[35] managed to demonstrate successfully the ability to learn complex motions, but there was explicit use of domain-specific knowledge in the algorithm. Yet another approach was taken in[36] for task abstraction, but goals of these tasks to be learned were known in advance. An interesting result can be found in[37] where a robot learned to develop complex tasks using reinforcement learning techniques, but again the incorporation of pre-programming and primitives hard-wiring was persistent.

Learning by exploration was also adopted for implementing the process of motor babbling. Actually many roboticists have attempted to mimic this developmental process using humanoid robotic platforms. An example is found in the work of the group[38], here a gradient descent method is used in order to enable the system to learn some of the unprovided elements of the system's kinematic model

where the rest of the elements were already provided and pr-eprogrammed. A more efficient approach than gradient descent was taken by group in[39] where the system starts off by a population of candidate possible models then, and through interaction with environment, the system evolve in approximating a more accurate model that represents the system in hand. Beside the explicit dependency, in this system, on artificial visual tags that are attached to segments of the robotic arm, this approach make use of Bayesian learning and Gaussian regression, their algorithm actually is very expensive on the computational side. A rather different approach was taken by the group[40]. Here a camera calibration based method were adopted together with open-loop mechanism for generation of an implicit body schema model, this system made use of look-up table learning mechanism which naturally requires longer time for learning. The research group in[41]used a more biologically inspired approach by incorporating concepts like population code and equilibrium-point hypothesis in order to enable the system to achieve reaching tasks. In a different approach[42]the research group used both Bayesian belief functions and social learning mechanisms to facilitate learning-by-imitation competence. This approach, too, made use of hard-wired motor primitives that were encoded manually into the robot. A reinforcement learning approach together with imitation methods using locally weighted regression was facilitated by[43], where a robot was taught specific motor primitives, that are specific to given task sittings, then the robot generated policies that enable it to learn those primitives in an episodic manner. Although the robot managed to perform the given tasks but it seemed like the system was kind of a task-specific oriented in the way it

learned each motor primitive.

As for the learner itself, research for achieving systems capable of continues incremental learning can be found in the work of the group[44], where a hierarchical architecture is employed in order to provide human-like motor abilities. Although their method has many advantages but the fact that learning was based on symbolic representation has imposed some constrains on the task-scalability of their approach. Another example is the work done by[45], in which probabilistic-entropy techniques are used to enable a neural network to evolve in structure while learning a regression tasks on noisy data. Their network was trained on a data set that was provided manually to the learner beforehand of testing. Some researchers[46, 47] managed to aid learning through the incorporation of internal reward functions as biasing-drives for the learner. Such techniques usually tend to push the system toward exploring salient features of the data-space on the basis of some criteria of measured progress. While incorporation of such internal drives proved to be effective in making the system capable of learning tasks that were not account for by the designer, but at the same time it was not completely clear how to control the trajectory of learning that system is driven into. In other words it is not clear how to get the system to learn a given predetermined task. An astonishing results was obtained by the work of the group[48]. The developed network was capable of unsupervised learning of high-level concepts, like human faces, without the need for the data to be labeled for the learner. The robustness of the method is appealing but the huge amount of data and computation that is required for the feature

detectors to be learned, is what constrained the ability of the system to operate in real-time conditions.

More biologically-inspired approach was taken by the research team [49], through the exploitation of Hebbian conditioning theories in order to facilitate associative learning within cross-modality perception, resulting in a system that incorporate multimodal cues for perception and object recognition. Similarly, another group[50] investigated hebbian learning process in order to build neural network architecture that is characterized by being dynamic and non-rigid in structure. In their work the process of excitation and inhibition shapes the internal organization of the network by a interactive process of generation and elimination of individual nodes, within the network, in respond to external stimulus.

A new neural network model is proposed by the group [51]. Their network focuses on subspace learning by relaying on hebbian and anti-hebbian local learning rules. Although their algorithm is considered biologically-plausible but it is not clear whether it is possible to implement their network as incremental learner for learning from the environment scenarios.

An example of self-organization can be found by the work of [52]. They proposed an algorithm based on a combination of self-organizing maps and state vector machines in order to facilitate time-series forecasting. In spite of the interesting outcomes of the proposed mechanism, but the investment of state vector techniques imposes some complexity issues due to the extensive computation usually involved in such techniques and the need for parameter selection. Another group [53] presented a self-organizing neural network model that is based on rewarding

techniques with intrinsic plasticity. Clustering self-organizing map is presented by the work of the research team [54], where data distribution is detected by the network through dynamic two-dimensional topological graphs that investigate the implicit relationships within data clusters. Although their algorithm facilitates incremental online learning, but most of their experimental results were based on artificial data sets.

2.3 Thesis Contributions

The work in this Thesis contributes various robot learning algorithms to facilitate the construction of robotic architectures capable of developmental learning.

Besides, a theoretical formulation of the developmental approach to robot cognition has been stated.

Finally, an investigation of various cortical mechanisms of learning and information processing has been done by taking insights from the findings of computational neuroscience and then designing machine learning algorithms to aid incremental on-line learning.

2.3.1 Theoretical Formulation

- A review of the question of robot intelligence and the various approaches to its implementation within realistic operation condition has been made
- Limitations of classic approaches to robot intelligence has been identified.

- A definition and identification of the main aspects of the theory of Developmental Robotics has been made.
- The principle of Motor Babbling and its application in robotics has been identified.
- Hebbian Plasticity mechanism has been incorporated in building neural network architecture.
- The concept of Receptive Fields has been used as a mechanism for data coding and representation.

2.3.2 Practical Achievements

- Proposing a novel architecture for online sensory-motor learning using self-organizing maps without the need to provide the system with a kinematic model or a preprogrammed joint-control scheme.
- Adopting Motor Babbling mechanism for constructing learning-through-exploration architecture.
- Providing mathematical formulation that governs the balance between exploration and and learning in developmental robotics.
- Construction of a mechanism that can compensate for unanticipated structural changes and alteration in robotic platforms

- Modification of original SOINN neuron activation function in order to facilitate non-linear data coding and representation.

- Construction of novel network for online, incremental learning through the incorporation of Cortically-inspired mechanisms as Self-organizing maps, Receptive Fields and Hebbian Plasticity.

2.4 Research Goal and Objectives

In order to build robots that can be practically utilized in our daily life, we adopt the developmental approach to robot intelligence and learning. So The major goal for this research is to construct robotic architectures and learning algorithms that can implement the paradigm of Cognitive Developmental Robotics in order to enable robots to learn, interact and to be fused with the world that we live in.

The path that we took to achieve this goal is summarized by investigating the studies and findings of Computational Neuroscience research in order to have insights into the mechanisms of learning and information processing within the Cerebral Cortex, and thus to use these insights as hints for designing algorithms that achieve the goal stated above.

2.4.1 Objectives

The main objectives of this thesis are:

- To propose a mechanism for autonomous mental development as the central cognitive architecture for information-representation, sensory-motor learning and interaction with the environment in developmental robotics.

- To implement a computational process for achieving adaptive systems that can react and compensate for unexpected structural alteration during operation.

- To design a novel neural network machine learning algorithm that is capable of online,incremental learning needed to implement developmental robotics.

Chapter 3

Learning Through Babbling Architecture

3.1 Introduction and Motivation

In order to put the robot on a consistent path of competence and intelligence development, it is essential to start off by acquiring abilities of a mere primitive nature of intelligence so ,later on, the system would build on these essential skills in order to bootstrap into tasks of higher intelligence.

One of the most essential and fundamental milestones of Development of Intelligence is the ability to use one's own body and to actuate in the evironment[21]. Teaching a robot to control its arm joints, in order to achieve reaching tasks, without explicit computation of forward and inverse kinematics, poses multiple

challenges on the control system. One of which, is the problem of enabling a robot to, autonomously, generate an internal representation of the control system given the fact that neither explicit kinematic model nor a transformation mechanism is provided by the designer. This means that the learning system must adaptively and incrementally build the required internal representations in order to guide the control process for multi-joint robotic arm in order to achieve goal-directed reaching motions.

Motor Babbling seems as a promising approach toward the generation of internal models and control policies for robotic arms.
In this chapter we propose a mechanism for learning sensory-motor associations using layered arrangement of Self-Organizing Neural Network (SOINN)[55] and joint-egocentric representations.

3.2 Sensory-motor Learning Architectures for Developmental Robotics

The most common approaches to implementing learning architecture for developmental robotics can be categorized into the following three approaches: Candidate model search-space approach, Body schema approach and Motor primitives with reinforcement learning approach. Of course this list is not exclusive as many other approaches do exist, but we chose to review these three as they are the most common ones.

Candidate model search-space approach is based on providing the robot with vast space of possible kinematic and control models, these candidate models are preprogrammed manually into the learning architecture, then through the incorporation of space-search techniques and during the actual interaction with the environment the robot would eliminate those models that are unlikely to reflect the actual structure of the system and then eventually would embark on a model that is as close as possible to the system, Figure 3.1

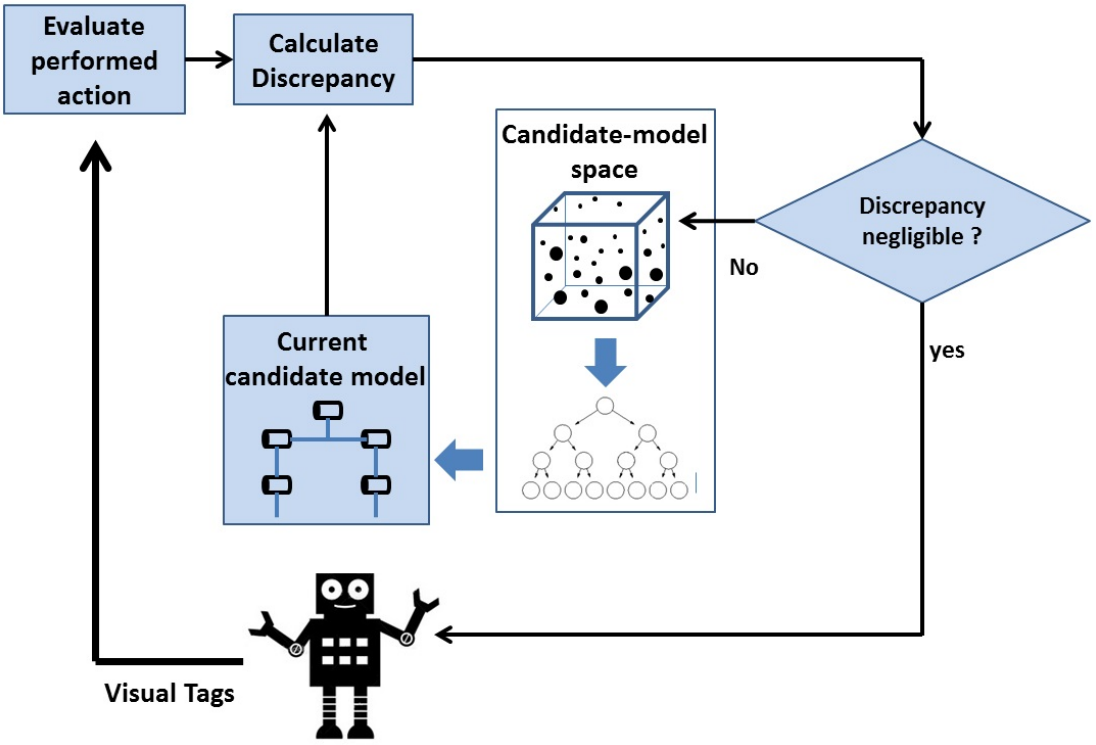

FIGURE 3.1: Candidate model search-space

At any given moment the architecture would settle on a given candidate model, that is expected to provide a competence of controlling the system, but when the robot actuates in the environment, the expected results of each action is matched

against the the actual results of the performed action, then a measure of discrepancy is generated. This measure of discrepancy drives various heuristic functions into various sub-regions of the candidate model search-space until an appropriate model is found.

Usually such approach requires embedding certain artificial visual tags into the physical robot structure. These visual tags are used to track trajectories and configurations of the robot as no description of the kinematic chain is explicitly provided to the system.

This approach toward learning the systems model makes heavy use of search algorithms which gives the system sort of brute-force characteristics. Besides, both the complexity and size of the space could hinder the real-time feasibility of such approach. In fact it is not guaranteed that searching the model space would result in finding a model that would accurately reflect the system at hand. Beside, generating the model space itself is another question that needs to be answered, taking into account the potential size of the resulting problem to solve in case of adopting automated method of generating the space. In contrast to the architecture we are proposing, model space approach seeks predefined and rigid paths of learning and exploration as solution trajectories are usually constrained by the way the search-space is generated and because the system is not truly exploring the surrounding environment, as in out approach, but rather it is exploring its own internal space of possible solution models.

The approach we are proposing in this thesis enables the robot to adapt and compensate for unexpected structural alterations and changes to the physical system

itself during operation, but it is not obvious whether model space approach can provide such ability with its rigid and environment-isolated space of exploration.

The second approach toward building sensory-motor learning architecture is body schema approach. This approach is based on manually providing the system with a general purpose model. A model that is not, yet, tailored to any specific structure, but rather it is a broad mathematical framework for kinematic chain definition and specification. Of course such a general model entails many missing parameters that need to be defined in order to match the actual kinematic structure of the robot. The task of tuning the model and specifying any missing descriptive parameters is solved through the incorporation of gradient descent optimization techniques, Figure 3.2.

At any given moment the robot relies on the current state of the provided general model in order to formulate estimated body schema approximation based on which the architecture simulates the expected posture of the robot after the last motor actuation. After that, the system optimizes the estimated values of the missing descriptive parameters in order to match the simulated posture through gradient descent methods, then updates the current approximation of the system model and thus propagating the model a step further toward representing the actual structure of the robot.

Hence the main concern in this approach is to determine the the schema of the kinematic chain rather that to formulate a control policy. The architecture we are proposing tries to solve both tasks of controlling the system and building

a representational description of the system. In contrast to our approach, body schema methods are explicit rather than implicit. Hence it does really comply with the concept of embodiment and situatedness, since the end result is constrained by the original, manually programmed, model framework.

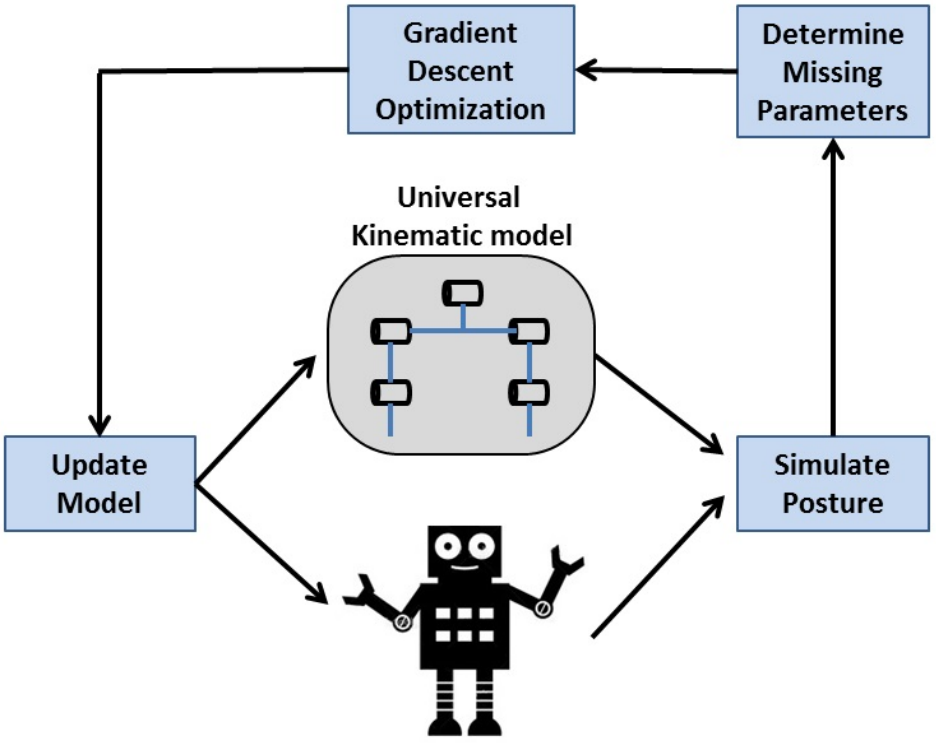

FIGURE 3.2: Body schema approach

Although the idea of body schema has its biological plausibility aspects but not as explicit representation. It is believed that the cerebral cortex has a representation of our body schema but as neural ensemble coding, [4], that is of implicit and adaptive nature. The architecture we are proposing implements implicit representation of the kinematic structure that is inspired by an abstraction of biological findings.

The third approach we are reviewing is motor primitives with reinforcement learning approach. First a set of initial motor primitives is manually designed and pre-programmed into the architecture. Each of these motor primitives can be thought of as a single atomic motor task that is independent, undividable, and performs a simple action like grasping, letting go, arm extending...etc. The architecture incorporate reinforcement learning techniques and exploit the set of motor primitive in order to generate a control policy that can combine these primitive into more complex motor tasks. By using the motor primitives as building blocks the system can generate complex actions through interaction with the environment during which reward feedback assist the control policy formulation, 3.3.

