





**LEARNING TO SELECT AN APPROPRIATE ACTION:  
IMPLEMENTATION ON BIOLOID ROBOT**

**BSc Thesis by**

**Emeç ERÇELİK**

**Department: Electronics and Communication Engineering**

**Programme: Electronics Engineering Programme**

**MAY 2013**



**LEARNING TO SELECT AN APPROPRIATE ACTION:  
IMPLEMENTATION ON BIOLOID ROBOT**

**BSc Thesis by**

**Emeç ERÇELİK  
(040080301)**

**Department: Electronics and Communication Engineering**

**Programme: Electronics Engineering Programme**

**Supervisor: Assoc. Prof. Dr. Neslihan Serap Şengör**

**MAY 2013**



**UYGUN HAREKET SEÇİMİNİ ÖĞRENME:  
BIOLOID ROBOT ÜZERİNDE BİR UYGULAMA**

**BİTİRME ÖDEVİ**

**Emeç ERÇELİK  
(040080301)**

**Bölümü: Elektronik ve Haberleşme Mühendisliği**

**Programı: Elektronik Mühendisliği Programı**

**Danışmanı: Doç. Dr. Neslihan Serap Şengör**

**MAYIS 2013**



*To my parents and brother,*



## **FOREWORD**

Above all, I would like to thank to my thesis advisor, Assoc. Prof. Dr. Neslihan Serap Şengör for her valuable guidance and always supporting me when I felt myself lost in a labyrinth. She opened the doors of neuroscience while establishing the Neuroscience Modelling and Research Group whose members I also would like to thank to, especially Berat Denizdurduran for enlightening my way during this graduation project.

Most of the basis of the programming knowledge I used for my graduation project is built up in OTOKON(Control and Automation Student Club of ITU). I would like to thank to all OTOKON members and especially, D. Çağdaş Akalın, Ahmet Sakallı, Alper Yükselen and Ali Şakır. This is an opportunity to thank to Onur Varol to make me feel as if I am doing a great work.

I also would like to thank to Aslıhan Büşra Kop for always being with me, feedbacking on my works and making me feel grateful for my life.

Lastly, I would like to thank to all my family, especially my parents and brother for always supporting and trusting me on my steps.

This work is partially supported by TUBITAK project no: 111E264.

May 2013

Emeç ERÇELİK



## TABLE OF CONTENTS

	<u>Page</u>
<b>FOREWORD.....</b>	<b>ix</b>
<b>TABLE OF CONTENTS.....</b>	<b>xi</b>
<b>ABBREVIATIONS .....</b>	<b>xiii</b>
<b>LIST OF TABLES .....</b>	<b>xv</b>
<b>LIST OF FIGURES .....</b>	<b>xvii</b>
<b>SUMMARY .....</b>	<b>xix</b>
<b>ÖZET .....</b>	<b>xxi</b>
<b>1. INTRODUCTION .....</b>	<b>1</b>
<b>2. EMBEDDED C ON BIOLOID.....</b>	<b>3</b>
2.1 RoboPlus.....	6
2.2 Programming With Embedded C .....	7
2.2.1 Functions .....	9
2.2.2 An Example Application on Motors.....	9
<b>3. IMPLEMENTATING BTC MODEL ON BIOLOID .....</b>	<b>11</b>
3.1 Basal Ganglia-Thalamus-Cortex(BTC) Model .....	11
3.2 Bioloid Robot Implementation.....	13
3.2.1 Implementing the Foraging Task on Bioloid Robot .....	15
3.2.2 Programming Bioloid .....	15
3.2.3 Scaling and Normalizing .....	17
3.2.4 Basal Ganglia Circuit .....	18
3.2.5 Reinforcement Learning .....	19
<b>4. RESULTS .....</b>	<b>21</b>
<b>5. CONCLUSIONS AND RECOMMENDATIONS.....</b>	<b>31</b>
<b>REFERENCES.....</b>	<b>33</b>
<b>CURRICULUM VITAE .....</b>	<b>35</b>



## **ABBREVIATIONS**

<b>AS</b>	: Action Selection
<b>BG</b>	: Basal ganglia
<b>BTC</b>	: Basal ganglia- Thalamus-Cortex
<b>BTK</b>	: Basal ganglia-Talamus-Korteks
<b>Ctx</b>	: Cortex
<b>DMS</b>	: Distance Measurement Sensor
<b>GPe</b>	: Globus Pallidus external
<b>GPi</b>	: Globus Pallidus internal
<b>IR</b>	: Infrared Sensor
<b>RL</b>	: Reinforcement Learning
<b>SDK</b>	: Software Developement Kit
<b>Stn</b>	: Subthalamic Nucleus
<b>Str</b>	: Striatum
<b>TDL</b>	: Temporary Difference Learning
<b>Thl</b>	: Thalamus



## LIST OF TABLES

	<u>Page</u>
<b>Table 4.1</b> : Test results of learning durations for three actions. ....	21
<b>Table 4.2</b> : Initial values of parameters on learning operations.....	22



## LIST OF FIGURES

	<u>Page</u>
<b>Figure 2.1</b> : DMS distance-value graph. The first is distance according to voltage while the other is according to the digitalized value [13]. ....	3
<b>Figure 2.2</b> : USB2Dynamixel and its connection types. ....	4
<b>Figure 2.3</b> : CM-510 controller of Bioloid and its connection inputs for the peripherals. ....	5
<b>Figure 2.4</b> : RoboPlus and its subprogrammes to programme CM-510 and communicate with motors.....	6
<b>Figure 2.5</b> : Atmel Studio 6 interface. 1: Menu, 2:Generates new project, 3: Shows recent projects, 4: Shows results and outputs of operation ....	8
<b>Figure 3.1</b> : Block diagram of Basal ganglia circuit: This diagram shows the relations, which are defined as inhibitory or excitatory bounds between substructures of Basal ganglia, cortex and thalamus. ....	11
<b>Figure 3.2</b> : Environmental setup for foraging task. An educational robot assembled to model rat and moves on a platform that has a different reflection coefficient from nest.....	14
<b>Figure 3.3</b> : Block diagram of the operation in Bioloid. Operation begins with reading sensory data. The data used to make decision in basal ganglia circuit as an input and evaluate the learning rules.....	16
<b>Figure 3.4</b> : Scaling functions of the inputs: These functions take the useful inside the data of environmental sensory inputs. ....	18
<b>Figure 4.1</b> : Expectation error which becomes zero at the end of learning.....	22
<b>Figure 4.2</b> : Test1:(a),(b),(c) shows expectation errors and (d),(e),(f) shows variations on diagonal's Wc on the learning of search, food and nest respectively. ....	23
<b>Figure 4.3</b> : Test1:(a),(b),(c) shows cortex outputs of BTC which shows whether the robot choose an action.....	24
<b>Figure 4.4</b> : Test2:(a),(b),(c) shows expectation errors and (d),(e),(f) shows variations on diagonal's Wc on the learning of search, food and nest respectively. ....	25
<b>Figure 4.5</b> : Test2:(a),(b),(c) shows cortex outputs of BTC which shows whether the robot choose an action.....	26
<b>Figure 4.6</b> : Test3:(a),(b),(c) shows expectation errors and (d),(e),(f) shows variations on diagonal's Wc on the learning of search, food and nest respectively. ....	27
<b>Figure 4.7</b> : Test3:(a),(b),(c) shows cortex outputs of BTC which shows whether the robot choose an action.....	28
<b>Figure 4.8</b> : Test4:(a),(b),(c) shows expectation errors and (d),(e),(f) shows variations on diagonal's Wc on the learning of search, food and nest respectively. ....	29

**Figure 4.9 :** Test4:(a),(b),(c) shows cortex outputs of BTC which shows whether the robot choose an action..... 30

## **LEARNING TO SELECT AN APPROPRIATE ACTION: IMPLEMENTATION ON BIOLOID ROBOT**

### **SUMMARY**

Computational neuroscience is interdisciplinary, while it gives an explanation on functions of central nervous system, it also inspires robotic applications. These applications are recently becoming a tool to either functionalize or test the brain-inspired models with reverse engineering technique. At the same time, the brain inspired mathematical models has an important role on some robotic applications including patient care robots or rescue robots. The main point of this work is to realize a computational model on a real robot to give an example on the realization of a decision-making task and learning process.

A previously proposed computational basal ganglia-thalamus-cortex (BTC) model is used to realize foraging task on Bioloid robot. Also, temporal difference learning(TDL), which is a reinforcement learning method, is utilized to realize learning process. A distance sensor and an infrared sensor are used to establish the connection of robot to the outer world. A 5x5x5 cm box is defined as a food, while a black band on the ground represents the nest. Further, microcontroller of the robot is programmed in C programming language to embed the model on the mobile robot whose output for motors is defined due to the result of the decision making process.