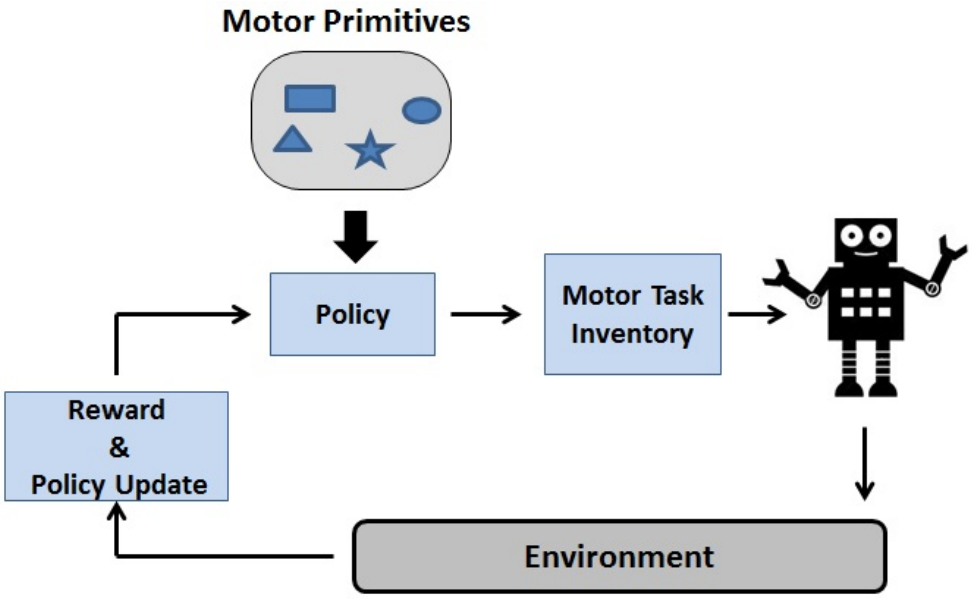

FIGURE 3.3: Motor primitives and reinforcement learning architecture

Relaying on reinforcement learning technique as the main learning mechanism give the learning process a cyclic, episodic nature which requires many iterations

in order to completely generate the required policy. This means that each learning experience should be structured and constrained until the goal task has been mastered. Unlike this approach our proposed architecture does not require any prior structuring of the learning process since, in our approach, the system learns continuously and uninterruptedly as long as the system is in operation. The concept of equipping a learner with prior set of motor primitives raises very crucial questions; How do we define what is primitive task and what is not? How do we decide what tasks should the set of primitives contain? And how many initial primitives is enough to start learning? All of these questions remain opened and undetermined. Unless these questions are answered, it is difficult to adopt this approach as a general task-independent framework for sensory-motor learning for developmental robotics.

3.3 SOINN

SOINN (Self-Organizing Incremental Neural Network), is a Self-organizing map that does not require any presumption to be made about the topology or the distribution of data. It has the ability to filter out non persistent, irrelevant data points that does not reflect the underlying representative pattern of perceived information, while at the same time it tries to capture the descriptive structure of input which enables the network to pertain only what is most relevant to the learned underlying pattern of perception.

The basic algorithm of SOINN is depicted in Figure 3.4. For a detailed explanation of the algorithm see [55, 56].

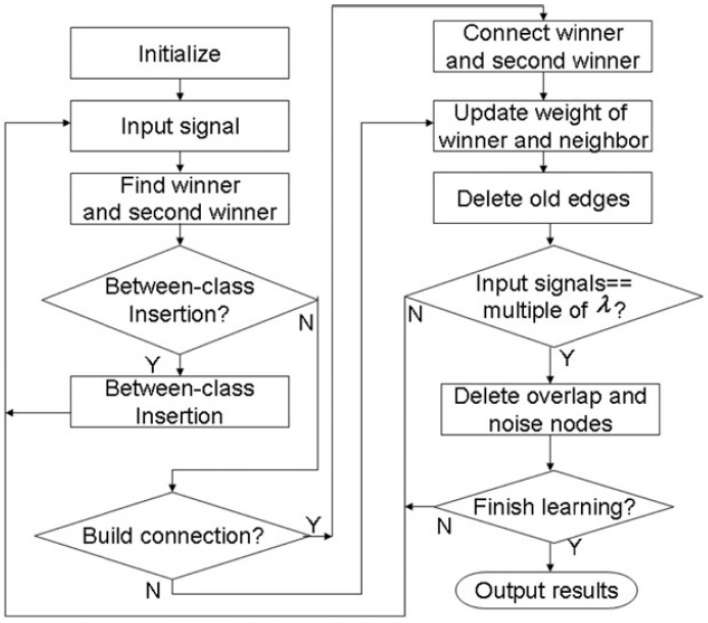

FIGURE 3.4: Flowchart of SOINN [55]

Basically SOINN works by propagating network topology in a way that would self-organize as to resemble the "hot zones" of perception. For example, if a new data point is presented to SOINN then the algorithm would find the closest two network nodes to this newly presented data point, Figure 3.5.a. once these, most closest, nodes are found, SOINN determines whether the newly presented data point is within the coverage zone of these nodes. If yes then these nodes would be now connected by an edge to make up a single cluster of nodes and then they would be altered as to reflect the current blobs of persistent activity, Figure 3.5.b. In the other case where the newly presented data point is out of the coverage zone of the closest nodes, Figure 3.5.c, then this data point itself would be stored by

SOINN as a node that represents a possible independent zone of activity, Figure 3.5.d.

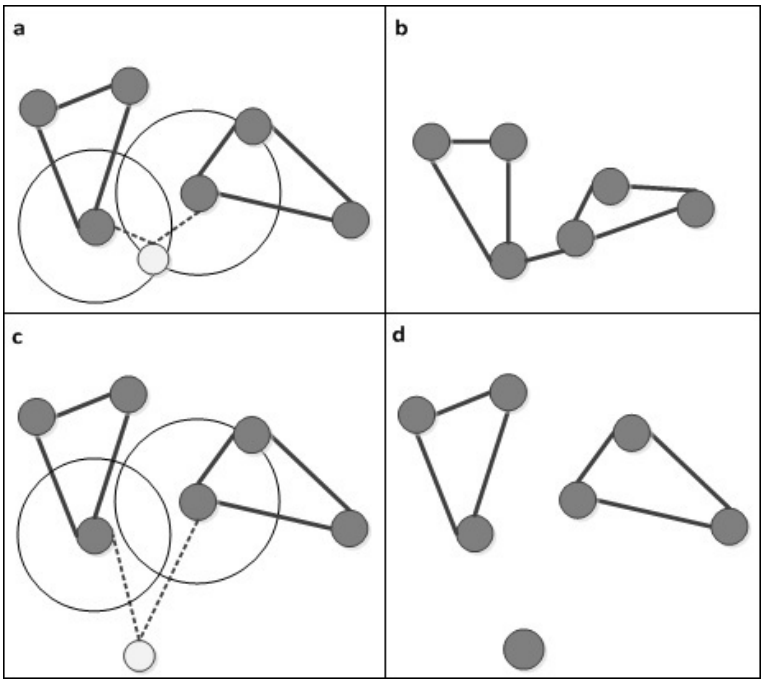

FIGURE 3.5: SOINN dynamics

SOINN has the feature of eliminating noisy and non-stable representations by checking the level of activity of each stored cluster of nodes and then discarding those stored clusters that doesn't represent regions of input space with high activity. So if a cluster of nodes has not been referenced frequently as being a coverage zone for input data points, then this cluster would eventually fade away and removed from the network.

3.4 The Theory of Motor Babbling

Motor Babbling is described as the exploratory learning process of generating sensory-motor associations through continues random motions with ballistic trajectories.

These motions serve the purpose of sampling representative data points that bootstrap the learning system into incremental generation of internal model and implicit control policy for the system at hand.

In other words, we can say that Motor Babbling is the autonomous process of forming an internal implicit model of a moving mechanical system through Hebbian-like Action-Perceptions episodes.

Evidence from Developmental Psychology literature[21, 57] suggests the presence of such exploratory learning processes in the behavior of infants during the first months of motor-ability development.

During the repetitive random motion of the arm, that is considered as a characteristic pattern of infant motor behavior, babies are believed to keep their hand constantly in visual field, which is supposed to serve the goal of building internal associations between actions and consequences in one's own body [58], see Figure 3.6.

This very process of babbling would make the core exploratory engine of the architecture proposed for robot motor learning Figure **??**.

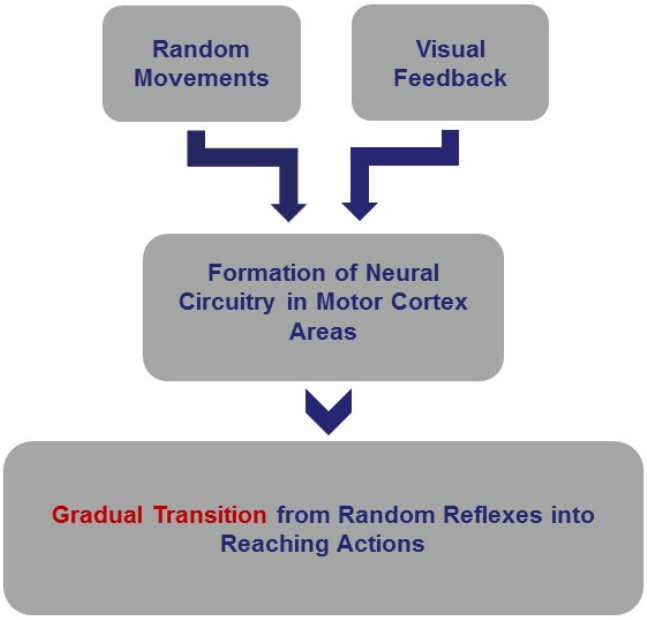

FIGURE 3.6: Correlation between visual information and proprioceptive information in motor babbling during the formation of cortical circuitry .

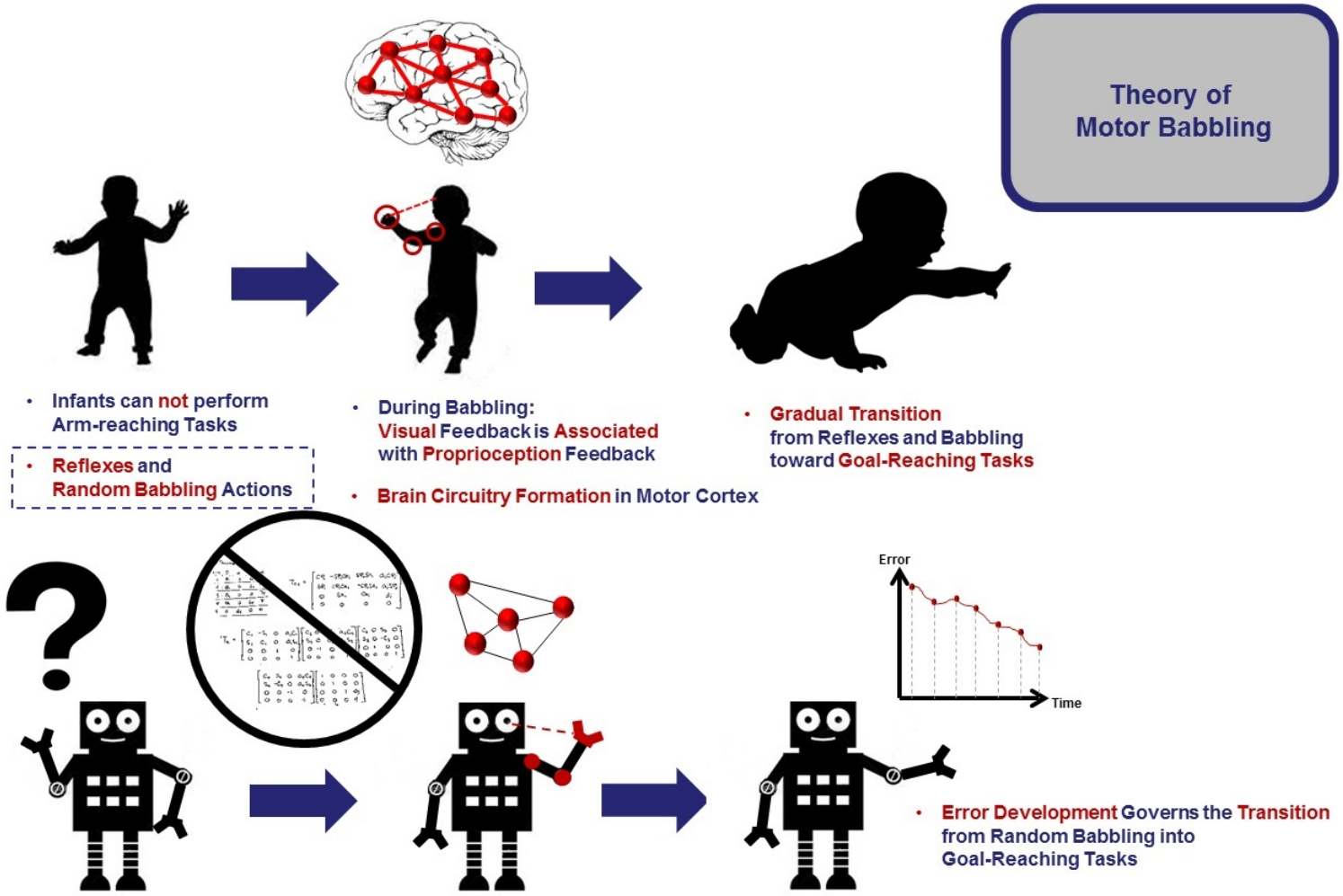

FIGURE 3.7: Motor Babbling process.

3.5 Methodologies

The mechanism we are proposing is based on the idea of autonomous, incremental generation of implicit system model and control policy using layers of self-organizing maps and joint-egocentric representation of reaching experiences.

The robot is not provided with any control models or methods for calculating inverse and forward kinematics. Besides, the learning process does not involve separated training and testing phases, but rather the system starts off by exploratory motor babbling behavior, then a gradual shifting toward goal-directed reaching trajectories is noticed to take place in a smooth manner that reflect the current level of maturity of the learned model.

3.5.1 Learning Through Babbling Architecture

First of all it is important to mention that each sensory-motor experience is represented and learned as a pairing between joint angle and the resultant gripper location in space. And this pairing is joint-related, or joint-egocentric, i.e. for a given joint this sensory-motor learning experience would be $[\theta_i, L_i]$ where θ_i is the angle of joint i, and L_i is the resultant location of the gripper represented in relevance to the joint i ,hence, in the peripersonal space of Joint i. The purpose of this joint-egocentric representation is to make sure that learning is achieved on the joint level, where each joint would learn, the required associations, in manner that is independent from the other joints.

Each joint has its own associative learner, implemented as self-organizing map (SOINN). This joint-specific associative learner is responsible of learning and storing only the most representative data point that resembles sensory-motor pairs of the form $[\theta_i, L_i]$ which are related to the joint to which the self-organizing map belong to, as mentioned above.

When a new target is presented to the system, the location of this target is represented in relevance to the first joint. After the target has been represented in the peripersonal space of the first joint, the system would ask the self-organizing map, of the first joint, for the best angle that would achieve as close gripper location as possible to the given target . Depending on the joint's previous experience that was generated through the motor-babbling-like exploratory actions, the self-organizing map would respond by retrieving the joint angle that is associated with the closest gripper location to the target. Of course the accuracy of the suggested angle would depend on the state of sophistication of the learned contingencies between joint actuations and the resultant consequences in the environment, so at early stages of learning, the suggested angle won't offer high-accuracy solutions. But gradually, with more experience and as the system explore the contingency space, more accurate associations would be learned by the self-organizing map.

The joint angle, which was offered by the self-organizing map (SOINN) of the first joint, would be used to actuate the first joint of the manipulator even before passing the control to the next joint. This means that after the system has found out the suitable joint angle for the first joint, the target perception would be

altered for the rest of joints on the manipulator, so in order for the next joint "$joint_{i+1}$" to ask its associative learner for suitable joint angle, θ_{i+1}, the robot must check the new altered location of target,L_{i+1}, in relevance to the next joint i.e. in the Peripersonal space of the next joint.

This strategy reveals the iterative nature of the solution proposed here, where the problem of finding the best set of joint angles for multi-joint manipulator is solved by breaking down the whole reaching task into smaller sub problems, each handled by an independent standalone subsystem that consist of single joint with its own perceptual peripersonal space and its own associative learner, which is implemented as a self-organizing map (SOINN).

When a new target is presented to the system, starting from the first joint and up until the last one, each subsystem would represent this target in its own peripersonal space then ask its learner for the best angle and actuate the joint according to this angle, after which it passes the control to the next joint and the next subsystem. This process continue until the control reaches the final joint and all the joint would be actuated resulting in a gripper being moved toward the target, Figure. 3.8.

After all joints have taken action, the system must be trained with the resultant, actual, consequences in the environment so the associative learner of each joint would be more accurate in learning the action-consequence model of its joint, Figure 3.9.