Once the robot gets acquainted with the environment, robot is able to achieve the tasks of searching for food, recognizing the food and lifting it, and placing the food to the nest. Robot was tested eight times with the same initial values of parameters to investigate the robot's behaviour on a changing environment like changed food size, shape and position quantity of light and the position of nest. Robot has learned in all tests in varying environment. Robot is also tested with different initial values of parameters to investigate how it is behaving with a different perception.

In this graduation project, the BTC model is tested with a real time application on a real Bioloid robot instead of using a robot simulation environment. This work is an example of realizing the control of a mobile robot using a mathematical model based on the relation between brain regions responsible for a specific task. The expected outcome is the robot's learning to make right decisions and to repeat them in a different environment. At the end, robot is able to realize this task, on a changing environment with different parameters.



## **UYGUN HAREKET SEÇİMİNİ ÖĞRENME: BIOLOID ROBOT ÜZERİNDE BİR UYGULAMA**

### **ÖZET**

Disiplinler arası bir bilim olan hesaplamalı sinirbilim, merkezi sinir sistemine ilişkin açıklamalar getirirken, tersine mühendislik yaklaşımıyla robotik uygulamalarında özellikle son yıllarda yer edinmekte ve bu çalışmalar insan beynine ilişkin modellere test ortamı sağlamaktadır. Beyindeki süreçlerin oluşmasında etkin yapılaraya dayalı geliştirilen matematiksel modeller aynı zamanda hasta bakımı ya da tehlikeli ortamlarda arama çalışması yapılması gibi, karar vermenin önemli olduğu robotik uygulamaları için de öneme sahiptir. Böylelikle karar vermeye dayalı robotik uygulamalar gerçekleştirilmeye çalışılırken aynı zamanda biyolojik yapıların çalışmasının anlaşılması ve oluşan hastalıkların oluşma sürecinin anlaşılması için de önemli bir araştırma ortamı sunar. Bu bitirme projesi ile karar verme sürecine ait bir hesaplamalı modelin gerçek bir robot üzerinde uygulanması gerçekleştirilmiştir.

Bitirme projesi gerçekleştirirken Bioloid robot üzerinde, hesaplamalı Basal ganglia-talamus-korteks (BTK) modeli kullanılarak bir fareye ait yiyecek arama ve saklama davranışlarını gerçekleştirmesi ele alındı. Bu modelde robotun dış dünya ile ilişkisi, bir uzaklık sensörü ve bir kıızılıtesi sensör ile sağlandı. Bu sensörlerden alınan veriler sonucunda model, robotun mikrokontrolörüne C dili ile programlanarak gömündü. Bu model yardımıyla seçilen hareket ile sağlanan karar verme işlemi robotun motorlarına bir çıkış olarak gönderildi.

Gerçekleme sırasında robot küçük boyutta bir kutuya yem olarak, büyük boyutta bir kutuya ise engel olarak algılayabildi. Bulunulan ortamdan farklı bir yansımaya sahip bir siyah bant ise yuva olarak algılandı ve yem buraya yerleştirildi. Böylelikle robotun sırasıyla yiyecek arama, yiyeceği tanıyarak alma ve yuvaya bırakma işlemlerini gerçekleştirmiş oldu. Robot ile yapılan sekiz farklı öğrenme deneyi sonucunda aynı başlangıç değerlerine sahip parametreler ile değişen ortam şartlarındaki başarısı incelendi. Parametrelerin değerleri, robotun üzerinde yer alan dinamik doğrusal olmayan modelin çalışma uzayında seçildiğinde robotun değişen ortam şartlarında (yiyeceğin şekli, konumu, yuhanın konumu, ortamdaki ışık miktarı) tüm denemelerde başarılı olduğu görüldü. Ayrıca robotun çalışması, üzerinde yer alan parametrelerin başlangıç değerlerinin değiştirilmesi ile de incelendi. Bunun sonucunda, başlangıç değerleri farklı olsa da öğrenme tamamlandıktan sonra BTK devresi ile karar vermenin gerçekleşebildiği görüldü.

Daha önce yapılan çalışmalardan farklı olarak bu çalışmada model, bir benzetim ortamında değil gerçek bir Bioloid robot üzerinde test edildi. Yapılan çalışma, beyindeki süreçlerin oluşmasında yer alan yapılaraya dayalı matematiksel bir model ile bir robotun kontrolünün mümkün olduğuna dair bir örnek teşkil etmektedir. Aynı zamanda bu çalışmanın biyolojik gerçekliği daha yüksek olan bir model ile devam ettirilmesi sonucunda biyolojik yapıların ve hastalık süreçlerinin çalışma mekanizmasına açıklama getirebileceği de öngörmektedir.



## 1. INTRODUCTION

Cognitive robotics is a recent area in robotics where intelligent behavior is modelled in order to realize the action-perception cycle. In cognitive robotics, identifying the behavior is the important part of the task and depends not only on the robot's abilities but also on environmental conditions. The ultimate goal is to find a plausible method to provide robot to make the right high-level decisions in the right time while environmental conditions are changing. To reach this goal, agents need behavioral and cognitive skills such as environmental orientation, perceptual ability, action selection and learning. One of the major challenge of the cognitive robotic is how to bring these different sub-level processes together in order to reach the desired goals. Mimicking the brain functions would be an ideal approach to solve the complex environmental tasks. Though it is not explicitly known how brain manages with high-level problems, still there are models focusing on the brain's ability in high-level problems. Those works on identifying the differences between the desired actions provides inspiration for cognitive robotics. Recently there is an interest to show the ability of the brain-inspired computational models on robotic applications [1–4].

To reach the desired goal in a varying environment, two major component of the goal-directed behavior are required; action selection (AS) and learning. The model that is proposed in [5] is mimicking a part of brain structures called basal ganglia circuit [7, 8] and investigated further to understand its ability for action selection [6]. It is now well-known that basal ganglia circuit plays a key role in action selection [7]. Nevertheless, basal ganglia circuit's role in reward-modulated learning is also considered, recently. The learning method that is used in this graduation project is called temporal difference learning (TDL) and it is now claimed that TDL is the best matched learning method to the basal ganglia circuit's role in reward-modulated learning [9].

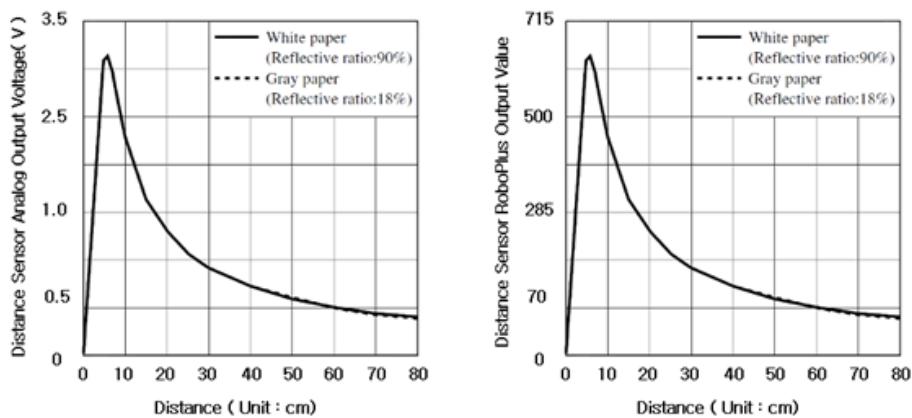
In the pioneered study of Prescott et.al. [1], a system level model of basal ganglia circuit is implemented on a mobile robot to demonstrate the model's ability for

action selection. Among different saliences or actions, model could switch the behavior when it is necessary. They used these saliences to control the mobile robot without any learning processes. They fixed the environmental conditions and assigned the coefficients before the task is started. Even though, this model does not have learning component, it has biologically plausible approach and leads to a new view in computational neuroscience [2, 10–12].

In the previous work related to this graduation thesis, the model is implemented on a mobile robot in simulation environment. The details about the proposed model can be found in [12]. The contribution of this thesis is to implement the action selection model and learning method on a real robot. The realization is done on Bioloid, so the computational model's ability to solve this high-level task even with limited sensor's ability of Bioloid is showed. Thus, to achieve this goal an explicit knowledge on programming Bioloid is required which will be given in Chapter 2. Matching the model with the C programming language is the next step, in Chapter 3, the BTC model will be focused on as it is used in programming and the structures of programming Bioloid will be mentioned. After the model is programmed on Bioloid, in Chapter 4, the tests with different parameters and their results will be given. The main point of this work is investigating the ability of the model on the real world to have a beginning for the high-level problems and a test environment of brain researches. To make a stride on the goal this thesis is presented in the 11<sup>th</sup> National Neuroscience Congress [17].