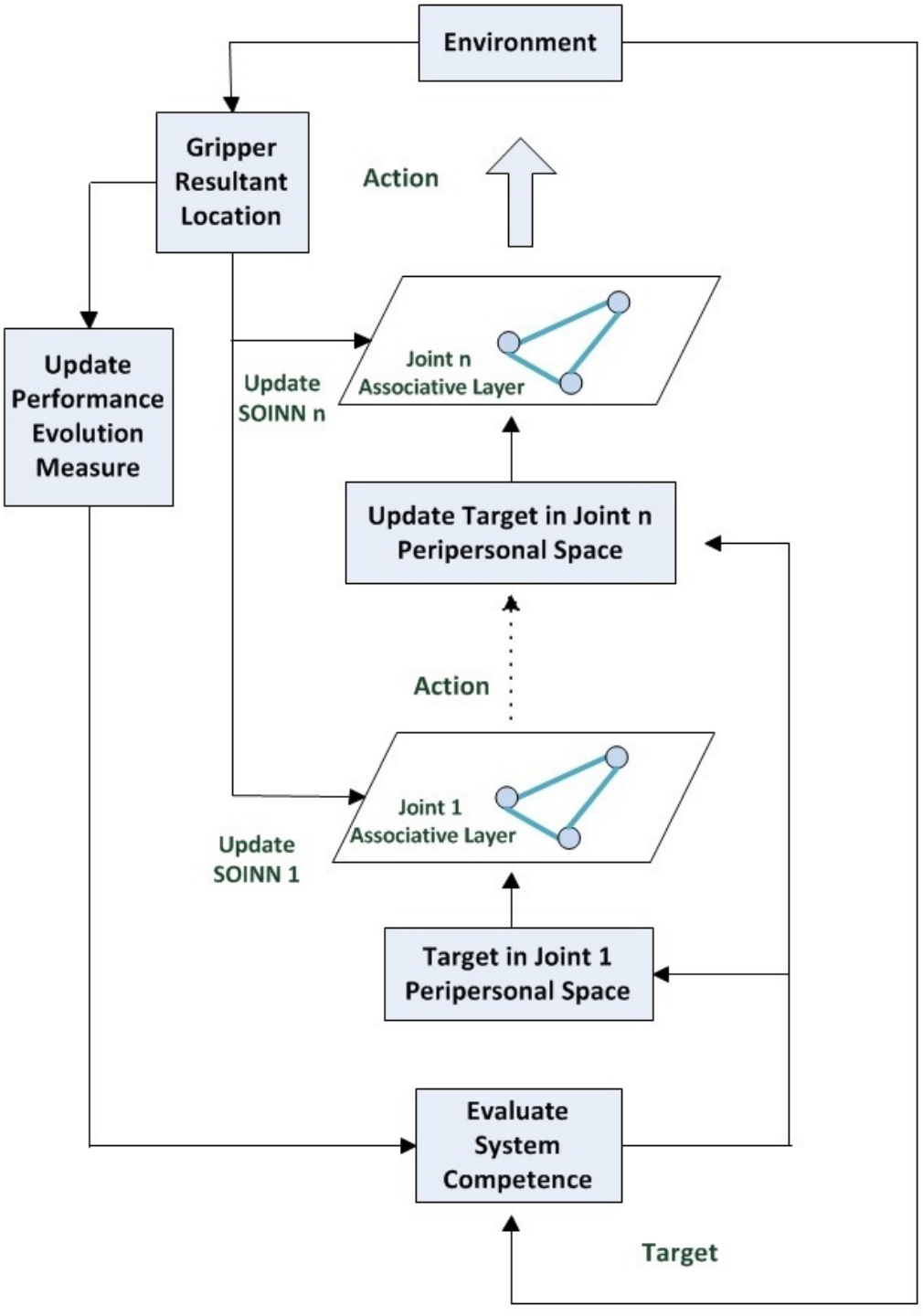

FIGURE 3.8: Learning through babbling architecture.

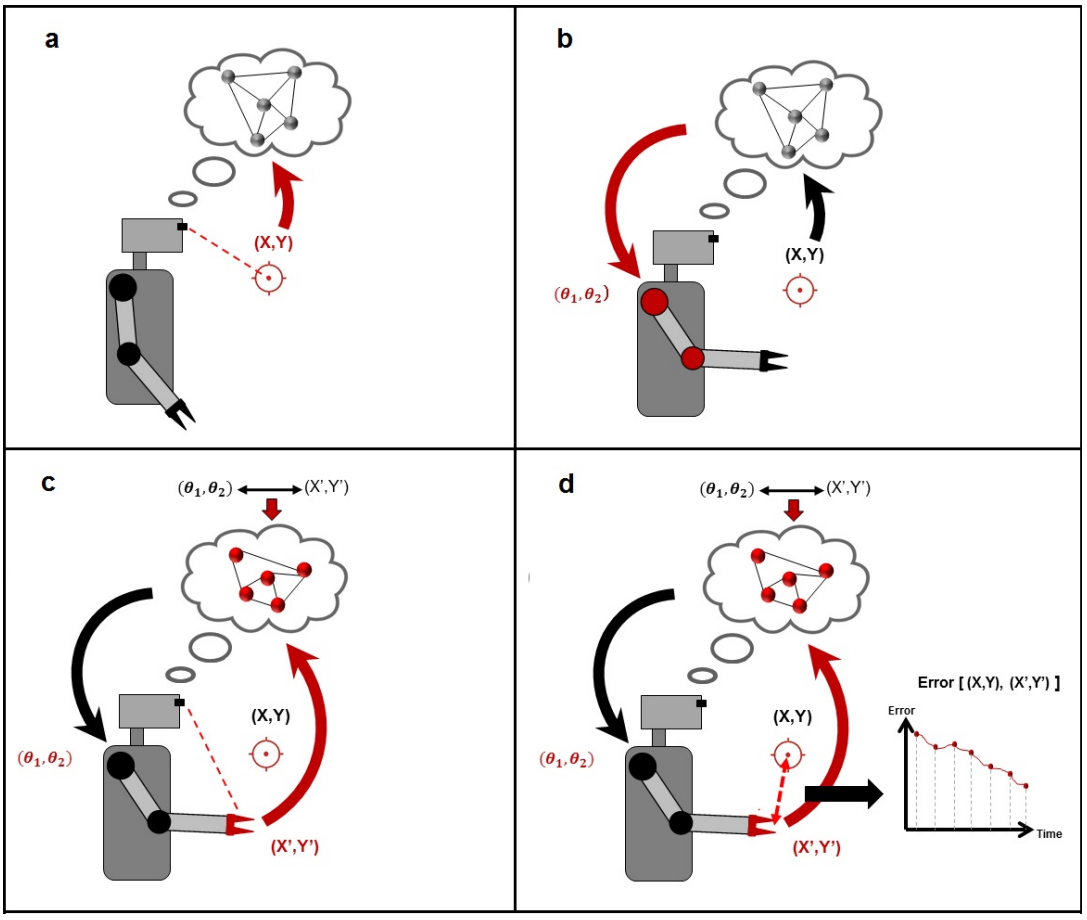

FIGURE 3.9: Robot motor learning by babbling process.

This training would be performed on the joint-level. So if the final location of the gripper, represented in the peripersonal space of joint i, after all joint have been actuated, is L_i, and the joint angle that was actually used by joint i in order for the gripper to reach this location, is θ_i then the joint i would be trained by the associative pair $[\theta_i, L_i]$, Figure 3.9.

This feedback training is essential to guarantee an incremental bootstrapping of the implicit model, generated for each joint, toward higher competence of target reaching operation, where training and performing are taken place side by side without the need for adopting separated-phases-approach for learning.

3.5.2 The vanishing of Babbling Actions

As mentioned before, the mechanism proposed here does not require separated learning and testing phases. In other words, there is no need to manually set a fixed number of episodes for training before the robot can be engaged in a real goal-reaching actions.

In the approach described here, training and learning take place in a real-time manner. The system itself decides when an exploration action is needed and when actual goal-reaching can be performed so a gradual transition from random exploration into reaching actions takes place while the system is being trained continuously in both cases. So initially when we run the robot for the first time, the robot has no idea about how to control its joint in order to reach a target, here the robot actions would be random ballistic trajectories similar to the ones performed by infants at early stages of motor development[57].
During this random motor babbling behavior the robot starts to generate an internal model for the control policy of its joints, through action-consequences coupling, which result in an increased ability to control these joints, hence a less resultant error in reaching a target.

To control the balance between motor babbling and target-reaching behaviors the following equation is used:

$$P(rnd) = 0.5 + \xi(m_{cp} - m_{fp}) \tag{3.1}$$

Where $P(rnd)$ is the probability of performing a random action, and $\xi(x)$ is the normalized value of x. The quantity m_{cp} is the mean error in the close past and m_{fp} is mean error in the far past.

The concept of close past and far past is generated by making the system maintain, at each time step, a list of measured error, described as the distance between the target location and the resultant gripper location, during the last n steps. This list then is divided in two halves; The most recent half, which consist of set of errors between $j = t$ and $j = t - (n/2)$, is considered as a set of errors in the close past. The other half, that consists of set of errors between $j = t - (n/2)$ and $j = t - n$, is considered as a set of errors in the far past.

Dividing the most recent n time steps into close past and far past serves the goal of altering the frequency and the necessity for random actions. So when error is reducing, and the robot performance is getting better, a negative value of $\xi(x)$ would result, which would decrease the random action probability, $P(rnd)$. on the other hand, when the error is increasing, a positive value of $\xi(x)$ would be generated resulting in higher motor-babbling probability, Eq. (3.1). see Figure 3.10

In other words,when the robot is getting better, in controlling its joints, then less random actions are needed, while increased error means that the robot has not yet learned the action-consequences nature of its arm joints and thus more motor babbling behavior is yet need to be practiced by the robot.

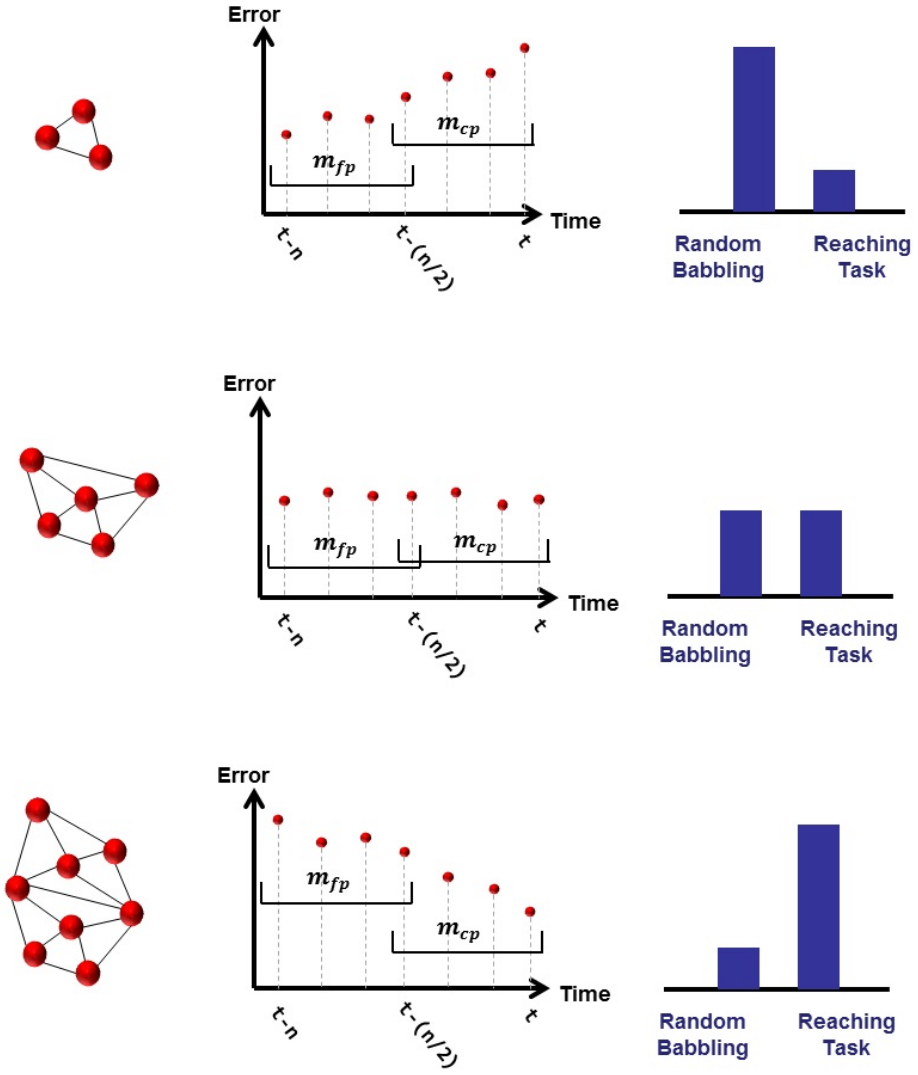

FIGURE 3.10: when error is being reduced then less random actions are needed, while increased error means more motor babbling behavior is yet need to be practiced by the robot.

3.6 The Experiments

3.6.1 Babbling Experiment

In this experimental setup, a simulated 2DoF planar robotic arm, Figure 3.11, is used to demonstrate the developmental sensory-motor learning process, starting

by random motor-babbling actions and then shifting gradually toward performing target-reaching trajectories.

It is crucial to mention that the robot was not provided with any knowledge about how to control its joints, besides no action-consequence model was preprogrammed by the designer beforehand of learning.

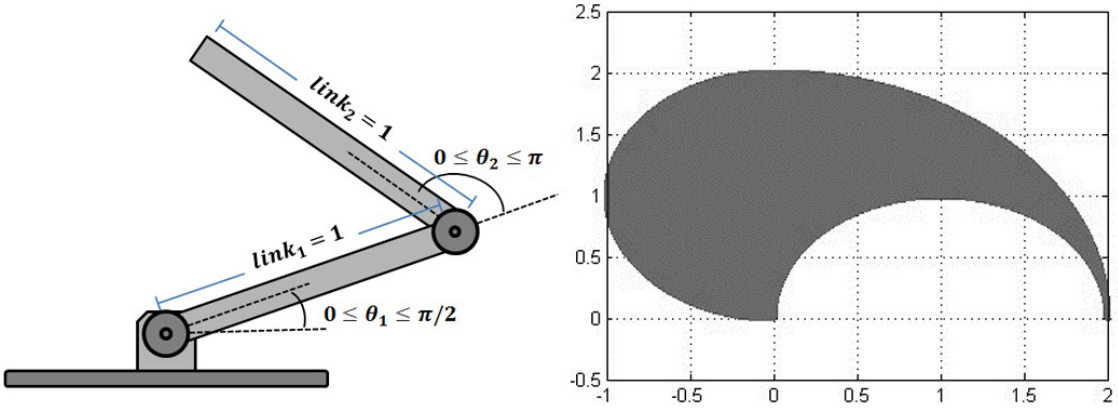

FIGURE 3.11: 2DoF robotic arm and its corresponding workSpace.

A red ball is used as the target that the robot is required to reach at any given time. The ball location is generated randomly and then the robot is asked to reach it with its end-effector, then, after the robot trail to reach the target, a new location is generated whether the robot has managed actually to reach the target or not, Figure 3.12.

As mentioned above, and illustrated in Figure 3.8, the trajectories that are performed by the robot, whether target-directed or random, are always used as a training signal for the learning system, which implies a continues adaptation and learning of the generated implicit model of control.

In Figure 3.13, a gradual decrease in error is noticed with more practicing of the learned model that was initiated by the babbling actions.

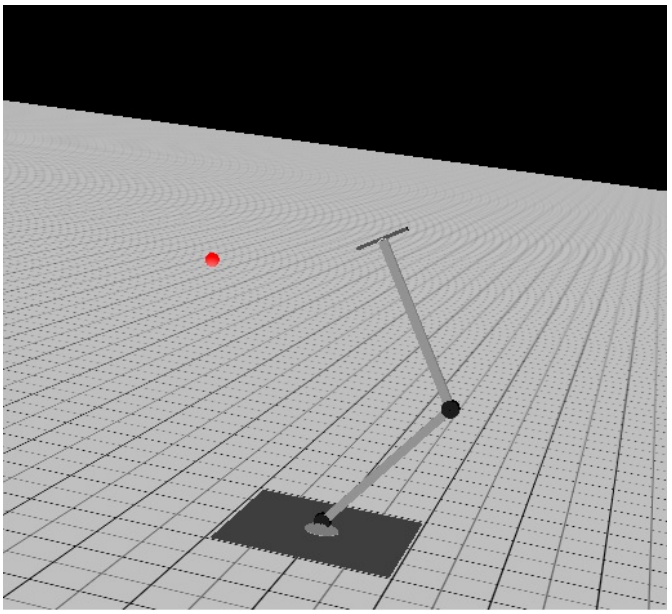

FIGURE 3.12: The ball location is generated randomly and then the robot is asked to reach it with its end-effector, then, after the robot trail to reach the target, a new location is generated whether the robot has managed actually to reach the target or not.

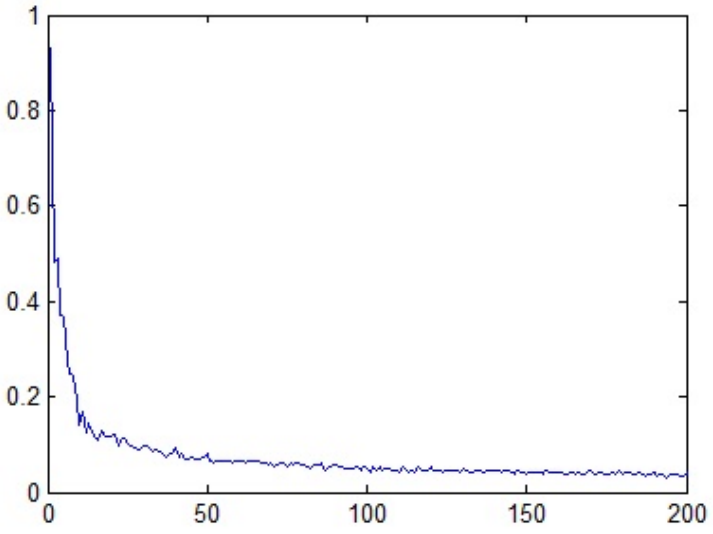

FIGURE 3.13: resultant error during learning.

The robot performance starts with high error rate. But with more training experiences the multilayer architecture of self-organizing map, SOINN, starts gradually to capture the contingencies behind joints angles and resultant end effector location. This incremental self-organizing process results in an observed decrease of the resulted error of generated actions.

3.6.2 Adaptive Babbling

In this second experiment we demonstrate the system's reaction to a sudden unexpected change in the physical structure of the robot. This sudden change could account for a breakage in a joint, alteration of a link or a displacement of the end-effector location in relevance to the arm links.

In this experimental setup we still have the same task of reaching a red ball, but now , after the system has learned its own implicit model, we suddenly increased the length of the arm's second link by 10% of its original one.
Altering the physical structure of the system means that now the learned implicit model does not accurately reflect the actual system nature. So if the system, before this unexpected change, had already reached a level of stability in term of the frequency of babbling actions, where a lower rate of random motion could be noticed, then now this stability won't last, and the robot would need to re-explore the contingencies of its action-consequence relation.

In Figure 3.14, the horizontal axis shows a sequence of groups of time-steps ,each consist of 10 actions, that depicts the transition of the robots performance between motor babbling and target-directed actions. The vertical axis shows the number of babbling motions that was performed in each group of 10 time-steps. The experiment was repeated three times in order to emphasize results repeatability.

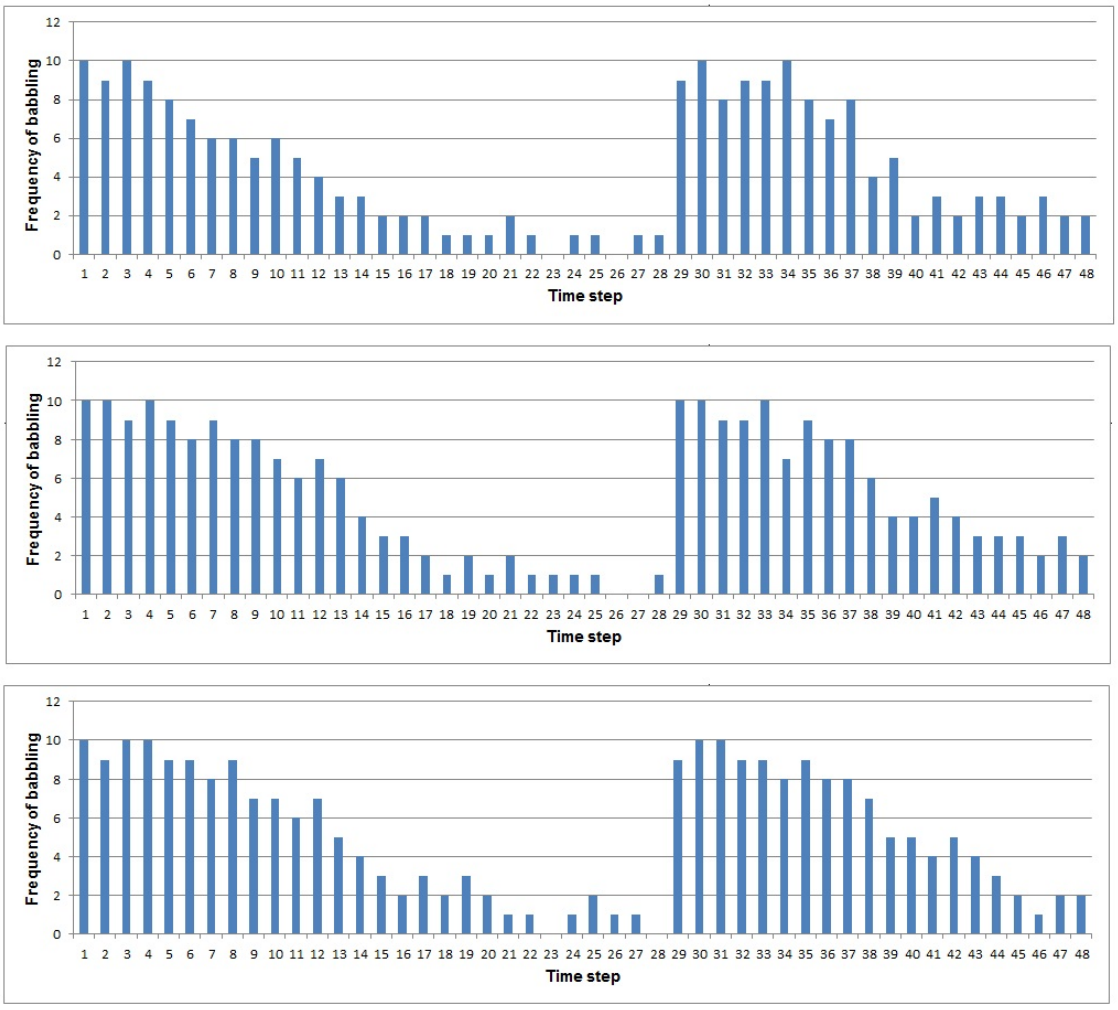

FIGURE 3.14: A real-time reaction to an unexpected change in the length of the second link.

As expected, most of the robot's actions, when it starts learning, are babbling ones and that is because the robot is not aware of its kinematic model. But then gradually this rate of babbling actions would decrease as the system proceed in

building an implicit model of its control. Eventually we notice that almost no babbling actions are being performed but rather almost all of the taken actions are goal-directed.

During the robot's performance we altered the second link length, as mentioned above. This change would increase the resultant error in the robot's reaching accuracy because the learned sensory-motor associations does not accurately reflect the actual current status of the system. This increased error would generate a positive difference between m_{cp} and m_{fp} from Eq. (3.1), what eventually results in a higher $P(rnd)$, which is the probability of performing a babbling action.

This change in the behavior of the system can be observed in Figure 3.14 where a peak in the frequency of babbling actions is clearly noticed around the point in time where the physical structure of the robot was altered.
What can be noticed also is that the domination of babbling actions won't last forever, but rather it would be there as long as the system hasn't fully recaptured the contingent action-consequence relation of its recently altered physical structure.

Next, we went on altering different parts of the robot's structure, for example in Figure 3.15 we did the same thing but this time we increased the length of the first link.

Figure 3.16 shows the reaction of the system when we applied some link offset to the first link, then the same thing was done but to the second link,Figure 3.17.

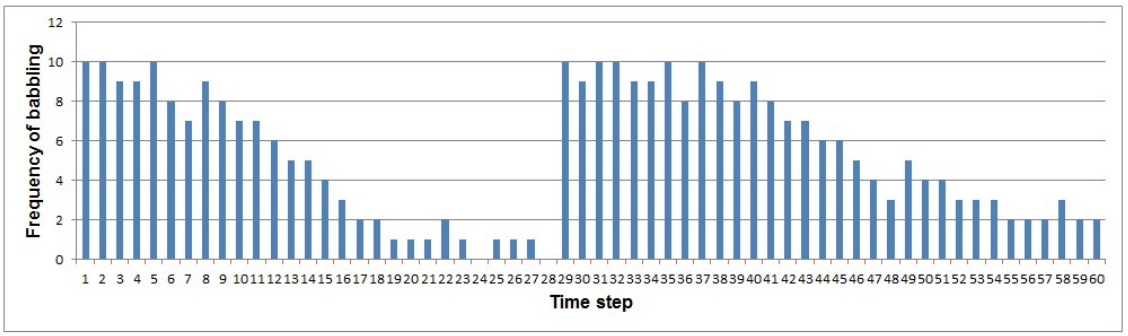

FIGURE 3.15: A real-time reaction to an unexpected change in the length of the first link.

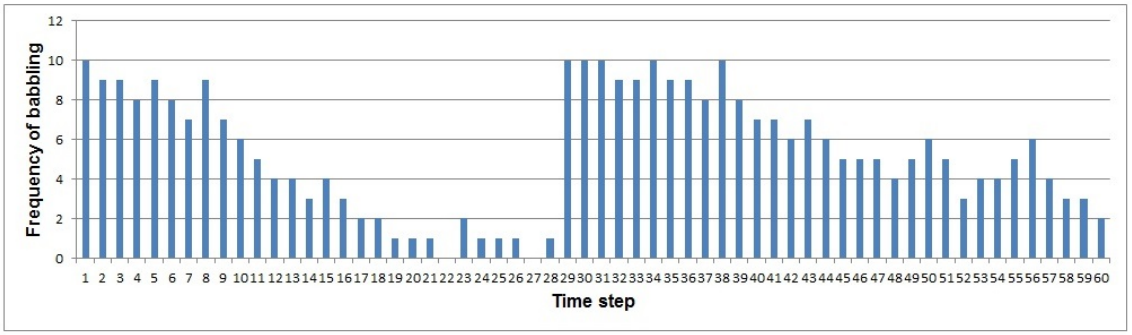

FIGURE 3.16: A real-time reaction to an unexpected change in the physical structure introduced as an offset in the first link.

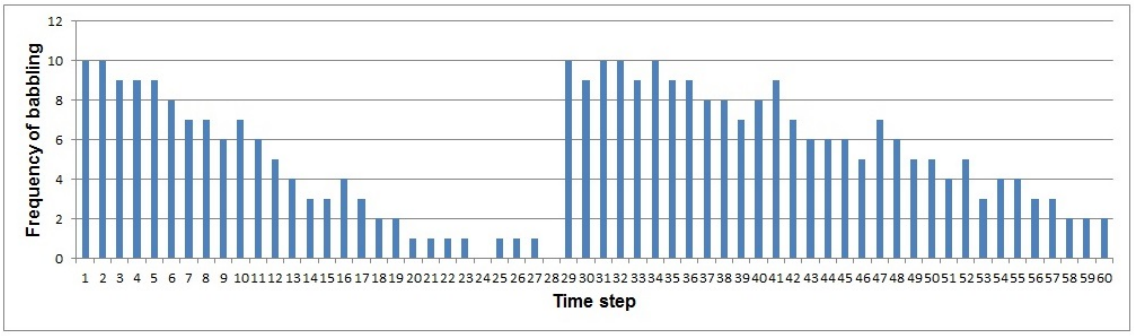

FIGURE 3.17: A real-time reaction to an unexpected change in the physical structure introduced as an offset in the second link.

In all of the structure-alterations scenarios the system showed noticeable sensitivity to sudden unexpected structural changes and always has responded in adaptive manner in order to compensate the discrepancy between the autonomously formulated implicit control model and the actual physical shape.

This observation emphasizes the impact of the concept of learning through babbling on the ability of the system to adapt and react to unanticipated changes and conditions.

3.6.3 Discussion

After learning, SOINN's nodes converges into a network of most representative data samples. These representative nodes would be responsible of approximating the sensory-motor associative patterns that makes up the implicit model of joint control.

In order to have an insight into the learned model of sensory-motor associative space, a visualization of the first layer of SOINN is depicted in, Figure 3.18. This network represent the learned implicit control model of the first joint of the robotic arm. Figure 3.18.a shows a 3-dimensional visualization of this resultant network, where each node represents a single representative associative sensory-motor pairing of the form $[\theta_1, L_i]$, as described in section3.5.1. if we look at the topological structure of this network from 2-dimensional perspective, Figure 3.18.b, we notice

that it captures a very similar structure to the Cartesian space but spawned across a third dimension of the associated angles of $joint_1$.

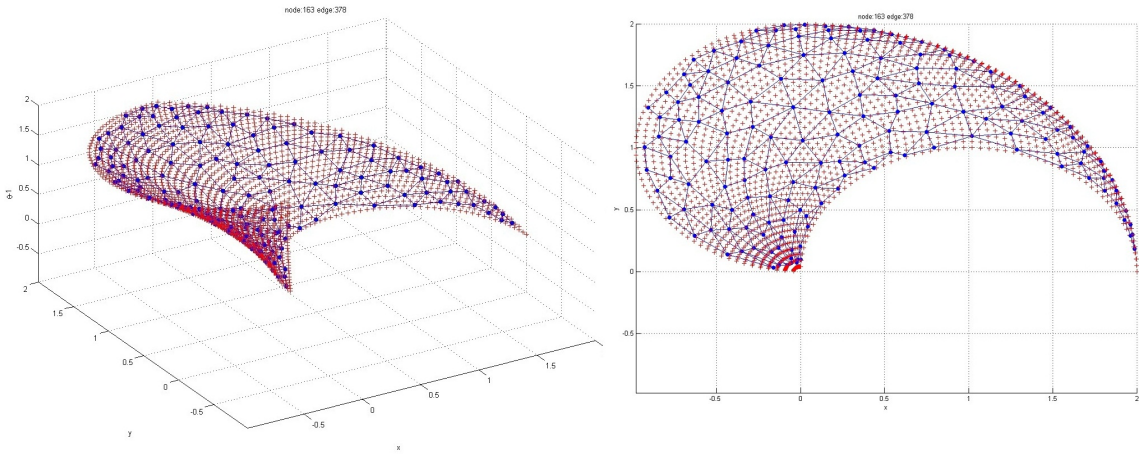

FIGURE 3.18: A 3-dimensional representation of the approximated sensory-motor associations that correspond to $joint_1$ (left). the same network but in 2-dimensional perspective (right).

Next is a visualization of the learned SOINN network but for $joint_2$, Figure 3.19. Again the network to the right, Figure 3.19.b, is the 2-dimensional perspective of the 3-dimensional SOINN network, Figure 3.19.a, that represents the sensory-motor associative model for $joint_2$.

Notice, from Figure 3.19.b, the egocentric characteristic of the learned model since the Cartesian part of the associative data points does not reflect the whole work space but rather it captures only locations that are taken from the perspective of $joint_2$.

In both learned networks, Figure 3.18 and Figure 3.19, we notice that SOINN has the ability to cover the whole input training space with consistent distribution of nodes that enables the system to generalize even for unseen data points that was not provided during the process of network generation.

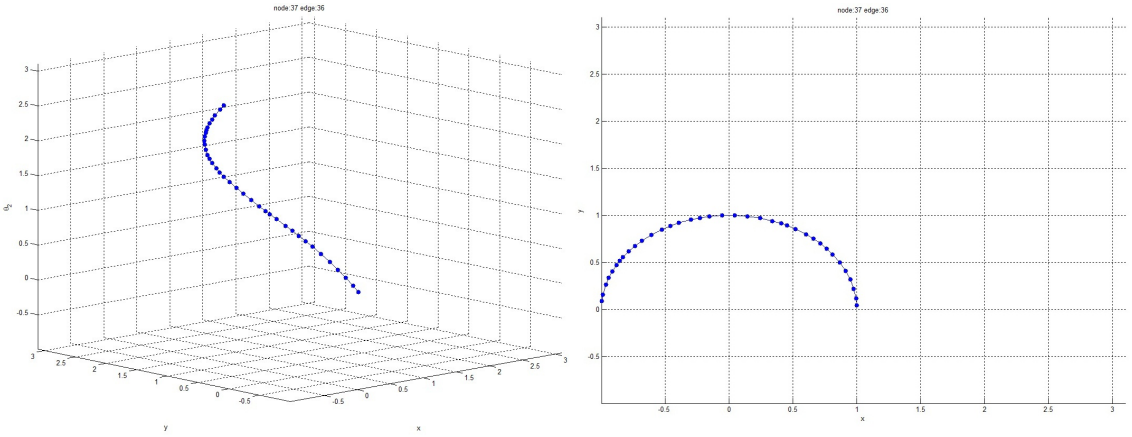

FIGURE 3.19: A 3-dimensional representation of the approximated sensory-motor associations that correspond to $joint_2$ (left). the same network but in 2-dimensional perspective (right).

This was demonstrated in Figure 3.13 where the ball location was generated in continues input space rather than discrete one, but yet the system managed to generate trajectories of decreasing error even without the need for separated training and testing phases.

3.7 Summary

In this chapter we have presented an architecture for learning sensory-motor associations for target-reaching tasks, using self organizing neural networks. The approach that was taking is inspired by developmental psychology where motor learning starts by babbling-like ballistic trajectories, similar to the ones observed during early stages of motor development in human infants, then the robot shifts toward reaching-actions with continuously decreasing error.

This developmental approach toward robot learning was demonstrated by the fact that no preprogrammed control policy was provided beforehand of learning. But rather the robot explored, on its own, the action-consequences contingencies of its joints and then, autonomously, generated an implicit control model through Motor babbling actions.