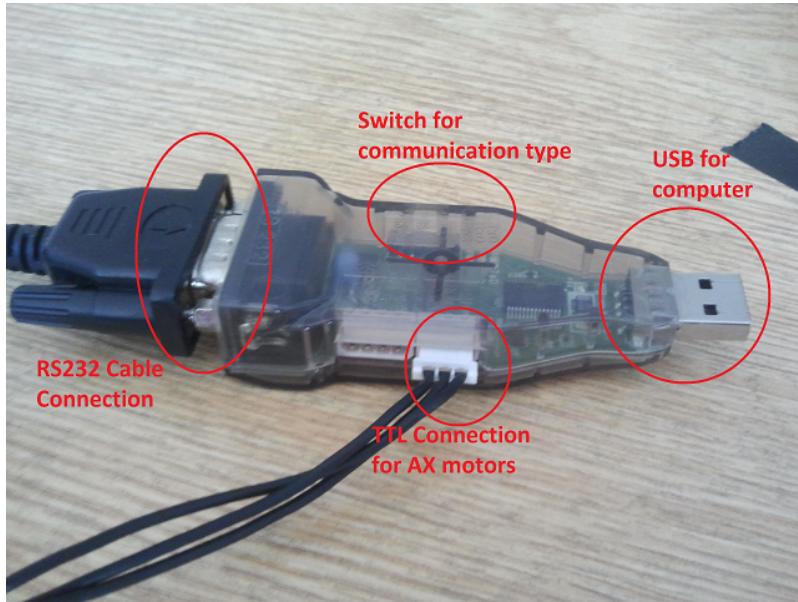
## 2. EMBEDDED C ON BIOLOID

Bioloid is an educational humanoid robot that can be modified in various forms [13]. Though Bioloid is not much suitable to improve a robot hardware, it can be used easily in designing test environments for robotic applications. Bioloid robot has motors and sensors on itself and a microcontroller to process the environmental data to act. In this part, how to programme these sensors and motors to communicate with environment is given. All the information to programme the Bioloid is retrieved from Robotis e-Manuel [13]. In Robotis e-Manuel, programming environment and examples to use peripheral devices of the controller are included. A CM-510 controller which includes an ATmega 2561 microcontroller inside, a distance measurement sensor(DMS) and an infrared sensor (IR) are utilized with AX-12+ model motors to realize the task which is mentioned in section 3. The programming environment consists of RoboPlus, which includes RoboPlus Task, RoboPlus Terminal, RoboPlus Manager and Dynamixel Wizard, and Atmel Studio 6. It is possible to reach the User's Guide by installing RoboPlus or by using web site [13]. The installation of RoboPlus can be downloaded from the web site of Robotis that has a 'Support' part which includes 'Download' section.



**Figure 2.1:** DMS distance-value graph. The first is distance according to voltage while the other is according to the digitalized value [13].

The DMS is used to detect the objects, which are in front of the robot, or the walls. The sensor converts the analog value that is related to the distance to digital value which



**Figure 2.2:** USB2Dynamixel and its connection types.

is between 0 and 1023. The distance-value graph is given in Figure 2.1. For the same color the object at a distant position has a higher value than the closer one. In Figure 2.1 it is shown that the color does not effect the returned value effectively.

The IR reflects an infrared light and senses the reflected light from a surface. The value that the sensor produces can vary according to light of environment, color and brightness of surface. This sensor also returns a discrete value between 0 and 1023. For the same color, IR gives a higher value for closer object or surface. For the same distance, it gives a higher value for lighter color.

Bioloid uses the Dynamixel AX-12+ motors of Robotis. Velocity and position of the motors can be controlled separately. There are two operating modes: joint and wheel modes. In joint mode position of motors can be controlled by sending motors a value between 0 and 1023 with velocity information. The turning direction is meaningless for this mode. In wheel mode, velocity and direction informations are important to make motors turn continuously as a wheel. Every motor has an ID and to communicate with a motor this ID is necessary. There are parameters to set the motors behaviour. All parameters have an address. Thus, motor ID is sent first and then parameter address. The address value is changed during the process and parameter value are sent to motors. The communication between computer and motors is provided by using USB2Dynamixel device or CM-510 controller. RoboPlus Manager or Dynamixel Wizard tools of RoboPlus programme are suitable for communication

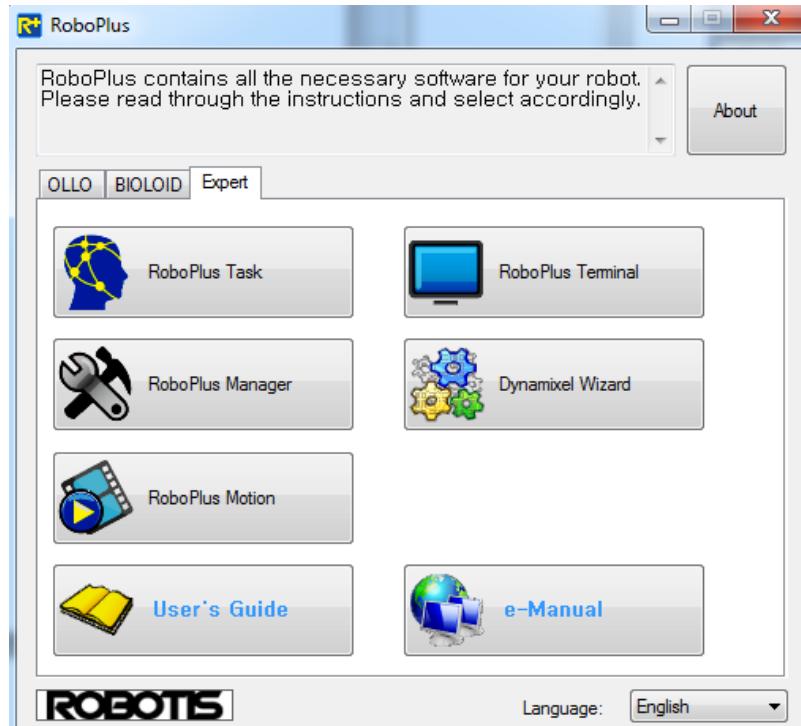
with USB2Dynamixel to set the parameters. Motors can also be communicated by using RoboPlus Terminal with embedded C or RoboPlus Task through CM-510 controller. These motors have to be supplied with 12V if they are communicated by USB2Dynamixel.



**Figure 2.3:** CM-510 controller of Bioloid and its connection inputs for the peripherals.

USB2Dynamixel is a device that provides serial communication between parts of Bioloid and computer. This device converts USB input to inputs of motors or CM-510. The driver for USB2Dynamixel is installed with RoboPlus installation automatically. There is a switch on the device that switches the communication type to TTL input, RS485 or RS232 input which are seen in Figure 2.2. The communication between CM-510 is provided with a serial communication cable. CM-510 controller of Bioloid also can be seen in Figure 2.3 with its connection inputs for its peripherals. If the motors are supplied by controller, to communicate with motors the controller must not send data to motors.

There are two ways of programming CM-510 controller. One is programming with RoboPlus Task and the other is using Embedded C. Both need RoboPlus to be installed on computer. In addition to RoboPlus Atmel Studio has to be installed to programme with Embedded C. Even RoboPlus Task is advantageous in the beginning stage, since RoboPlus Task has limited abilities for computational tasks, Embedded C is more suitable. To give a perspective RoboPlus Task is also explained briefly.



**Figure 2.4:** RoboPlus and its subprogrammes to programme CM-510 and communicate with motors.

## 2.1 RoboPlus

RoboPlus is a programme that is developed by Robotis to programme CM-510 and communicate with motors. RoboPlus has subprogrammes which are shown in Figure 2.4 to provide programming and communication.