Chapter 4

Receptive Fields.. Self Organization.. Hebbian Plasticity and Robot Learning

4.1 Introduction and Motivation

Providing a mathematical model that describes the behavior of a dynamic adaptive system is not always an easy or even a possible task. In environments that are characterized by being either continuously-changing or not fully determined, it is considered unpractical to design system functional policy based on an explicit formal description of the expected conditions of operation. besides, sometimes the system itself could be of non-rigid descriptive characteristics that result in model

parameters adaptive in nature or difficult to predict[14].

Consider for example the challenge of designing a locomotion system for planetary exploration robot that would navigate unfamiliar terrains with undetermined physical characteristics like viscosity or surface friction. On the other hand, let's take the case of designing a robotic manipulator with the ability to compensate for any unmodeled nonlinearities caused by actuator wear-out-scenario, in order to achieve a self-calibrating system. Any predetermined control policy for such systems won't operate effectively unless an anticipated mathematical model of the operation environment, or system dynamics, would be embedded into the system model ahead of time, which is not always a computationally plausible task.

Here we notice the necessity for an approach to system model based on reactive learning rather than rigid hard wiring [16]. Such an approach tends to exploit system-environment interactive experiences, through continues sampling of input-output correlative incidences, in order to gradually scaffold through real-time formulation of implicit functional structures that govern the system's behavioral patterns. Adopting such learning approach would reduce design rigidity that is imposed by manual model-hard-wiring, thus resulting in system with increased flexibility concerning task specificity or the applicable environment of operation [59].

Designing a learning mechanism capable of capturing implicit modeling associations means that the system must make use of its own past experience in order to shape any emergent governing relations between input and output samples.

In other words, the learner need to be in constant state of system-environment interaction through embodied alteration of the actual surrounding environment and then keeping track of these interactions through situated perception of the outcomes of these experiences[60].

During this cyclic relation between the learner and the world around, implicit cognitive structures start to form and take shape not as explicit modeling policies but rather as emergent regularities governing the intrinsic contingencies of both environment characteristics and system's behavioral dynamics[61]. These acquired input-output relational schemes are not rigid or final, but rather they are subject to continuous reshaping and alteration by further experiences and incidences of system-environment interactions, and this what would provide the ability for the learner to evolve and scaffold without manual embedment of any priori descriptive system parameters.

The fact that past experience serve as training data points, requires taking into consideration that in many applications, the availability of collected data from the environment would be of a sparse, distributed nature where sample data points are scattered discontinuously over the data space [62].

With only a limited number of acquired data points we need a mechanism that would allow us to generalize behind what has been captured by means of interpolating those sparsely-collected samples in order to have a system that can respond continuously to the external world. In order to achieve such generalization ability,

it worth noting that within data space, learned data samples can be of a topological significance on the basis of their physical location in data space[63]. This positional significance makes up the essential aspect of the process of approximating unseen data points by aggregation of locally active learned samples, where the contribution of each learned sample would be based on its location in the space. Such approach is inspired, in a very limited extend, by biological mechanisms of neuronal receptive fields in the mammalian nervous system[64].

In this chapter we propose an incremental learning neural network that can bootstrap from a state of complete ignorance of any representative sample associations. Our approach captures the problem's data space components using emergent arrangement of receptive field neurons that self-organize incrementally in response to sparse experiences of system-environment interactions. These learned components are correlated using a process of hebbian plasticity that relate major components of input space to those of the output space.

4.2 Methodologies

The network we are proposing is an Incremental learning mechanism that can bootstrap from a state of complete ignorance of any representative sample associations.

In order for the proposed network to learn the problem in hand, three sub-problems need to be learned: first of all the input-space of the underlying function need to

be captured. Second, the corresponding output-space as well is modeled as a set of representative elements. Finally,associative relations between major components of these input and output spaces are captured and learned in order to make up the backbone of the governing associations implicit to the learned function.

The main approach is that each of input and output space is modeled by a set of self-organizing nodes that represent emergent, major and most representative components (called space-component nodes) of the corresponding space. After the space-component nodes of each space have been learned, a hebbian plasticity process is employed in order to correlate the space-component nodes of input space with those of the output space.

Any data point, within each space, is represented as a combination of locally activated space component nodes. The activation for each of these node is influenced by the excitation level of of the node's receptive field in response to the topological location of the stimulus data point within the data space. So as a result, the stimulus data point is represented as a vector of activation levels of various space-component nodes that make up the data space within which this data point is located.

As noticed above the proposed mechanism relays on some biologically-inspired concepts as receptive fields and hebbian plasticity. That's why these concepts are reviewed briefly in the next section.

4.2.1 Receptive Fields

Let's imagine standing in the middle of a pitch-black dark room, and someone, couple of meters away, is holding a small candle. The fact that the candle is in your visual field, and you can see it, means that some of the neurons in your retina, the retinal ganglion cells, are activated by the presence of the candle stimulus within their receptive field portion of the visual field.

Now let's imagine that the person holding the candle is moving across the room but still within your visual field while you are still holding the same location and gaze direction, then as the location of the candle within your visual field changes, the set of activated neurons in the retina would change as a response to the presence of the candle stimulus in their corresponding receptive field portion of the visual field.

This gives us the impression that our visual field is divided into overlapping regions so that a given retinal neuron would be sensitive to the applied stimulus only if located within given region, or set of adjacent regions, of the visual field. In other words, a neuron that respond to the presence of a stimulus in its associated region of the visual field would show little or no response if that stimulus was located in different region of the visual field. That is way we call this particular region, to which a given neuron shows sensitivity to applied stimulus, as the receptive field of the sensory neuron[65].

The concept of receptive field is not only associated with retinal ganglion cells but rather it is a broad abstract perceptual and computational concept that is present

across different sensory modalities and even within higher hierarchies of cortical information-processing neural assemblies[4, 66].

So as a general definition we can say that The receptive field of sensory neuron is the portion of the "perceptual" space within which the neuron would respond to applied stimuli. In other words, it is the portion of the space that would elicit response from the corresponding neuron if a stimulus was located within that particular portion of space. Hence, it is a region of the sensory surface to which a particular neuron show sensitivity to the applied stimulus[66].

It is important to mention that the relation between portions of the sensory space and the corresponding neurons is not usually a one-to-one relation. Meaning that the presence of a stimulus in a given region will not usually activate only one neuron but rather a set of neurons would respond to that stimulus[67]. The reason behind such observation is that usually the receptive fields in the sensory space do not form mutually exclusive divisions but rather they have overlapping coverage of the sensory space. This means that if a stimulus was presented in a given point of the sensory space, then we would have a given neuron with maximal respond to this stimulus together with a set of other neurons with a less level of response.

The maximal-response neuron is the one who's corresponding receptive field center point is the closest to the point where the stimulus was applied, and the set of less-response neurons are the ones who's corresponding receptive fields are overlapped with that of maximal-response neuron[68].

Although different sensory neurons have different receptive field models, but it is

convenient to approximate each receptive field model as a Gaussian function of stimulus location, as shown in Eq. (4.1), where the μ value of each Gaussian is at the center of the corresponding receptive field ,which represents the preferred location to which the receptive field would demonstrate maximal response. σ determines the spread of each receptive field.

$$f(x) = \exp\left(\frac{-\|x-\mu\|^2}{2\sigma^2}\right) \tag{4.1}$$

In other words, each neuron would respond maximally when the stimulus is at the center point of the corresponding receptive field, then the response would fade and decrease gradually as the stimulus location moves away from the center point until no response at all is noticed when the stimulus moves completely outside the coverage area of the receptive field[68].

4.2.2 Hebbian plasticity

Hebbian plasticity, as explained by Donald Hebb in his book "The Organization of Behavior" [69],is a theory that attempts to explain the organization of neuronal wiring and its adaptive dynamics in response to external experiences and stimulation.

The main idea is based on the concept that a correlated activity of two neurons would somehow increase the wiring efficiency among these neurons. In other words, a synchronous firing of pre-synaptic and post-synaptic neurons would strengthen

the synaptic connection between them so that a firing of pre-synaptic neuron alone would provoke correlated firing in the post-synaptic neuron.

Hebb's original Postulate stated that:

> *"When an Axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased."*[69]

This statement is usually summarized loosely as "Neurons that fire together wire together", giving an approximate idea of the causality behind synaptic plasticity in the nervous system. Actually this causal relation does not govern only the generation of new connections among neurons but also it accounts for the elimination of existent ones as well[70, 71].

In other words, a repeated and persistent joint-activity of two neurons results in a long-lasting effect of increased efficiency in the synaptic transmission between these neurons, this phenomena is known as Long-term potentiation,or LTP. While on the other hand, a repeated out-of-synchronization firing results in long-lasting effect of decreased synaptic transmission efficiency known as Long-term depression,or LTD [70–72].

Both long-term potentiation and long-term depression processes constitutes the basic mechanisms of synaptic plasticity in the nervous system that account for adaptation to behavioral, environmental and physiological changes[73].

In its simplest form hebbian learning rule could be described as:

$$\Delta w_{ij} = \eta r_i r_j \tag{4.2}$$

where: η is the learning rate. r_i is post-synaptic firing rate and r_j is pre-synaptic firing rate. Such simple formulation emphasizes the dependence of synaptic efficiency change on the activity of both pre-synaptic and post-synaptic neurons together.

A more complex and general mathematical formulation of the learning rule could be described as:

$$\Delta w_{ij} = f_1[(r_i - f_2)(r_j - f_3) - f_4] \tag{4.3}$$

Now if we consider: $f_1 = \eta$ (the learning rate). $f_2 =< r_i >$ (average firing rate of neuron i). $f_3 =< r_j >$ (average firing rate of neuron j) and f_4 as a weight decay term, that we choose to neglect here. Then the rule would look like:

$$\Delta w_{ij} = \eta[(r_i - < r_i >)(r_j - < r_j >)] \tag{4.4}$$

it is clear from Eq. (4.4) that the average firing rate of each neuron is what determines the direction of plasticity whether it would be toward synaptic potentiation or depression adaptation.

Hence, a long-term potentiation would take place when both pre-synaptic and post-synaptic neurons both have either over-average or under-average firing rate.

While a long-term depression would take place when the activity of pre-synaptic and post-synaptic neurons has different directions[74].

4.2.3 Self Organization

As mentioned previously our approach captures each data space components using emergent arrangement of receptive field neurons that self-organize incrementally in response to sparse experiences of system-environment interactions. The mechanism through which these space component nodes are captured is based on Self-Organizing Incremental Neural Network (SOINN)[55]. See section 3.3.

4.2.3.1 RF-SOINN: a modified SOINN Neuron

In order to use SOINN neuron as space-component nodes, we had to create a modified SOINN nerons. We call this Neuron RF-SOINN (Receptive Field SOINN). Actually RF-SOINN is combination of regular SOINN network and the concept of receptive field. Thus resulting in SOINN neurons that have receptive field-based activation function.

Originally SOINN neuron has step activation function. This is demonstrated by the fact that no matter where the input point is located within a given SOINN neuron's coverage area, then the neuron would respond by maximum activation.

Thus all activated SOINN neurons would exhibit uniform activation level if the stimulus is within their perception spot. This fact would limit the possibility of

using SOINN network as data-coding mechanism.

In order to overcome this limitation, we modified the activation function of SOINN neuron, by giving it a Gaussian activation function Figure 4.1, thus allowing some non-linearity in the way a given data-point is coded. Now two active neurons won't show the same activation level in response to given stimulus, but rather each neuron's activation level is governed by the topological location of the stimulus, within the data space, in relevance to that neuron.

The significance of this modification would appear in the following sections.

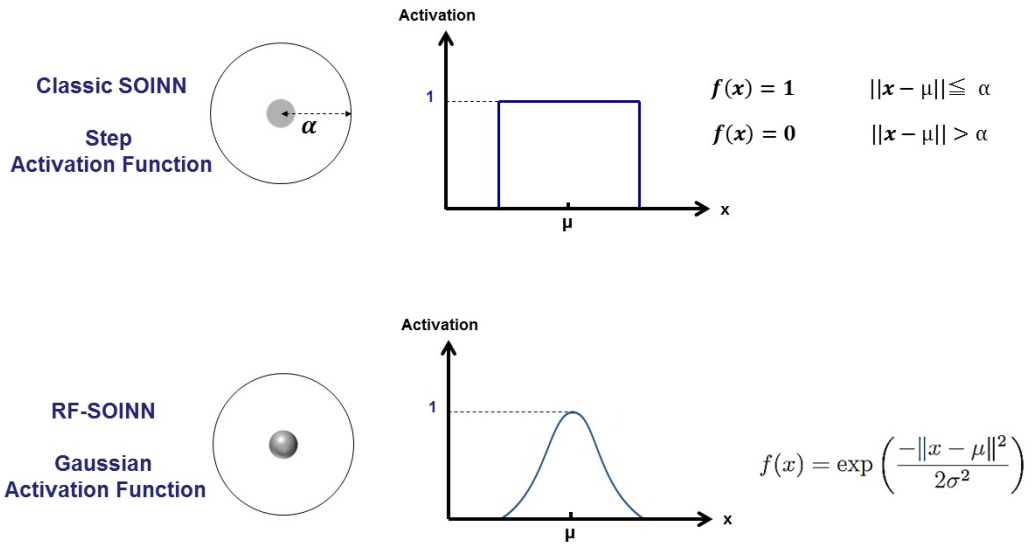

FIGURE 4.1: SOINN Neuron has step activation function, while RF-SOINN nur has Gaussian activation function.

4.2.4 The Network Architecture

4.2.4.1 Overview

In the formulation that follow, the term, Learning Experience E_k, refers to a single learning instance that is represented as input-output pairing that correlate an arbitrary data-point from the input-space with an arbitrary data-point from the output-space. So the kth learning experience is written as $E_k : [X_k, Y_k]$, where X_k is g-dimensional vector that represent an arbitrary input-data-point of g-dimensional input-space, and Y_k is h-dimensional vector that represent an arbitrary output-data-point of h-dimensional output-space.

The main idea, on which our proposed architecture is based, is that we model a given data space by a set of representative data points each of which can be thought of as a Space-Component Node that reflects a zone of persistent stimulation in the data space. This means that the input data-space S_{input} is written as a set $S_{input} : \{C_{in_1}, C_{in_2}, ...C_{in_m}\}$ of m input-space-component Nodes C_{in_j} each of which is a g-dimensional node that constitute a coding component within the g-dimensional input-space. Hence, any arbitrary input data-point X_k in the input-space is coded as a non-linear combination of the activation levels of various input-space-component nodes. This non-linear combination does not extend over the whole set of space-component nodes but rather it includes only those space-component nodes that were locally activated by the topological location of the

input data-point, within the space, in relevance to space-component nodes, Figure 4.2.

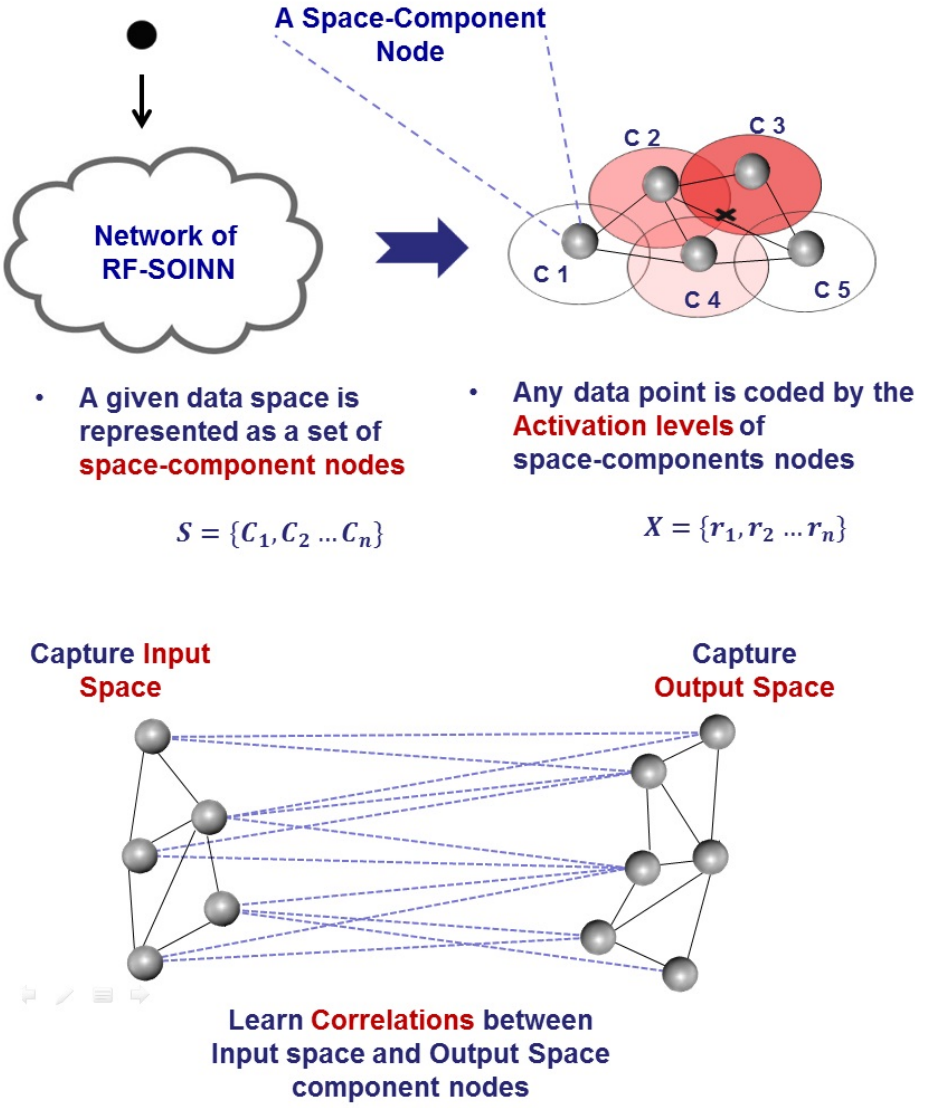

FIGURE 4.2: Each data space is modeled by a set of space-component nodes each of which reflects a zone of persistent stimulation in the data space. After capturing and learning the set of space-component nodes that make up each data space, space component from input-space are correlated with synchronously activated ones from the output-space through hebbian connections.

Equivalently , the output-space S_{output} is written as a set $S_{output} : \{C_{out_1}, C_{out_2}, ...C_{out_n}\}$

of n output-space-component nodes C_{out_i} and, again, any arbitrary output data-point Y_k is coded as a non-linear combination of locally-activated output-space-component nodes.

Finally, we need to learn the correlative relation between each space-component node of the input-space with those space-component nodes of the output-space. Since any data point is represented as a combination of space-component nodes of the corresponding space, then instead of learning the correct output for each data point in input-space, the network needs only to learn the activation pattern of output-space-component nodes for each active input-space-component node. In other words, we need to determine the set and pattern of space-component nodes of the output-space that would response if a given single space-component node from the input-space is excited. This information is exactly what we try to learn and capture through hebbian plasticity process. After that, the correct output for any arbitrary data point within the input-space is determined as an integration of output-space-component nodes that were activated by each input-space-component node encoding that arbitrary data point,Figure 4.3.

4.2.4.2 Data learning

As mentioned above, each data-space is represented by a set of space-component nodes. This set is acquired as SOINN network of self-organizing space-components nodes so that the topological structure of the learned map captures the distribution of regions with persistent occurrence of data-points within each data-space. Each

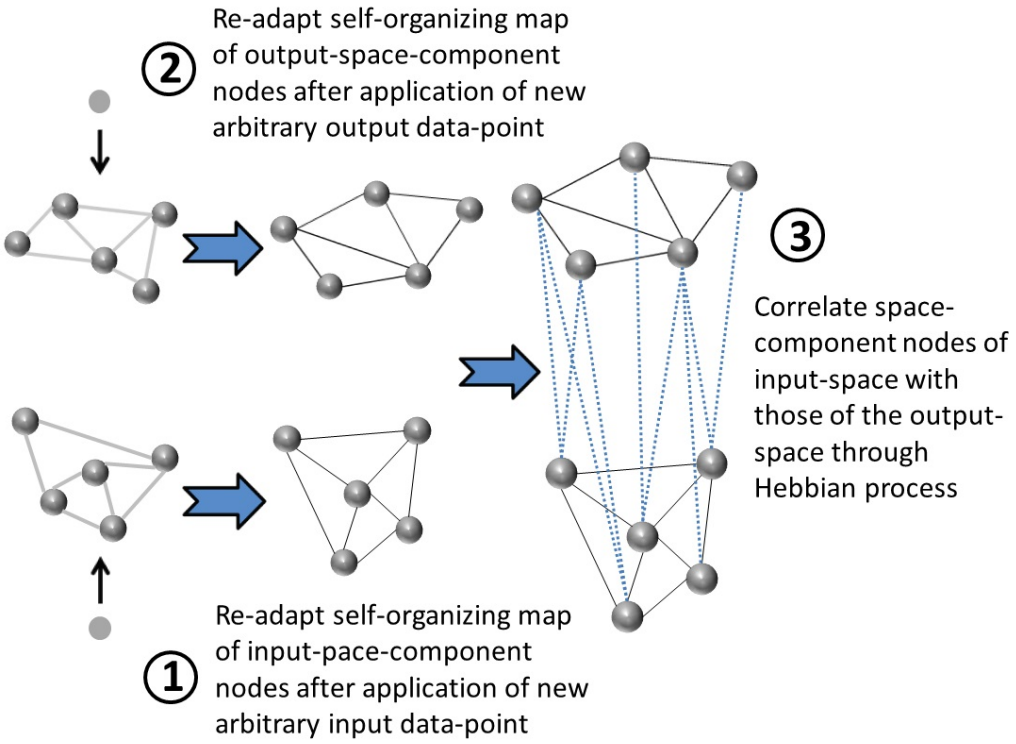

FIGURE 4.3: 2D circle represents an arbitrary data point. 3D spheres represent space-component nodes.

time a new data-point is applied SOINN would re-organize itself,as explained in section 3.3, in incremental fashion to facilitate on-line process of learning the set of representative space-component nodes that makes up each space.

This process takes place when learning the set of space-component nodes for both input and output spaces. i.e. for each learning experience, $E_k : [X_k, Y_k]$, the input-data-point, X_k, is fed into the self-organizing map, $S_{input} : \{C_{in_1}, C_{in_2}, ...C_{in_m}\}$, of input-space, resulting in re-adaptation of node-distribution within the space. The same thing happens after obtaining the correct output-data-point Y_k, either from environment or by explicit teacher, and feeding it into the output-space map $S_{output} : \{C_{out_1}, C_{out_2}, ...C_{out_n}\}$.see Figure 4.3.

Since each of input and output spaces are represented by space-component nodes,

and because any arbitrary data-point, within each space, is simply coded as an activation pattern of various space-component nodes, then in order, for the architecture, to learn the functional relation between arbitrary data-points from these spaces, the architecture then needs only to learn a correlative relation that associate each space-component node from input-space with component nodes of the output-space. These learned associations between space-component nodes represent the backbone that governs the implicit relation between input and output spaces. The reason for that is that, in our proposed mechanism, the input-output relation between any two arbitrary data-points is obtained from the correlative associations that connect the space-component nodes that code these arbitrary data-points. Learning these associations requires, first, obtaining the receptive field activation level r_j for each input-space-component node C_{in_j} as response to the applied input-data-point X_k, then we need to obtain the receptive field activation level r_i for each output-space-component node C_{out_i} as a response to the acquired output-data-point Y_k. For example, The activation level r_i of the receptive fields that belongs to output-space-component nodes C_{out_i} as response to an applied data-point Y is governed by Eq. (4.1), or we can use a normalized version of Eq. (4.1), in order to insure that at any time at least one of the receptive fields is being activated[75], and that is by making the activation function to be:

$$r_i = f(Y) = \frac{\exp(-\|Y - C_{out_i}\|/2\sigma^2)}{\sum_{l=1}^{n} \exp(-\|Y - C_{out_l}\|/2\sigma^2)} \tag{4.5}$$

Where, Y: is output-data-point location within the output-space. σ: determines

the spread of the Gaussian. n is the total number of output-space component nodes. The same equation is employed in order to compute The activation level r_j of the receptive fields that belongs to input-space-component nodes C_{in_j} as response to an applied input-data-point X.

After obtaining the receptive field activation level of each space-component node from input and out data-spaces, synchronously activated input and output space-component nodes are correlated by mean of hebbian learning through the use of a variation of hebbian learning rule called Oja rule[76] as follows:

$$\Delta w_{ij} = \eta(r_i r_j - r_i \sum_{l=1}^{n} w_{lj} r_l) \tag{4.6}$$

where: w_{ij}: is the weight of the associative connection between the input-space-component node C_{in_j} and output-space-component node C_{out_i}. η: is the learning rate. r_i: is the activation level of the output-space-component node C_{out_i}. r_j: is the activation level of the input-space-component node C_{in_j}. n is the total number of output-space component nodes.

The purpose of incorporating hebbian process, after the application of each learning experience $E_k : [X_k, Y_k]$, is to makes sure that space-component nodes from input and output spaces, that were activated at the same time ,are recorded and learned as co-active regions reflecting implicit functional relations between these spaces.

So each time a space-component node in the input-space is activated, the architecture would detect the set and pattern of responding space-component node from the output-space. These responding output-space-component nodes together with the active input-space-component node would be considered as co-active component nodes that reflect functional implicit relation between the input-space and the output-space, and thus this relation needs to be captured and recorder through hebbian learning.

4.2.4.3 Output prediction

When we want to predict the output value for a given input, then a process demonstrated in Figure 4.4 is employed. First of all when an input-data-point, shown as small 'x' in Figure 4.4.a, is presented to the network, its location within the input-space would activate the receptive fields of some of the input-space-component nodes,using Eq. (4.5).

Then, The activated input-space-component nodes would activate, in their turn, some of the output-space-component nodes that are connected with them.
The influence that the activated input-space-component nodes would demonstrate on the activation of output-space-component nodes is mediated by the weight of hebbian connection between them after being modulated by the activation level of input-space-component nodes, Figure 4.4.b. In other words, the application of input-data-point X_k would evoke a response, r_j, in the receptive field of some

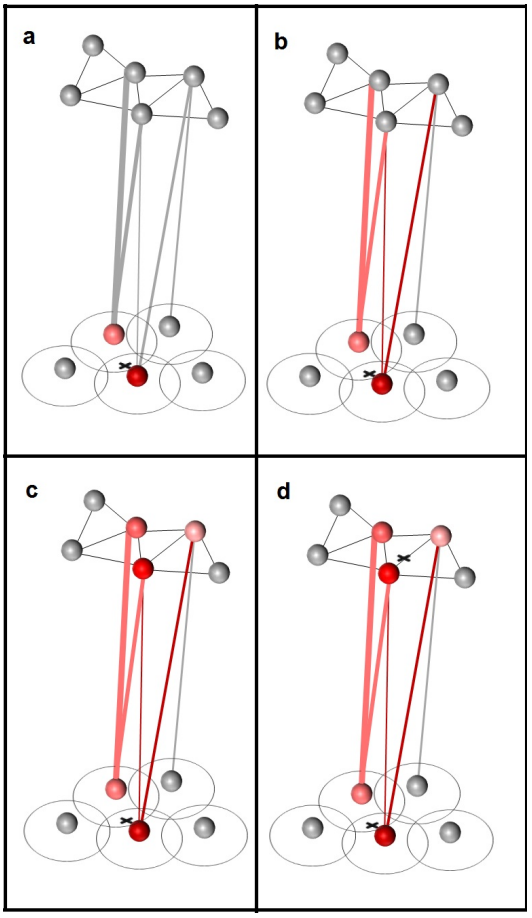

FIGURE 4.4: a: the location of input-data-point within the input-space would activate the receptive field of some space-component nodes. level of redness represents level of activation. b: modulation of connection efficiency by the activation level of input-space-component nodes. The thickness of connections represent weight strength. c: the ultimate activation level of each output-space-component node is what determines the prediction output value.

input-space-component nodes, this respond constitute an activation-pattern coding the data-point X_k. And because we don't have an actual value of Y_k to activate the receptive field of output-space-component nodes, then we need to estimate the activation pattern, $\hat{r}_i$, of these output-space-component nodes in order to have an estimated coding of the predicted output-data-point,$\hat{Y}_k$. Estimating the activation level, $\hat{r}_i$, of each output-space-component nodes, C_{out_i}, is driven by, both, the efficiency, w_{ij}, of each activated input-space-component node to provide a response in

this output-space-component node, and the degree, r_j, at which the input-space-component node has been activated in response to X_y. This is demonstrated in Eq. (4.7), where $\hat{r}_i$: is an estimation of the activation of output-space-component node C_{out_i}. r_j: is the recorded activation level of each input-space-component node, C_{int_j}, when the input-data-point X_k was applied. w_{ij}: is the weight of connection from each input-space-component node, C_{in_j}, to the output-space-component node who's activation level is being estimated.

$$\hat{r}_i = \frac{\sum_j w_{ij} r_j}{\sum_j w_{ij}} \tag{4.7}$$

Now after we have estimated an activation pattern for output-space-component nodes, we need to use this estimated pattern as coding for the predicted output-data-point $\hat{r}_i$. i.e, we need to derive the output-point itself from the estimated activation pattern. The interpolatory process that is involved is demonstrated in Eq. (4.8). Here the predicted output $\hat{Y}_k$ is function of both the location of every output-space-component node C_{out_i} together with the, previously estimated, activation level $\hat{r}_i$ of this output-space-component node.

$$\hat{Y}_k = \frac{\sum_i \hat{r}_i C_{out_i}}{\sum_i \hat{r}_i} \tag{4.8}$$

So we can think of the predicted value $\hat{Y}_k$ as a combination of multiple output-space-component nodes C_{out_i}, where the contribution of each output-space-component node is determined by its estimated activation level.

4.2.4.4 On-line Incremental Learning

Within the architecture proposed, learning takes place in a continuous incremental manner. In other words, there is no need for the network, in order to learn the problem in hand, to go through separated training then testing phases. Rather, the network can start from a state of complete ignorance of the nature of the problem being learned, and then, through interaction with the environment, the network scaffolds into gradual shaping of internal structures and organizations that all together constitutes an implicit representation of the learned problem.
In order for such emergent organization of network elements to take place we need to close the loop between the network and the surrounding environment, as seen in Figure 4.5.

At each time-step, $_k$, when a new input-data-point X_k is entered into the network, this data-point would activate the receptive fields of some input-space-component nodes, Eq. (4.5). The recorded activation levels of input-space-component nodes together with the hebbian connections, between input and output space-component nodes, would provoke response in some output-space-component nodes according to Eq. (4.7). These responses are used, Eq. (4.8), in order to predict the value of output-point $\hat{Y}_k$, which would be applied to the surrounding environment as the network output.
After the actual output-data-point,Y_k , is measured, or obtained explicitly, from the environment, the network need to go through re-adaptation process in order to learn the whole learning experience, $E_k : [X_k, Y_k]$. This would gradually formulate

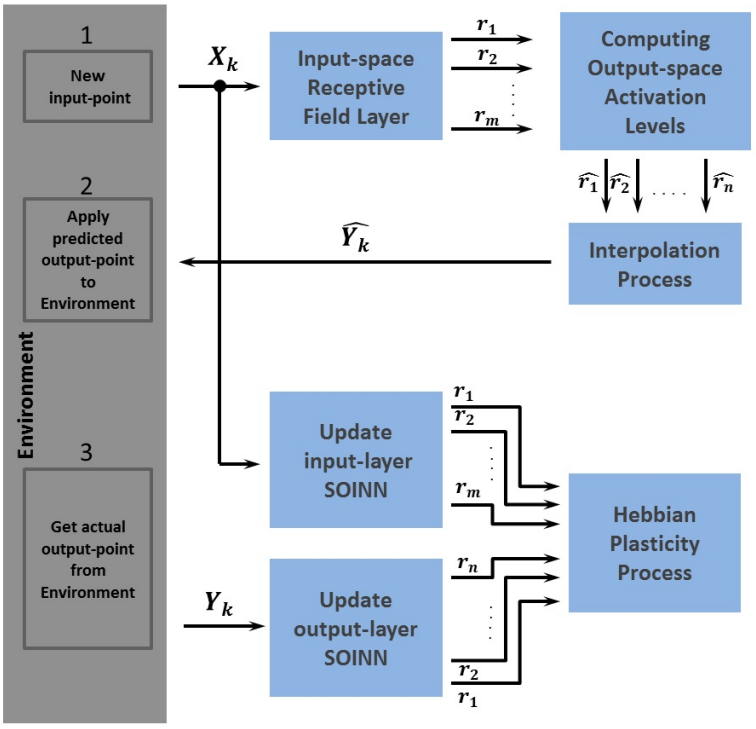

FIGURE 4.5: The network learns in an on-line incremental way through continuous interaction with the environment.

and shape the structure of the sets of component nodes representing both of the input-space and the output-space and the hebbian connection that correlate these spaces. This incremental process is summarized by re-organizing SOINN maps of both input and output spaces, as explained in section 3.3, and then re modifying the weights of the correlating connections thought a hebbian plasticity process Eq. (4.6), see Figure .4.6.

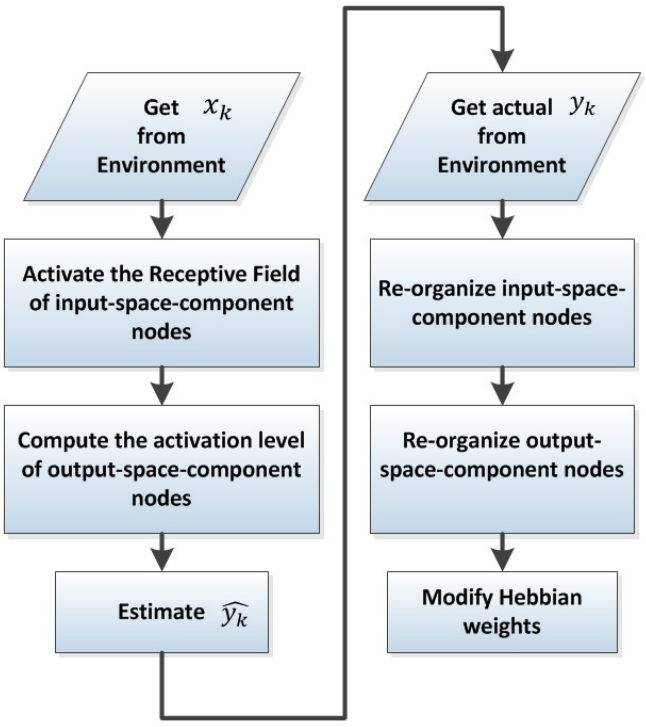

FIGURE 4.6: Learning takes place in On-line incremental manner where the loop between the network and the environment is closed.