RoboPlus Task is a subprogramme to programme Bioloid quickly. This programme provides the user a definite number of commands and parameters. The robot is programmed by using these. Programming is realized choosing commands with mouse in a C kind of programming environment. Assignment, definition, loops, conditions, computations, functions can be realized to use sensor and motor parameters. The code prepared with RoboPlus Task is transformed to a source code which has '.tsk' extension. The code is uploaded to CM-510 to control the robot. Robot needs a firmware to run the Task code. The firmware can be installed by using RoboPlus Manager whose instructions are in [13] in case of a damaged firmware manner. The instructions of RoboPlus Task to programme CM-510 are in 'Software Help' section of [13]. However, the firmware needs some memory to read the source code. Thus, this situation limits the available memory for programming. Even all the parameters

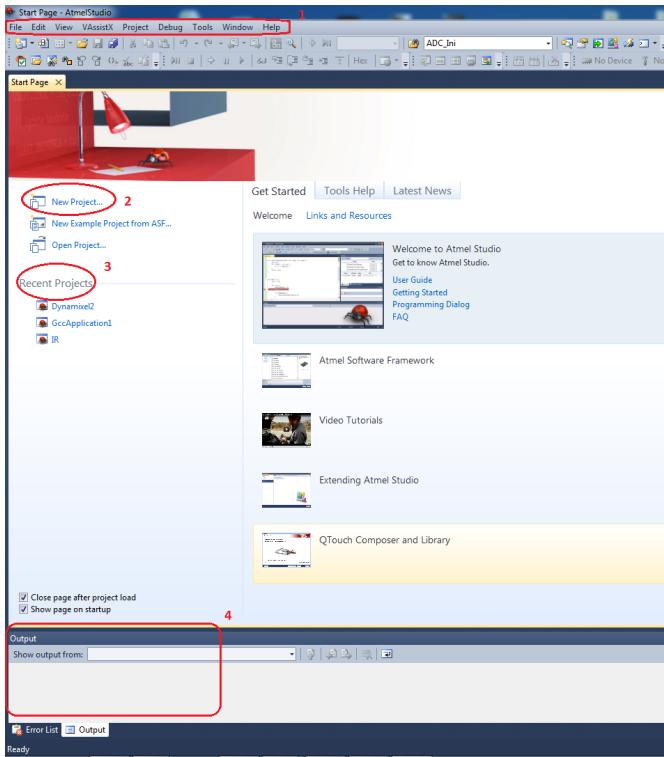
are ready for programming in RoboPlus Task, all calculations are limited to be done in integer domain.

RoboPlus Terminal provides communication with CM-510 if the firmware for RoboPlus Task is not installed on CM-510. This subprogramme is a text based interface of controller. After connection is provided between computer and USB2Dynamixel by clicking "Connect", "Boot Loader" has to be run on CM-510 to communicate with. To run "Boot Loader" while "Alt Gr+3" combination is pressed on keyboard, CM-510 is switched on. By completing this communication with CM-510, the source code can be transmitted by clicking "Transmitting File" command. The "Recieve File" command can also be used to get the values coming from CM-510 to a text file. The instructions to use RoboPlus Terminal take place in [13].

Dynamixel Wizard is also a subprogramme of RoboPlus that communicates with motors to adjust parameters directly. To reach motors using Dynamixel Wizard, motors are connected to USB2Dynamixel.

## 2.2 Programming With Embedded C

Programming with Embedded C provides user to develop its own firmware. Robot runs with a programme coded in C. Loading this programme deletes the other firmware that was uploaded before and because of that RoboPlus Task can not be used while using a source code written in C. CM-510 communicates with computer using RoboPlus Terminal instead of RoboPlus Task. There are libraries of Robotis to programme easily in C. These libraries include the functions to communicate with motors and peripheral devices. So, the robot is programmed with the libraries and using the properties of ATmega2561. Since programming CM-510 means programming ATmega2561 microcontroller, Atmel Studio is used to programme ATmega2561. There are instructions to programme ATmega2561 in User's Guide [13]. Section "Software Help" in [13] includes "Embedded C" section which contains programming help inside. There are also examples that show how to use the parameters and functions. The functions related to motors are in "Dynamixel SDK" topic instead of "Embedded C".



**Figure 2.5:** Atmel Studio 6 interface. 1: Menu, 2:Generates new project, 3: Shows recent projects, 4: Shows results and outputs of operation

Atmel Studio has an interface as given in Figure 2.5. First a new project has to be opened in order to write programme code. The project is generated as "GCC C Executable Project" and while generating a new project ATmega2561 microcontroller has to be chosen. After generating a new project, the programme is written in while(1) loop. Before writing the code, libraries, which can be downloaded as "embedded\_c(cm510\_v1.01).zip" folder from "CM-510/CM-700" topic in User's Guide, has to be identified to Atmel Studio. This is done by adding "include" and "lib" folders of the zip file to the "Toolchain" part of "Properties" which takes place in "Project" that is seen in menu.

The new project includes "io.h" library defaultly. "stdio.h" that is a standard C input output library, "interrupt.h" that is interrupt register library of ATmega2561, "dynamixel.h" that includes motor functions and "serial.h" that defines the serial communication functions can be defined at the beginning of source code to include if they are useful for the project. To achieve a proper coding, input output configuration of microcontroller has to be known and can be achieved from "Hardware Port Map" topic in [13]. After finishing coding, the code is compiled with "Build Solution (F7)" command which is in "Build" part of menu. The "hex" file is created after this process

to be loaded to CM-510 by using "RoboPlus Terminal" through the instructions which are detailed in the related topic of [13]. Following the load process, the code will run on the controller with switching off and on.

### **2.2.1 Functions**

Here the functions that were used mostly are explained briefly. Dynamixel SDK(Software Developement Kit) can be used in order to programme the motors. These functions take place in "Dynamixel SDK" topic of User's Guide.

*dxl\_initialize(devIndex,Baudnum)* function starts serial communication elements. This function has two parameters that the first one defines the port number while the second is related to communication speed. *dxl\_read\_byte(id,address)* and *dxl\_read\_word(id,address)* functions are used to read 1 byte or 2 bytes data respectively. "id" parameter represents for the motor number and "address" stands for indicating the parameter of motor which will be read. *dxl\_write\_byte(id,address,value)* and *dxl\_write\_word(id,address,value)* functions write the parameter value to related address on the motor which is identified by id parameter. They are used to write 1 byte or 2 bytes to motors.

### **2.2.2 An Example Application on Motors**

This application includes the coding style of motors for "Probing Robot" [13] to make the robot move in forward direction.

```
void Forward(int Speed){  
  
    dxl_write_word(1,P_GOAL_SPEED_L,Speed+1024); //in CW direction with Speed  
    velocity  
  
    dxl_write_word(2,P_GOAL_SPEED_L,Speed); //in CCW direction with Speed  
    velocity  
  
    dxl_write_word(3,P_GOAL_SPEED_L,Speed+1024);  
  
    dxl_write_word(4,P_GOAL_SPEED_L,Speed); }  
  
The above code assigns a Speed parameter, which defines the velocity of robot  
in the forward direction, to "Forward" function. The first and third motors are
```

move in clockwise direction while the other two turn in counter clockwise direction. The turning directions are related to robot's assembly. To make the robot move *dxl\_write\_word* function is used. The predefined value of P\_GOAL\_SPEED is 32 which is the address of low byte of speed parameter. The first byte of Speed parameter is written on the address of 32 while the second byte is written on the address of 33 which corresponds to the high byte of speed parameter. Adding 1024 to Speed parameter corresponds to making the tenth bit of parameter one which defines the turning direction. This code can be combined as the one below which makes the robot go in forward direction and after a while stop. This move and stop process goes on permanently. *serial\_initialize()* function defines the speed of serial communication. *sei()* function activates interrupts which is used while communicating with the motors. *\_delay\_ms()* function is a delay function which makes the process wait on that line of code for the input value milliseconds.

```
int main(void) {
    serial_initialize(57600);
    dxl_initialize( 0, DEFAULT_BAUDNUM );
    sei(); // Activation of interrupt
    _delay_ms(300);
    while(1){
        Forward(400);
        _delay_ms(1000);
        Forward(0);
        _delay_ms(300);
    }
    return 0;
}
```

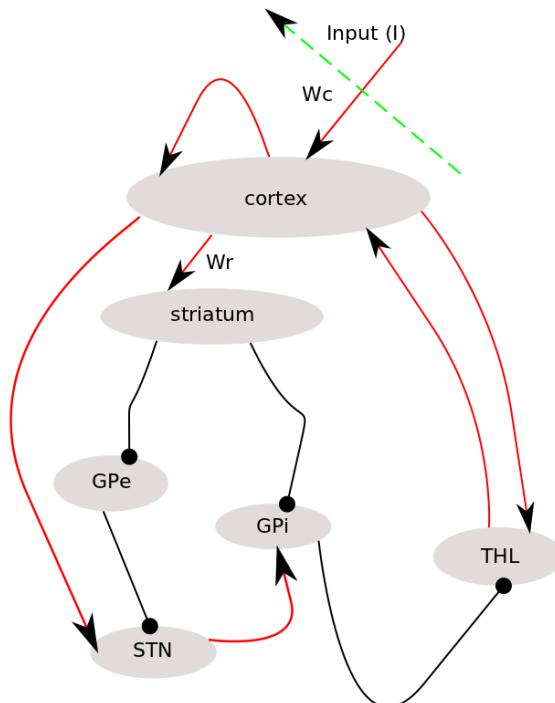
Even programming with embedded C doesn't have an easy user interface, it provides user a flexible structure to programme. It is possible to construct more complex programming structures with embedded C. Despite in the beginning phase of this project RoboPlus Task was used, after noticing that RoboPlus Task has a limited environment, embedded C is utilized.

### 3. IMPLEMENTATING BTC MODEL ON BIOLOID

In order to implement learning to select a proper action in a changing environment to the Bioloid robot, the computational model of Basal Ganglia-Thalamus-Cortex (BTC) for action selection given in [5] is considered. The learning process is implemented in such a way that by effecting the dynamics of the BTC model, Bioloid robot is led to choose appropriate action; this is accomplished by reinforcement learning.

#### 3.1 Basal Ganglia-Thalamus-Cortex(BTC) Model

The BTC circuit used for action selection is the one in [6] and its sheme is given in Figure 3.1 . Basal ganglia (BG) along with the related parts of Cortex (Ctx) and Thalamus (Thl) take part in decision making tasks [1, 5, 6].



**Figure 3.1:** Block diagram of Basal ganglia circuit: This diagram shows the relations, which are defined as inhibitory or excitatory bounds between substructures of Basal ganglia, cortex and thalamus.

Once the cortex is activated by sensory inputs which is denoted by  $I$  in the model, these sensory inputs are evaluated through reinforcement learning and their significance in

the environmental context is determined through reinforcement learning by modifying the weight of each sensory input. These weights are denoted by  $W_c$  and the weighted sensory inputs are denoted by  $S$ . The relation between sensory inputs  $I$  and  $S$  is linear and is given by Equations 3.1 .

$$S_{11} = W_{c11}I_{11} + W_{c12}I_{21} + W_{c13}I_{31} \quad (3.1)$$

$$S_{21} = W_{c21}I_{11} + W_{c22}I_{21} + W_{c23}I_{31}$$

$$S_{31} = W_{c31}I_{11} + W_{c32}I_{21} + W_{c33}I_{31}$$

As, three different sensor information obtained from two different sensors; distance sensor and infrared sensors are used,  $I$  and  $S$  are vectors of dimesion  $3 \times 1$  and  $W_c$  is a matrix of  $3 \times 3$ . The substructures of BG considered in the model are striatum (Str), globus pallidus external (GPe), subthalamic nucleus (Stn), globus pallidus internal (GPi). There are three different actions to choose for the task considered, so the dimension of each variable considered in BTC is  $3 \times 1$ , and actions are coded binary. The interrelation between these substructures and Ctx, Thl are modeled with difference equations as the process is a dynamical one and represented by Equations 3.2.

$$Ctx(k+1) = f(\lambda Ctx(k) + Thl(k) + S(k)) \quad (3.2)$$

$$Str(k+1) = Wrf(Ctx(k))$$

$$GPe(k+1) = f(-Str(k))$$

$$Stn(k+1) = f(Ctx(k) - GPe(k))$$

$$GPi(k+1) = Wdf(Stn(k) - Str(k))$$

$$Thl(k+1) = f(Ctx(k) - GPi(k))$$

$\lambda$  coefficient denotes the recurrent behavior within the cortex.  $W_r$ , in these equations is a  $3 \times 3$  dimensional diagonal matrix and represents the effect of dopamine on action selection. In this study  $W_r$  is taken constant.  $W_d$  is a  $3 \times 3$  dimensional matrix whose elements have the same value and it represents the diffusive influence of Stn on GPi. The function  $f(\cdot)$  is a tangent hyperbolic function, and it is used to model the mean activity of a population of neurons considered and given by Equation 3.3.

$$f(x) = 0.5(\tanh(3(x - 0.45)) + 1) \quad (3.3)$$

The Equations 3.2 are related to the BTC circuit which is responsible for decision making. The result of action selection is determined by  $Ctx$  values. Once the variables of BTC model converge to an equilibrium point [5], the  $Ctx$  value determines which action is chosen. The learning part of the model consists of temporal difference learning (TDL) rules which is a method of reinforcement learning [15]. It is claimed to be related to reward based learning in basal ganglia [14]. The learning is accomplished through updating the weight of sensory inputs, namely,  $W_c$ . Only, diagonal elements of  $W_c$  matrix are updated, the others are kept same at a value close to zero. The detailed structure is available in [5, 6]. The Equations 3.4 to 3.7 define the TDL rules. In these Equations  $v$  is the value function and  $r$  is the reward obtained as a result of action selected. First, the value of sensory input is determined according to TDL with Equation 3.4 using value function  $v$ , then expectation error  $\delta_c$  is calculated as given in Equation 5. This expectation error is used to modify the diagonal of weight matrix and value function as given in Equations 3.6 and 3.7.

$$V(k+1) = Wv(k)S(k) \quad (3.4)$$

$$\delta_c(k+1) = r_c + \mu V(k+1) - V(k) \quad (3.5)$$

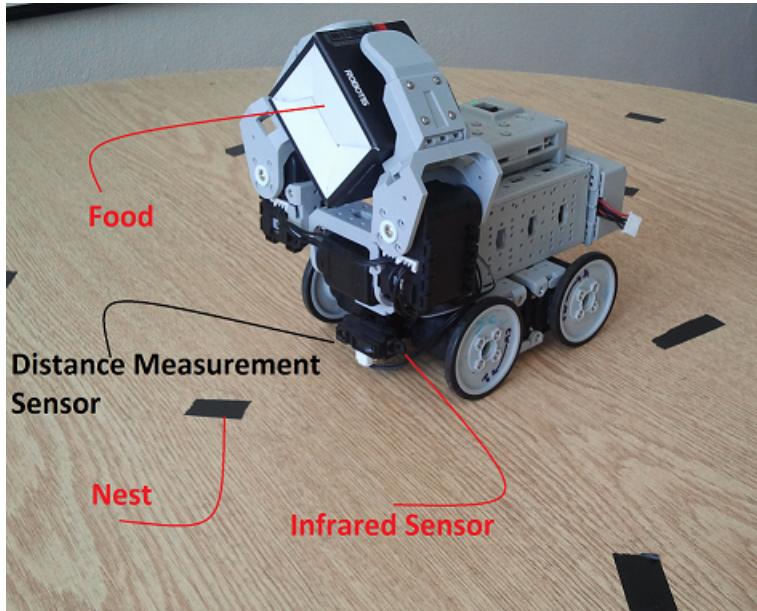
$$Wv(k+1) = Wv(k) + \eta_c \delta_c(k+1) S(k) \quad (3.6)$$

$$Wc(k+1) = Wc(k) + \eta_c \delta_c(k+1) Ctx(k) S(k) \quad (3.7)$$

With the updated values of  $W_c$ , the behavior of BTC is determined. This behavior is kept, if due to it a reward is obtained, and is expected to be changed if no reward is obtained. In the next section how this model is implemented on Bioloid robot will be explained.

### 3.2 Bioloid Robot Implementation

The Bioloid Robot is an educational robot that can be assembled in various forms. The robot is able to get data from environment with its sensors and respond to these data with motor actions after processing them [13]. The task considered is foraging task of a rat, thus Bioloid robot is expected to mimic the behavior of a rat. So, Bioloid must have abilities as noticing an object or a nest, taking the proper object, lifting it and carrying it to a proper place, which corresponds to nest. Hence, Bioloid is assembled as "Probing Robot" example of Bioloid Premium Kit as given in the Robotis e-Manuel [13].



**Figure 3.2:** Environmental setup for foraging task. An educational robot assembled to model rat and moves on a platform that has a different reflection coefficient from nest.

The sensors and movements of Bioloid are set parallel with a rat's behaviour and its interaction with the environment. There are two sensors on Bioloid suitable for interaction with the environment, these are infrared and distance measurement sensors.

Robot uses its distance measurement sensor (DMS) to find an object or food and infrared sensor (IR) to find the nest. In this work, the food is represented as a box whose dimensions are 5x5x5 cm and the nest is represented as a black band whose reflection coefficient is different from ground the experiment is set up. Robot uses its gripper to grab the food and places to the nest in the same way. To provide an easy movement of robot, robot is chosen to have wheels instead of foot (Figure 3.2). The environment to programme Bioloid consist of RoboPlus and Atmel Studio 6. These tools are useful to programme Bioloid with embedded C. In addition to these programmes, Dynamixel Wizard tool of RoboPlus is used to communicate with and programme the motors of Bioloid externally [13]. Furthermore, Bioloid Robot has a controller CM-510 on itself which has an ATmega2561 microcontroller to control and communicate with all peripherals and Atmel Studio 6 is used to programme ATmega2561. RoboPlus Terminal is advantageous in loading hex file to microcontroller and visualising the programme output on computer. So, with these facilities, sensor information is carried to the motors through embedded C to control

motion of them and this control in this graduation project is done by employing the BTC on Bioloid.