4.3 The Experiments

In this section we demonstrate the network ability to perform on-line incremental learning through continues application of learning experiences, $E_k : [X_k, Y_k]$. Multiple experiments are performed; one of them is a real-world regression problem, and the other one is from robotics field in order to test the network ability to learn by interaction with the environment. One more experiment is performed, using 4 DoF robotic arm, in order to compare performance of RF-SOINN neuron with that of regular SOINN neurons.

4.3.1 Pressure-endurance regression experiment

In this experiment the network is required to learn a regression problem for predicting pressure-endurance value using a data set from KEEL Data set repository, [77]. The task in this experiment, is to learn the amount of pressure a given plastic material can withstand under applied force and temperature.

The input-data-point part of each learning experience, $E_k : [X_k, Y_k]$, is represented as two-dimensional vector,$X_k = [x_1, x_2]$, where x_1 is the applied force. x_2 is material temperature. On the other hand, the The output-data-point part, Y_k, is one-dimensional vector that represents the estimated pressure that the material can withstand. The employed data set consist of over 1600 entries of real-world collected data[77].

As mentioned previously, the purpose of this experiment is to demonstrate the network capability of on-line, incremental learning so there are no separate training and testing phases, hence the data set was not divided into training set and testing set, but rather the whole data set is treated as testing data set. In other words,as mentioned in section 4.2.4.4, each cycle takes place by first applying X_k into the network that would compute an estimation of pressure value output, $\hat{Y}_k$, then a training feed-back is provided to the network through the application of the actual output value, Y_k, Figure 4.5.

This incremental training feed-back enables the network to improve its ability

to learn the underlying pressure endurance function by gradual adaptation of its input and output self-organizing maps, Figure 4.3.

A demonstration of network capability to learn in an on-line, incremental fashion is illustrated in Figure 4.7, where the network's prediction error was collected during three experimental runs then the average error was plotted against the number of applied learning experiences.

We notice a clear reduction in prediction error with more application of learning experiences to the network.

We reemphasize that this reduction in error was not recorded after a separated phase of training with dedicated training data set, but rather it was recorded during the incremental learning process through instance-by-instance application of learning experiences.

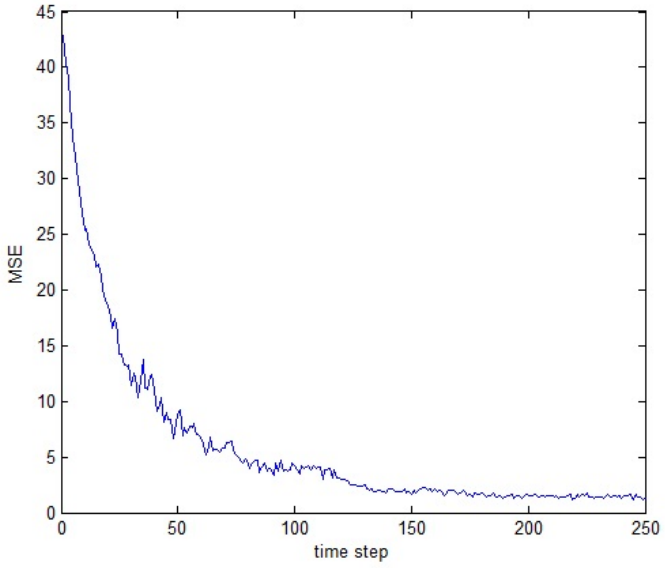

FIGURE 4.7: Prediction Error evolution during application of learning experiences to the network.

In Figure 4.8, a scatter plot of the actual output values, Y_k, on the horizontal axis is plotted against the predicted output values, $\hat{Y}_k$, on the vertical axis.

We notice that the model places most of the dots along the diagonal. Hence the model ability to approximate output values with a presence of some unbiased prediction error.

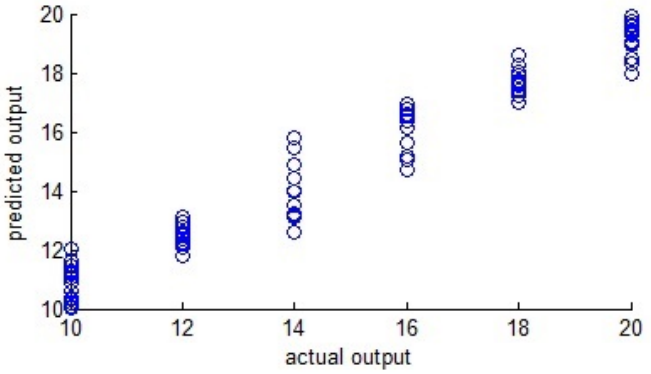

FIGURE 4.8: A scatter plot of actual output values on horizontal axis against predicted output values on the vertical axis.

A group of collected residuals are sampled in Figure 4.9 where the distribution of collected residuals shows the ability to predict a normal distribution in the population. Notice from Figure 4.9 that the normality assumption is likely to be true as seen by the even distribution of residuals around zero.

4.3.2 Robotic arm reaching-task experiment

In this experiment we demonstrate the proposed network ability to serve as the core learning-engine for acquiring implicit internal representation of system-environment contingencies. The goal is to teach a planner 2DoF robotic arm, similar ti the one in section 3.6.1, to perform target-reaching tasks without providing it with any

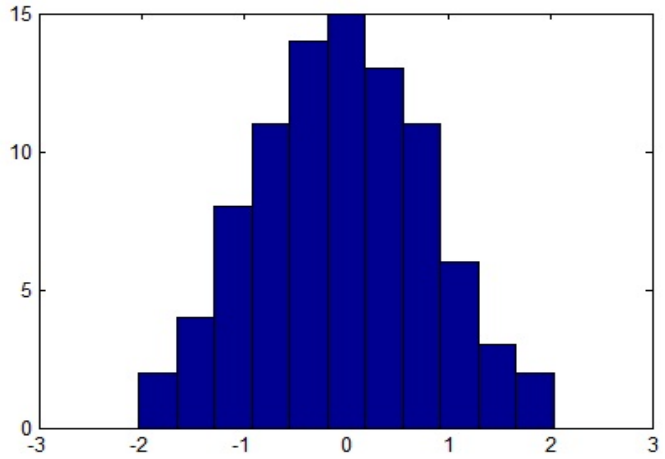

FIGURE 4.9: Histogram of collected residuals.

mechanism for performing forward or inverse kinematics or any explicit control model. In other words, the robot was not provided with any knowledge about how to control its joints and even we did not preprogram the robot with any action-consequence model beforehand of learning.

At each learning experience, a random reach-target is generated and then the robot is asked to achieve this location. After the robot performs its motor action in attempt to reach the target location, the actual resultant end-effector location is used to train the network. This training feedback is essential to guarantee an incremental bootstrapping ,of the implicit model formed by the network, toward higher competence of target reaching capability, where training and performing take place side by side without the need for adopting separated-phases-approach to learning. So a learning experience, $E_k : [X_k, Y_k]$, is a sensory-motor coupling of the form $E_k : [[l_x, l_y], [\theta_1, \theta_2]]$. Hence, $X_k = [l_x, l_y]$ represents the resultant end-effector location as a point in the Cartesian-space, and $Y_k = [\theta_1, \theta_2]$ is a point in C-space representing arm's configuration that is associated with the corresponding

Cartesian-space point.

Monitoring the arm performance during system-environment interaction, Figure 4.10, reveals a clear degradation of actuation error, which implies a clear Improvement of the network capability to capture system-environment contingencies through sensory-motor learning.

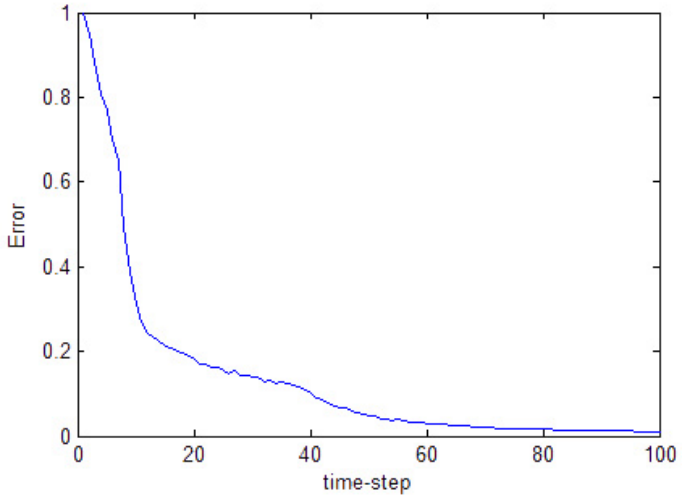

FIGURE 4.10: performance of the target reaching capability through sensory-motor learning.

The accuracy of the suggested configuration, $\hat{Y}_k = [\hat{\theta}_1, \hat{\theta}_2]$, would depend on the state of sophistication of the learned contingencies between joint actuation and the resultant consequences in the environment. So at early stages of learning, this suggested configuration won't offer high-accuracy solutions. But gradually, with more experience and as the system explore the contingency space, more accurate associations would be learned by the network.

4.3.3 RF-SOINN vs. Classic SOINN Performance (4 DoF Robotic Arm Reaching Experiment)

In order to demonstrate the performance of RF-SOINN neurons over regular SOINN neurons when used as coding elements within the proposed network, we designed an experiment similar to the one in section 4.3.2 but using a robotic arm with higher degree of freedom (4 DoF), Figure .4.11

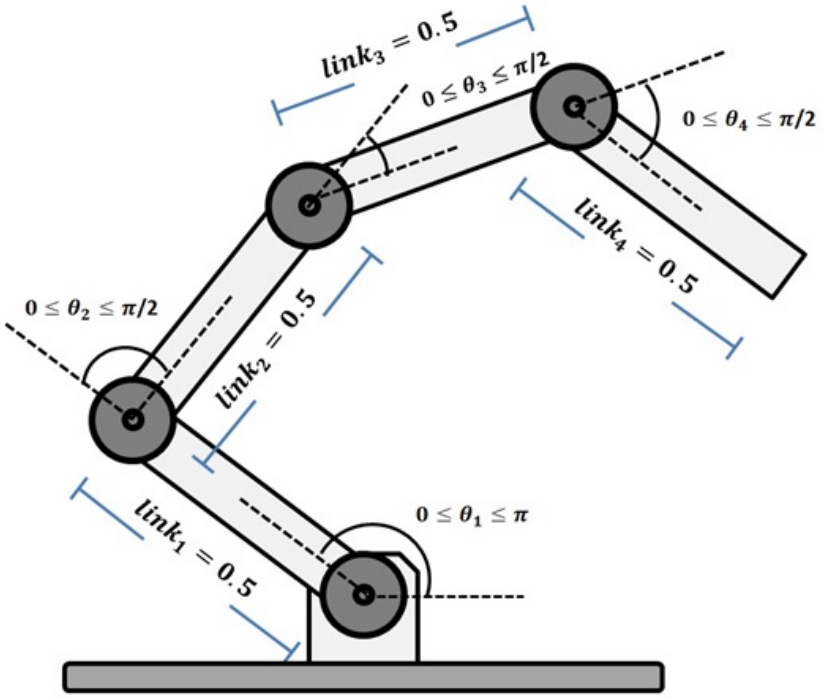

FIGURE 4.11: A reaching target is generated randomly and then the robot is asked to reach it with its end-effector, then, after the robot trail to reach the target, a new location is generated whether the robot has managed actually to reach the target or not.

The performance of the proposed network is compared with two other networks; The first one is SOIAMM network[78], which is a version of SOINN with regression

capability. The second network is exactly similar to the proposed network but equipped with regular SOINN neuron instead of RF-SOINN neurons.

After running the experiment and collecting measured error data we notice clearly that the proposed network with RF-SOINN neuron has superior performance to those of the other tow networks, Figure .4.12. We notice that RF-SOINN neurons give better performance when used within the proposed network than regular SOINN neurons.

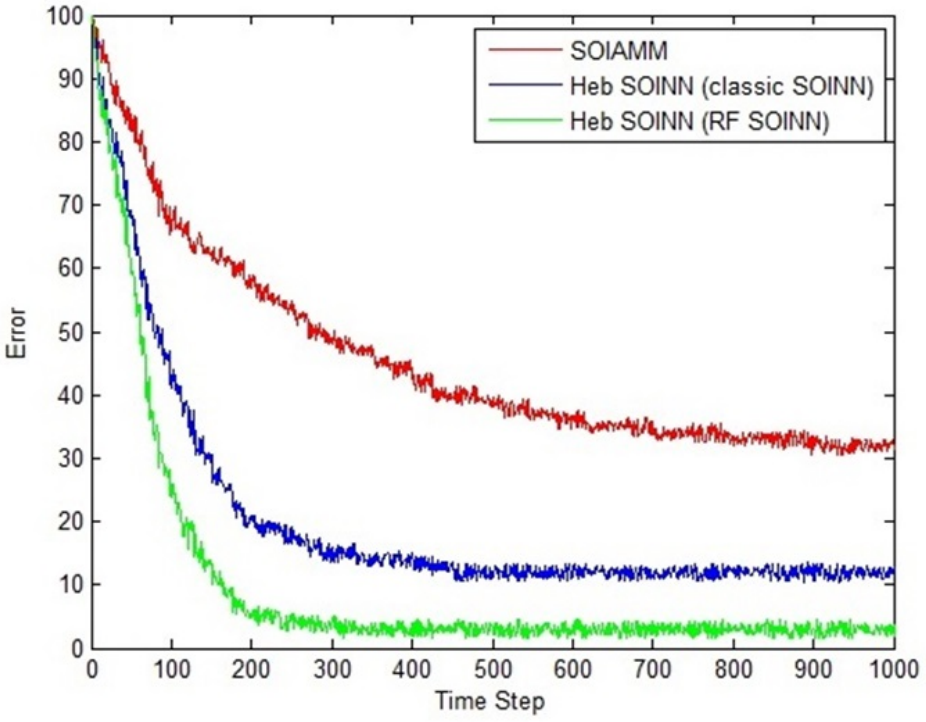

FIGURE 4.12: the proposed network with RF-SOINN neuron has superior performance to those of the other tow networks.

A possible explanation for such better performance, when RF-SOINN is used, is that having a Gaussian activation function enable better and more accurate coding of a data point within the data space.

Actually using Gaussian function means that the topological location of the stimulus within the data space is more accurately recorded among the active neurons than when a step function is used.

4.3.4 Discussion

As we have mentioned above, in section 4.2.4, we model a given data space by a set of representative data points that are elements of a self-organizing map reflecting zones of persistent stimulation within the data space.
These SOINN maps, "RF-SOINN maps" to be accurate, make up the essence of data-space modeling within the network. Hence, it is responsible of problem-domain coding and representation, which makes it the core incremental engine that gives the network its on-line, incremental features.

The fact that SOINN is unsupervised learning mechanism that does not require any prior knowledge about the problem in hand, gives it the ability to bootstrap its topological structure with more application of learning experiences from the surrounding environment. This continuous re-shaping and organizing of SOINN's node topology is governed by node-dynamics, Figure 3.5, that reacts to stimulation tendencies within data-space.

The consistent distribution of each space-map nodes reveals the nature of SOINN ability to cover a given data domain in a compact manner that allows the generalization beyond the recorded map elements. It is this spanning structure of

convergent map, what gives the receptive field mechanism its perceptual coding characteristics that aids output prediction within the network. Besides, the receptive-field-inspired mechanism for activating each node is the key factor behind more accurate coding ability than traditional SOINN nodes as was illustrated in section 4.3.3.

4.4 Summary

In this chapter we have presented a novel network that is capable of online, incremental learning through the incorporation of cortically-inspired mechanisms as self-organizing maps, receptive fields and hebbian plasticity.

The main idea is based on capturing the input-space and output-space of the problem in hand through self-organizing maps of receptive field neurons that would make up the core coding elements within each space. Then the correlative associations between these core elements are learned by means of hebbian process.

The learning ability of the proposed architecture is demonstrated through multiple experimental setups; one from real-world data regression problem , and two other experiments from robotic arm sensory-motor learning problem.

Chapter 5

Conclusions

Getting a robot to achieve a given task can be a daunting job due to the amount of manual work and preprogramming that is required on the engineer's side. This preparatory work, includes task formulation, operation-scenario specification, concepts and symbol definition and algorithm hard-wiring.