### **3.2.1 Implementing the Foraging Task on Bioloid Robot**

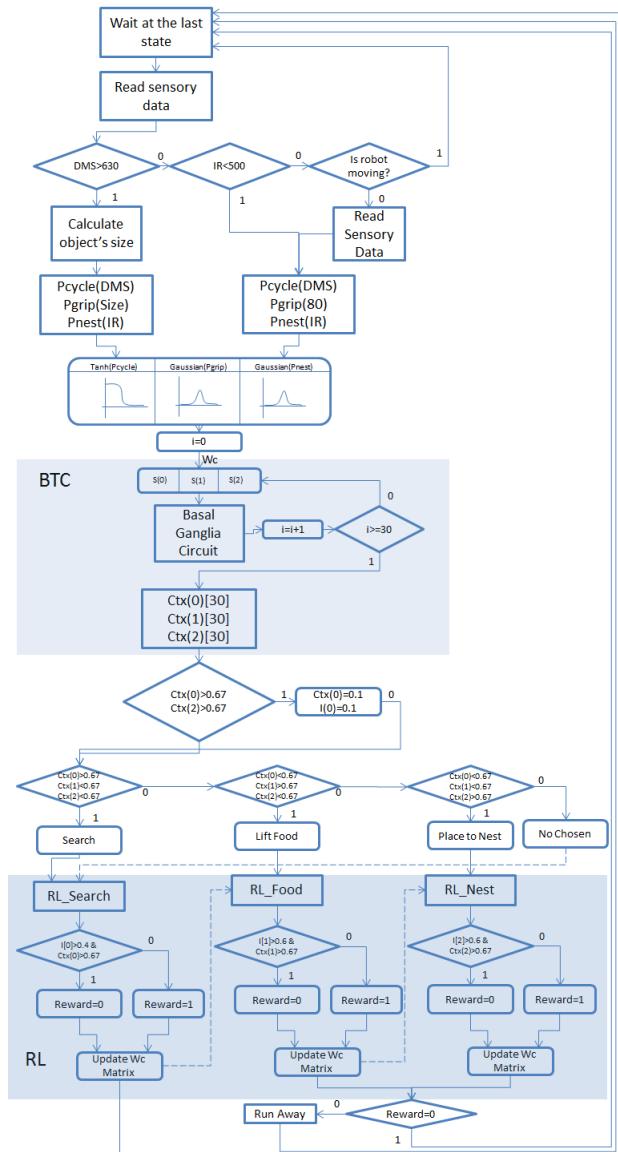
Studies have shown that a rat stands still until it gets used to a new environment. After rat is accustomed to the new environment, feeling of hunger becomes more dominant than feeling of fear [3]. Thus, the rat begins foraging. Rat can distinguish the possible food by smelling or touching it and has to find a secure place to hide its food. Therefore, it seeks a nest. So, Bioloid robot should decide between actions of searching, recognizing food and placing food to nest in order to represent foraging behaviour of a rat. Here, all these decisions will be given using the BTC model and learning of these actions are accomplished by reinforcement learning rule as summarized in Section II. Thus, once a decision is made, Bioloid evaluates its actions and learns searching, recognizing food and nest by using reinforcement learning.

Firstly, robot gets accustomed to the environment and learns that it has to search for food. During the on going searching process, robot has to recognize the food. To do that, it tries to lift a lot of boxes and identifies the ones that it can lift as possible food. Once, it learns that the little box means food, and the big box that it cannot lift means the object to be avoided, robot needs to find a nest. Hence, it has to choose the most secure one among possible nests. In this work the most secure nest is defined with a black band. Thus, Bioloid comes across the black band many times before recognizes it as the nest. The Bioloid robot and its environment is given in Figure 3.2.

### **3.2.2 Programming Bioloid**

The block diagram of operation of robot can be seen in Figure 3. There are three possible situations which urge robot to evaluate sensor values and sensors return analog values that correspond to discrete values between 0 and 1023. First two of these possible situations are detecting an object at the front of robot and a possible nest under the robot. The third situation is being motionless, as standing motionless is an unwanted situation, this triggers to get new sensory data and observe the environment. At the beginning, Bioloid waits motionless, this phase represents observing environment. In this situation robot gets data from its sensors and with this sensor information

reinforcement learning updates the first  $W_c$  value till it reaches a value that triggers robot searching. Robot searches till it gets across with an object, if there is an object in front of the robot, DMS returns 630 at an appropriate distance. The detecting of an object as food or not food depends on the size of an object, thus an object with value of 80 or less is determined as food, while an object with value greater than 80 is considered as an obstacle to avoid. To detect nest, information from IR sensor is considered. IR returns 700 in normal ground, if this value decreases under 500 then there is a nest possibility. These three environmental sensor data are used to form



**Figure 3.3:** Block diagram of the operation in Bioloid. Operation begins with reading sensory data. The data used to make decision in basal ganglia circuit as an input and evaluate the learning rules.

sensory information denoted by  $I$ . As the BTC system values are in the interval of  $[0,1]$  due to sigmoid function, these sensor data should be scaled. The sensor value

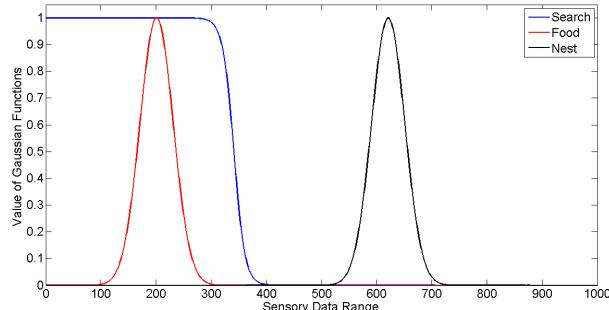
for searching obtained with DMS data is scaled to [0,1] interval by tangent hyperbolic function given in Figure 4 as blue line. The second sensor value is related to lifting a food. Since the lifting food behaviour is related to the size of an object, the value is calculated by using DMS. The size information is also scaled into the [0,1] interval with Gaussian function given in Figure 4, as red line. The third value is related to placing the food to the nest. Since for detecting nest IR data is used this IR data is scaled to [0,1] interval with Gaussian function given in Figure 3.4, as black line.

After scaling sensor data, the calculated input values are fed to the BTC model which is the decision making block. To achieve a valid output from BTC model, output values have to set to a stable equilibrium point so the discrete time system outputs are calculated at the end of 30 iteration steps. Basal ganglia circuit generates output to choose a behaviour. If just one of the outputs value is over 0.67, the behaviour that is represented by the chosen output is realized. Learning an appropriate behaviour is realized with reinforcement learning rules after related decision is made. If there is no decision, then the coefficients of all three behaviour are updated with RL rules. After updating  $W_c$ , the process turns to the beginning and a new sensor data is read.

### **3.2.3 Scaling and Normalizing**

In the previous section, the need for the scaling is mentioned, here first normalization of raw data will be explained in detail and then how scaled values are used to control Bioloid robot will be given. Though the sensor data are scaled to [0,1] interval with Gaussian and tangent hyperbolic functions before this scaling normalization is done. The data of DMS which is used to determine the searching behaviour is up to 800 in practice even the highest value that the DMS data can get is 1023 in theory. Thus, normalization is made by dividing the sensory output to 1000. Hence, the normalization function is  $DMS/1000$ . For lifting food process, the size information of an object is used. This is also calculated with DMS. The value is between 65-70 for the little box and 80 for a big object. The normalization is made by dividing the size value to 85 and extracting this from 1. Because the values over 80 is meaningless for this study. Hence, the normalization function for lifting food behaviour is  $1-(Size/85)$ . Placing food to the nest behaviour is determined by the information on the IR sensor output. Even the output range is between 0 to 1023, the meaningful range is between 0

to 800. Hence, the normalization is made by the  $1-(IR/1000)$  function. The normalized sensory outputs are then scaled with Gauss and tangent hyperbolic functions. The normalized sensory data is suppressed in the meaningless areas for the robot and left same in the meaningful area. This operation can be associated with an object being more familiar to a rat. Therefore, robot's coherence between BTC circuit and real behaviour is provided by scaling with these functions. The front side of robot has to be clear in order to realize the searching task. A clear front side is represented by 0.3 or below considering the normalized values. Hence, to realize this, a tangent hyperbolic function is used to give the same value for 0.3 or below and to decrease values more than 0.3. The meaningful situation for lifting food task is that object is little enough for robot to lift. This value is normalized 0.2 for lifting task. To leave same 0.2 value and around a Gaussian function is used whose mean value is 0.2. The same situation is valid also for placing food to the nest task. The significant values are the normalized ones which are around 0.62. Hence, a Gaussian function with a mean value of 0.62 is used. The Gaussian functions that are used can be followed in Figure 3.4.