The unstructured nature of our daily environment, together with the complexity and diversity of perceptual concepts and objects within this environment, make this job even more difficult and unpractical. Besides, let's not forget the fact that we cannot program for every scenario in our world that the robot might encounter. Learning robots seem like a solution to this problem, but actually the problem of manual symbolic abstraction and task customization boils-up to the surface again. Recently, Developmental Robotics has been endorsed as the paradigm for designing robotic systems capable of learning without an explicit presence of the human-factor in the picture. Unfortunately, no clear systematic schema for realizing such

developmental approach to robot intelligence has been proposed so far.

In this Thesis we address this problem by building a Robot Control Architecture that adopt the mechanism of Motor Babbling which describes the way infants advance through early stages of development within their first months of life. Motor Babbling is the autonomous process of forming an internal implicit model of a moving mechanical system through Hebbian-like action-perceptions episodes.
The proposed learning-through-babbling architecture is a mechanism for learning sensory-motor associations using layered arrangement of Self-Organizing maps and joint-egocentric representations. Based on this architecture the robot starts of by random exploratory motion, then it gradually shifts into goal-directed actions based on the measure of error-change.
Experimental results showed that adopting motor babbling proved to be a promising approach toward the generation of internal models and control policies for robotic arms. Equipped with babbling architecture the robot managed to learn sensory-motor contingencies and joint-control shames in an incremental on-line manner. Besides, unanticipated sudden structural changes to the robotic platform were learned and compensated for in adaptive and reactive way.

After building babbling control architecture, we move on into designing the core learner itself that lays at the center of the robot's learning system. This step was motivated by the fact that interaction with the environment can provide a sparse and discrete set of sample correlations of input-output incidences. These incidences of associative data points can provide useful hints for capturing underlying

mechanisms that governs the system's behavioral dynamics. In many approaches to solving this problem, of learning the System's input-output relation, a set of previously prepared data points need to be presented to the learning mechanism, as a training data, before a useful estimations can be obtained.

So in this Thesis we proposed a novel neural network as an Incremental learning mechanism that can bootstrap from a state of complete ignorance of any representative sample associations. The proposed neural network captures the problem's data space components using emergent arrangement of receptive field neurons that self-organize incrementally in response to sparse experiences of system-environment interactions. These learned components are correlated using a process of hebbian plasticity that relate major components of input space to those of the output space. The viability of the proposed network has been validated through multiple experimental setups from both regression and robot motor learning problems.

- It has been found that enabling a robot to learn how to control arm joints for goal-directed reaching tasks is one of the earliest skills that need to be acquired by developmental robotics in order to scaffold into tasks of higher Intelligence.

- Associative Learning plays a major role in the formation of the internal dynamic engine of an adaptive system or a developmental robot.

- Learning through babbling makes a viable framework for realizing developmental approach to robot intelligence.

- Babbling-based mechanisms have the ability to adapt to unanticipated changes to the system itself or the operational conditions.

- Cortical mechanisms of learning can give us valuable insights in order to build better learning algorithms.

- Learning is NOT purely developmental but rather is build and based on seed-concepts and knowledge elements intrinsically embedded into the learner before the actual interaction with the world around can take place.

5.1 Summary

The work presented in this research is not final nor problem-free of course. Actually our work is far from finished, as many points need to be tackled more thoroughly. Our experimental results need to be improved by further modifications and optimization of the proposed algorithms.

5.1.1 Future Work

Basically we can summarize the future road map of our research by the following points:

- More architectural optimization modifications need to be performed in order to fine-tune some internal parameters of the algorithm so a better learning performance can be achieved.

- The question of transfer of learning from one platform into another was not tackled in the current state of our work, we need to investigate it in the future.

- The incorporation of elements of reward and motivation into the core mechanism of learning-through-babbling that we presented.

- In our research the system is not relaying on an intrinsic space coding and localization approach. That is why, we are thinking of working on building localization mechanism that enables the system to formulate implicit neuronal coding of the surrounding peripersonal space, thus providing the system with localization technique in which any arbitrary location in the peripersonal space is rather represented as an activation level within certain nodes of coordination-oriented neural circuitry, which would spare us the need to relay on an explicit Cartesian coordinate coding. We expect such

approach to have computational advantages, in term of time and complexity, over traditional Cartesian coordinate approach.

.

Bibliography

[1] Alan M Turing. Computing machinery and intelligence. *Mind*, pages 433–460, 1950.

[2] Juyang Weng. A theory for mentally developing robots. In *Development and Learning, 2002. Proceedings. The 2nd International Conference on*, pages 131–140. IEEE, 2002.

[3] Edward A Feigenbaum, Julian Feldman, and Paul Armer. *Computers and thought*. AAAI press, 1995.

[4] Bernard J Baars and Nicole M Gage. *Cognition, brain, and consciousness: Introduction to cognitive neuroscience*. Academic Press, 2010.

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

[6] Ronald C Arkin. *Behavior-based robotics*. MIT press, 1998.

[7] Juyang Weng, James McClelland, Alex Pentland, Olaf Sporns, Ida Stockman, Mriganka Sur, and Esther Thelen. Autonomous mental development by robots and animals. *Science*, 291(5504):599–600, 2001.

[8] Lisa A Meeden and Douglas S Blank. Introduction to developmental robotics. 2006.

[9] Alexander Stoytchev. Some basic principles of developmental robotics. *Autonomous Mental Development, IEEE Transactions on*, 1(2):122–130, 2009.

[10] R Sutton. Verification, the key to ai. *on-line essay.[Online]. Available: http://www. cs. ualberta. ca/sutton/IncIdeas/KeytoAI. html*, 2001.

[11] Juyang Weng. Task muddiness, intelligence metrics, and the necessity of autonomous mental development. *Minds and Machines*, 19(1):93–115, 2009.

[12] Rolf Pfeifer and Christian Scheier. *Understanding intelligence*. MIT press, 1999.

[13] Minoru Asada, Koh Hosoda, Yasuo Kuniyoshi, Hiroshi Ishiguro, Toshio Inui, Yuichiro Yoshikawa, Masaki Ogino, and Chisato Yoshida. Cognitive developmental robotics: a survey. *Autonomous Mental Development, IEEE Transactions on*, 1(1):12–34, 2009.

[14] Max Lungarella, Giorgio Metta, Rolf Pfeifer, and Giulio Sandini. Developmental robotics: a survey. *Connection Science*, 15(4):151–190, 2003.

[15] Minoru Asada, Karl F MacDorman, Hiroshi Ishiguro, and Yasuo Kuniyoshi. Cognitive developmental robotics as a new paradigm for the design of humanoid robots. *Robotics and Autonomous Systems*, 37(2):185–193, 2001.

[16] Juyang Weng and Wey-Shiuan Hwang. From neural networks to the brain: Autonomous mental development. *Computational Intelligence Magazine, IEEE*, 1(3):15–31, 2006.

[17] Alexander Stoytchev. Five basic principles of developmental robotics. In *NIPS Workshop on Grounding, Perception, Knowledge, and Cognition*, 2006.

[18] John J Weng. The developmental approach to intelligent robots. In *IN AAAI SPRING SYMPOSIUM SERIES, INTEGRATING ROBOTIC RESEARCH: TAKING THE NEXT LEAP*. Citeseer, 1998.

[19] Jordan Zlatev and Christian Balkenius. Introduction: Why òepigenetic roboticsó? 2001.

[20] Jean Piaget. *The child's conception of the world*, volume 213. Rowman & Littlefield, 1929.

[21] Jean Piaget and Margaret Trans Cook. The origins of intelligence in children. 1952.

[22] Juyang Weng. On developmental mental architectures. *Neurocomputing*, 70 (13):2303–2323, 2007.

[23] Douglas Blank, Deepak Kumar, Lisa Meeden, and James B Marshall. Bringing up robot: Fundamental mechanisms for creating a self-motivated, self-organizing architecture. *Cybernetics and Systems: An International Journal*, 36(2):125–150, 2005.

[24] Christopher G Prince and Lakshmi J Gogate. Epigenetic robotics: behavioral treatments and potential new models for developmental pediatrics. *Pediatric research*, 61(4):383–385, 2007.

[25] Luc Berthouze and Max Lungarella. Motor skill acquisition under environmental perturbations: On the necessity of alternate freezing and freeing of degrees of freedom. *Adaptive Behavior*, 12(1):47–64, 2004.

[26] Max Lungarella and Luc Berthouze. On the interplay between morphological, neural, and environmental dynamics: a robotic case study. *Adaptive Behavior*, 10(3-4):223–241, 2002.

[27] Paschalis Veskos and Yiannis Demiris. Developmental acquisition of entrainment skills in robot swinging using van der pol oscillators. 2005.

[28] Brian Scassellati. Theory of mind for a humanoid robot. *Autonomous Robots*, 12(1):13–24, 2002.

[29] Christian Balkenius, Petra Björne, et al. Toward a robot model of attention-deficit hyperactivity disorder (adhd). *Proceedings of epigenetic robotics: Modeling cognitive development in robotic systems*, 2001.

[30] Sergio Roa, GM Kruijff, and Henrik Jacobsson. Curiosity-driven acquisition of sensorimotor concepts using memory-based active learning. In *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on*, pages 665–670. IEEE, 2009.

[31] J Michael Herrmann, Klaus Pawelzik, and Theo Geisel. Learning predictive representations. *Neurocomputing*, 32:785–791, 2000.

[32] Jürgen Schmidhuber. Curious model-building control systems. In *Neural Networks, 1991. 1991 IEEE International Joint Conference on*, pages 1458–1463. IEEE, 1991.

[33] Sridhar Mahadevan and Jonathan Connell. Automatic programming of behavior-based robots using reinforcement learning. *Artificial intelligence*, 55(2):311–365, 1992.

[34] Rudolf Lioutikov, Alexandros Paraschos, Jochen Peters, and Gerhard Neumann. Sample-based informationl-theoretic stochastic optimal control. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 3896–3902. IEEE, 2014.

[35] Gerhard Neumann, Wolfgang Maass, and Jan Peters. Learning complex motions by sequencing simpler motion templates. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 753–760. ACM, 2009.

[36] Ozgür Simsek and A Barto. Betweenness centrality as a basis for forming skills. Technical report, Technical report, University of Massachusetts, Department of Computer Science, 2007.

[37] Nuttapong Chentanez, Andrew G Barto, and Satinder P Singh. Intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pages 1281–1288, 2004.

[38] Micha Hersch, Eric Sauser, and Aude Billard. Online learning of the body schema. *International Journal of Humanoid Robotics*, 5(02):161–181, 2008.

[39] Jürgen Sturm, Christian Plagemann, and Wolfram Burgard. Body schema learning for robotic manipulators from visual self-perception. *Journal of Physiology-Paris*, 103(3):220–231, 2009.

[40] Giorgio Metta, Giulio Sandini, and Jurgen Konczak. A developmental approach to visually-guided reaching in artificial systems. *Neural networks*, 12 (10):1413–1427, 1999.

[41] Daniele Caligiore, Domenico Parisi, and Gianluca Baldassarre. Toward an integrated biomimetic model of reaching. In *Development and Learning, 2007. ICDL 2007. IEEE 6th International Conference on*, pages 241–246. IEEE, 2007.

[42] Yiannis Demiris and Anthony Dearden. From motor babbling to hierarchical learning by imitation: a robot developmental pathway. 2005.

[43] Jens Kober and Jan Peters. Learning motor primitives for robotics. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 2112–2118. IEEE, 2009.

[44] Min Shuo Li and Ming Hai Yao. Cognitive developmental grasp robots: From infant development to computational modeling. *Advanced Materials Research*, 706:682–686, 2013.

[45] Eric Wai Ming Lee. An incremental adaptive neural network model for online noisy data regression and its application to compartment fire studies. *Applied Soft Computing*, 11(1):827–836, 2011.

[46] Hung Q Ngo, Matthew D Luciw, Alexander Förster, and Juergen Schmidhuber. Learning skills from play: Artificial curiosity on a katana robot arm. In *IJCNN*, pages 1–8, 2012.

[47] Jeremy L Wyatt, Peter Dayan, Ales Leonardis, and Jan Peters. Exploration and curiosity in robot learning and inference (dagstuhl seminar 11131). *Dagstuhl Reports*, 1(3), 2011.

[48] Quoc V Le. Building high-level features using large scale unsupervised learning. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8595–8598. IEEE, 2013.

[49] Julien Hubert, Eiko Matsuda, and Takashi Ikegami. Hebbian learning in a multimodal environment. In *Advances in Artificial Life, ECAL*, volume 12, pages 698–705, 2013.

[50] C Gyorodi, R Gyorodi, George Pecherle, Livia Bandici, and Mihai Dersidan. An improved hebbian neural network with dynamic neuronal life and

relations. In *Soft Computing Applications (SOFA), 2010 4th International Workshop on*, pages 233–236. IEEE, 2010.

[51] Cengiz Pehlevan, Tao Hu, and Dmitri B Chklovskii. A hebbian/anti-hebbian neural network for linear subspace learning: A derivation from multidimensional scaling of streaming data. *Neural computation*, 2015.

[52] Shuhaida Ismail, Ani Shabri, and Ruhaidah Samsudin. A hybrid model of self-organizing maps (som) and least square support vector machine (lssvm) for time-series forecasting. *Expert Systems with Applications*, 38(8):10574–10578, 2011.

[53] Witali Aswolinskiy and Gordon Pipa. Rm-sorn: a reward-modulated self-organizing recurrent neural network. *Frontiers in computational neuroscience*, 9, 2015.

[54] Hao Liu and Xiao-juan Ban. Clustering by growing incremental self-organizing neural network. *Expert Systems with Applications*, 42(11):4965–4981, 2015.

[55] Shen Furao, Tomotaka Ogura, and Osamu Hasegawa. An enhanced self-organizing incremental neural network for online unsupervised learning. *Neural Networks*, 20(8):893–903, 2007.

[56] Shen Furao and Osamu Hasegawa. An incremental network for on-line unsupervised classification and topology learning. *Neural Networks*, 19(1):90–106, 2006.

[57] ALH Van der Meer, FR Van der Weel, David N Lee, et al. The functional significance of arm movements in neonates. *SCIENCE-NEW YORK THEN WASHINGTON-*, pages 693–693, 1995.

[58] Claes Von Hofsten. Eye–hand coordination in the newborn. *Developmental psychology*, 18(3):450, 1982.

[59] Pierre-Yves Oudeyer, Adrien Baranes, and Frédéric Kaplan. Intrinsically motivated learning of real-world sensorimotor skills with developmental constraints. In *Intrinsically motivated learning in natural and artificial systems*, pages 303–365. Springer, 2013.

[60] Rolf Pfeifer, Max Lungarella, Olaf Sporns, and Yasuo Kuniyoshi. *On the information theoretic implications of embodiment–principles and methods*. Springer, 2007.

[61] Itamar Arel, Derek C Rose, and Thomas P Karnowski. Deep machine learning-a new frontier in artificial intelligence research [research frontier]. *Computational Intelligence Magazine, IEEE*, 5(4):13–18, 2010.

[62] Pardis Noorzad and Bob L Sturm. Regression with sparse approximations of data. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pages 674–678. IEEE, 2012.

[63] Jin Li and Andrew D Heap. A review of comparative studies of spatial interpolation methods in environmental sciences: performance and impact factors. *Ecological Informatics*, 6(3):228–241, 2011.

[64] Gregory C DeAngelis, Izumi Ohzawa, and Ralph D Freeman. Depth is encoded in the visual cortex by a specialized receptive field structure. *Nature*, 352(6331):156–159, 1991.

[65] H Keffer Hartline. The response of single optic nerve fibers of the vertebrate eye to illumination of the retina. *Am J Physiol*, 121(2):400–415, 1938.

[66] Ch S Sherrington. Observations on the scratch-reflex in the spinal dog. *The Journal of Physiology*, 34(1-2):1, 1906.

[67] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106, 1962.

[68] Robert W Rodieck. Quantitative analysis of cat retinal ganglion cell response to visual stimuli. *Vision research*, 5(12):583–601, 1965.

[69] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology Press, 2002.

[70] Thomas H Brown, Alan H Ganong, Edward W Kairiss, Claude L Keenan, and Stephen R Delso. Long-term potentiation in two synaptic systems of the hippocampal brain slice. 1989.

[71] Masao Ito. Long-term depression. *Annual review of neuroscience*, 12(1): 85–102, 1989.

[72] Tim VP Bliss and Terje Lømo. Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *The Journal of physiology*, 232(2):331–356, 1973.

[73] Jean-Luc Gaiarsa, Olivier Caillard, and Yehezkel Ben-Ari. Long-term plasticity at gabaergic and glycinergic synapses: mechanisms and functional significance. *Trends in neurosciences*, 25(11):564–570, 2002.

[74] Thomas Trappenberg. *Fundamentals of computational neuroscience*. Oxford University Press, 2009.

[75] Stephen Marsland. *Machine learning: an algorithmic perspective*. CRC press, 2014.

[76] Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15(3):267–273, 1982.

[77] J Alcalá, A Fernández, J Luengo, J Derrac, S García, L Sánchez, and F Herrera. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, 17:255–287, 2010.

[78] Akihito Sudo, Akihiro Sato, and Osamu Hasegawa. Associative memory for online learning in noisy environments using self-organizing incremental neural network. *Neural Networks, IEEE Transactions on*, 20(6):964–972, 2009.