**Figure 3.4:** Scaling functions of the inputs: These functions take the useful inside the data of environmental sensory inputs.

### 3.2.4 Basal Ganglia Circuit

BTC model, which is mentioned in the second part, is used to determine the robot responses to the sensory inputs. Once the scaled sensor data which corresponds to  $I$  in the BCT circuit model is obtained,  $S$  vector is determined by multiplying  $I$  with  $W_c$  matrix.  $W_c$  provides weightening of sensor inputs, which has been differentiated from each other by using scaling functions, to provide the compatibility of the robot to the environment . The input of BCT circuit which are elements of  $S$  vector are applied to cortex, striatum, GPe, Stn, Gpi respectively. At last, the information which is the result of relations between these basal ganglia parts is transferred to thalamus.

The new calculated results are the inputs of the next iteration. It is a must for a BTC model to make the right action selection that the outputs have to settle to a stable equilibrium point. Hence, the iterations must continue until the difference of the last two outputs is infinitesimal. However, since the robot works in real time and increasing load of calculations causes delay, an assumption that the output values can converge in 30 iteration, has been made. Thus, the outputs of cortex in the last iteration are the outputs of BCT circuit also. To conclude action selection, an if condition is used. To select an action, only one of the cortex output has to be greater than 0.67 while the others are smaller than 0.67. Otherwise, at the output of BCT more than one action (or behaviour) is selected and this is meaningless for robot application in this study. If none of the cortex outputs are greater than 0.67, that means none of the actions are selected. When only one action is selected, the related  $W_c$  coefficients are updated. If none of the actions are selected, then all diagonal elements of  $W_c$  are updated. These updates are made according to the reinforcement rules given with Equation 3.4 to 3.7. While trying to work robot under these conditions, a situation is noticed. When the robot meets a possible nest position, there will not be an object in the front of the robot. Therefore, robot will try to select two actions at the same time, one being to leave food to the nest and the other is continuing to search. The values that are related to the search behaviour are suppressed when this situation occurs.

### **3.2.5 Reinforcement Learning**

Updating a diagonal element of  $W_c$  matrix is related to reward. Reward is denoted by  $r$  in the equation 3.5. If the robot has chosen the right action then the value of  $r$  is determined zero otherwise one. Though this is not the conventional way of assigning reward, for the robot application this worked out as explained in the sequel. If robot does the right action, expectation error given in Equation 3.5 decreases. Contrarily, when a wrong decision is made, the expectation error function increases and this rises the related element of  $W_c$  matrix. Making wrong decision indicates that the value of related element in  $W_c$  matrix is high enough to choose that action. Thus, the expectation error increases the value of related coefficient. Giving reward to the robot is determined by evaluating the inputs and outputs of BCT circuit. When inputs are high enough to select an action and cortex outputs select the action than reward is given to the robot

by making the value of  $r$  zero. This means decreasing expectation error. When inputs are high enough to select an action, but cortex outputs do not respond to this,  $r$  value is set to one making the expectation error high. When a wrong action is selected or none of the actions is selected, robot gets the run away command, because, there is a possibility of an object standing in front of it. Thus, robot has to run away from the object not to collide. After all the updating operations, robot turns to the beginning and realizes the last selected action.

## 4. RESULTS

The robot is expected to learn to make right decisions on searching, recognizing the food and the nest. These actions together complete the three phases of foraging task. The meaning of being successful on learning for all three phases of foraging task is to behave similarly in a changing environment without an outer affect.

After programming the robot, eight trials for training are completed. In all these training trials, only the diagonals of  $W_c$  matrix represents the ability of perceiving the environment are changed by RL. The duration of learning process depends on the initial values of diagonals. During these eight trials, the initial values of diagonals are kept as  $W_{c11}=0.1201$ ,  $W_{c22}=0.0875$  and  $W_{c33}=0.3764$  which are determined randomly. The robot accomplished learning on all trials and is able to make the right decisions. As seen in Table 4.1, the robot learned to decide to begin the search task in an average of four seconds. After beginning to search, the robot is encountered with the possible foods and tried to recognize them. The number of trials is in average 6.4 times. In the last one of the trials, it was able to decide to lift the food. After lifting the food, the robot tried to recognize the nest and this trial took an average of 4.25 times. At last, it decided to set the food to the nest. In all eight training trials, the robot achieved the goal and completed the three phases.

**Table 4.1:** Test results of learning durations for three actions.

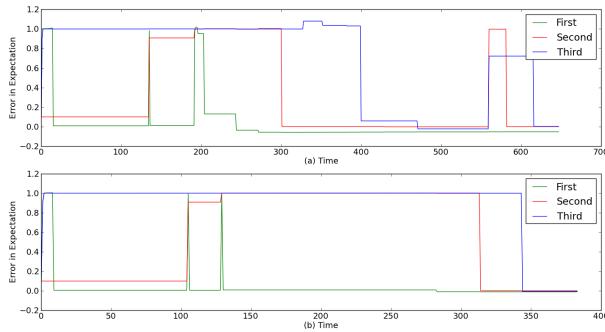
A. Experiments			
	Search	Pick up	Deposit
<b>Average</b>	4.3sn	6.4 trial	4.25 trial
<b>Standard Deviation</b>	0.29sn	2 trial	1.4 trial

This training process can be followed from the change in expectation error, too. Expectation error gives the difference between the robot's expectation on the response of environment to its action and the real response of environment to the action. The expectation error is evaluated with the equation 3.5. The responses to the actions are evaluated with the value function which is given by equation 3.6. As seen in Figure 4.1a, the expectation error of searching task is decreasing to a value close to zero, after

**Table 4.2:** Initial values of parameters on learning operations.

Test	Step Count	Wc11	Wc22	Wc33	Wr11	Wr22	Wr33
Test1	1428	0.1201	0.0875	0.3764	0.1	0.2641	0.1
Test2	3408	0.1001	0.0675	0.2243	0.1	0.2641	0.1
Test3	2994	0.1201	0.0875	0.3764	0.9	0.2641	0.1
Test4	1416	0.1201	0.0875	0.3764	0.3	0.9000	0.1

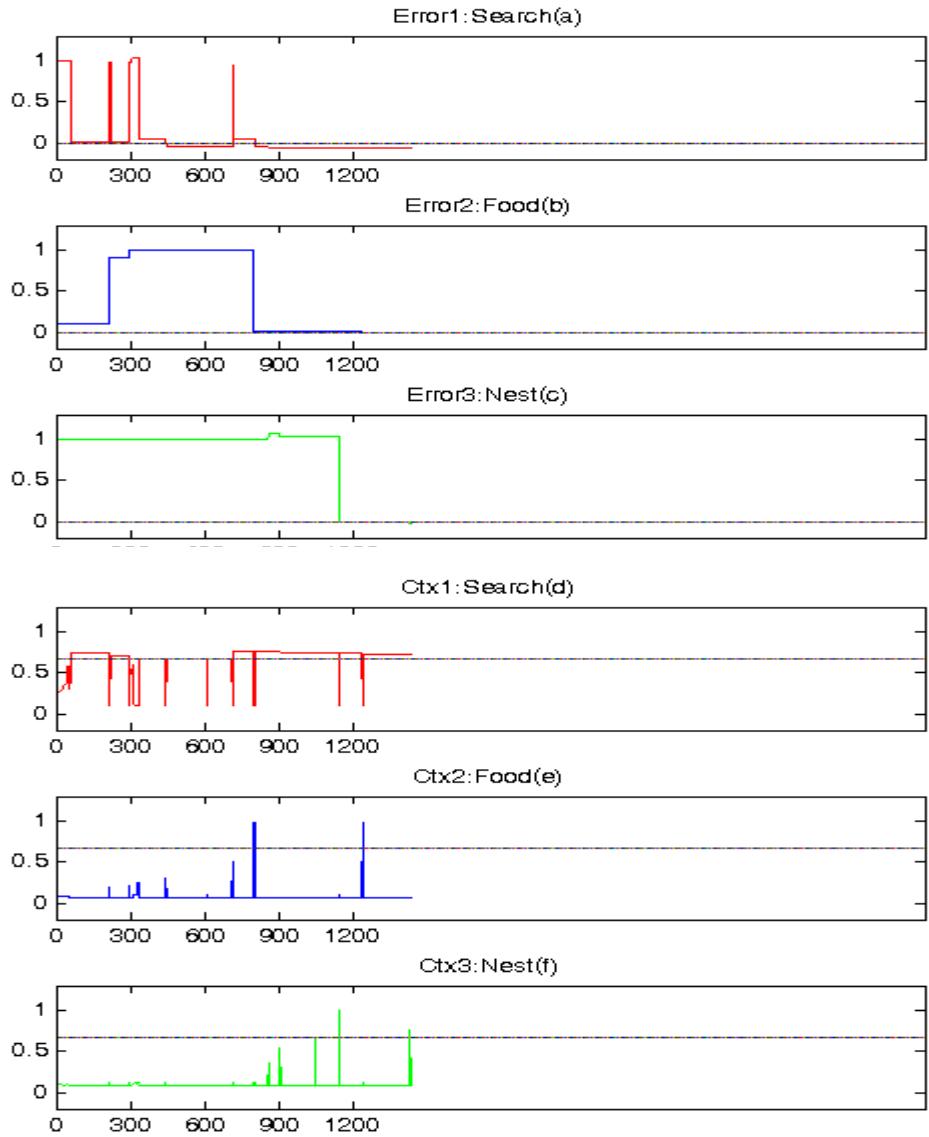
learning searching task as all decreases to zero at the end of trial. Since the initial value of error is taken 0.1, increasing to 1 means that the trials on learning begins. Like the searching task, the expectation error for decision making on lifting food task decreases to a zero after robot recognizes the food. At last, the error related to recognizing the nest decreases to zero that means learning is successful.



**Figure 4.1:** Expectation error which becomes zero at the end of learning.

Although the learning of process of the nest begins after learning the food, the error related to the nest increases to one at the beginning in Figure 4.1b. This is a situation related to sensors which produced an incorrect data at that moment. The wrong data, which comes from the sensory inputs, make the robot to behave as there is a possible nest, so the robot starts learning nest also. Learning is possible while the decision making process goes on. In Figure 4.1a a different food, which has slightly different size from the prior one, is presented to the robot at the 550th step. Even the robot has learned the prior food, it didnot lift the last one. After one encountering, the learning process of the food with different size is also accomplished and the robot began to lift the food. This means that the robot is able to adapt to a change in the environment.

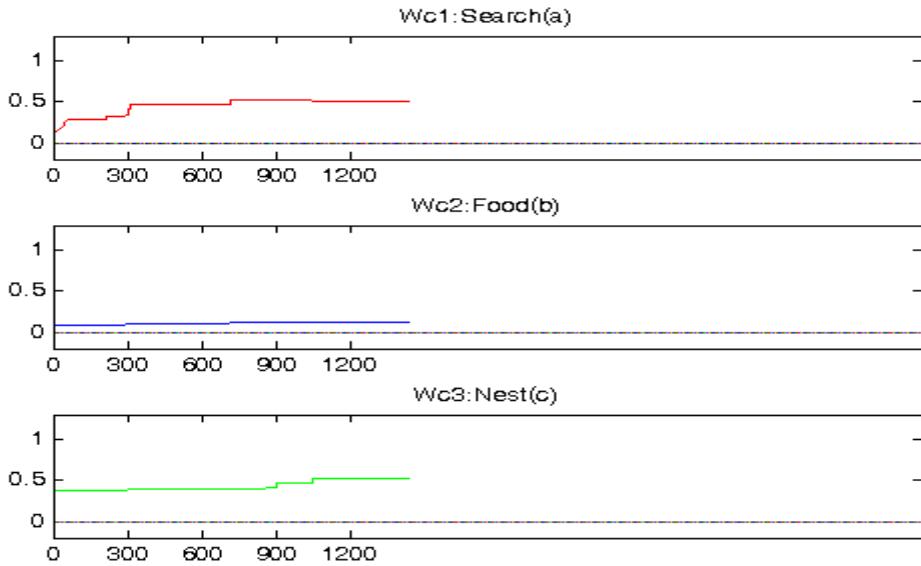
The learning duration and decision making depends of initial values of diagonals of Wc and Wr. Despite the robot learns to choose one action, it may not be able to select the right action because of Wr which models dopamine effect on action selection in basal ganglia [6]. To observe the effect of Wc on duration and Wr on action selection,



**Figure 4.2:** Test1:(a),(b),(c) shows expectation errors and (d),(e),(f) shows variations on diagonal's Wc on the learning of search, food and nest respectively.

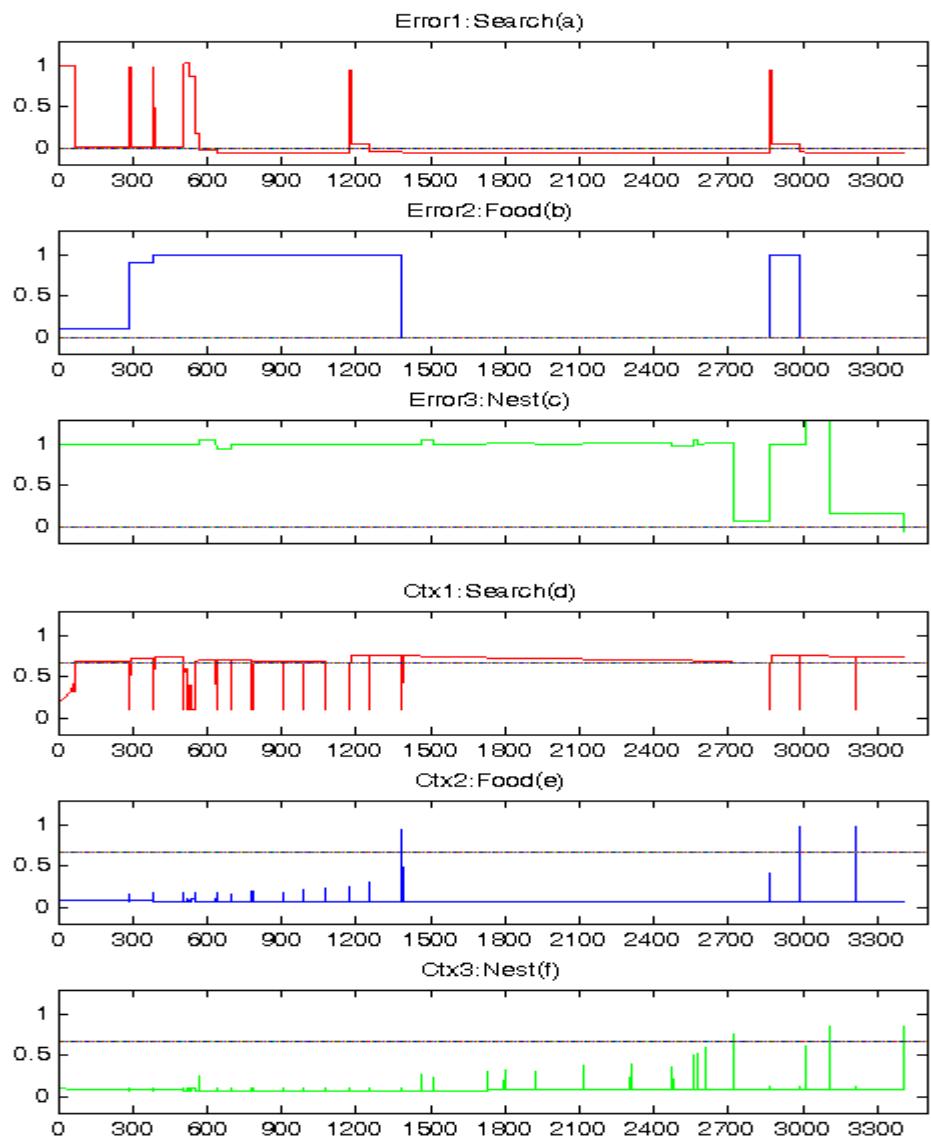
tests are repeated with different values of Wc and Wr. Four of the tests' results are shown here. The changed initial values of Wc and Wr are given in Table 4.2 according to the tests. The first one is control test and the values are the default ones. Test2 shows the effect of different initial values of Wc on duration. Test3 and Test4 show the effect of dopamine on action selection process. The four processes finished in different step numbers. However, their graphs have been given here for the same interval to make the comparison possible.

In Figure 4.2 there are graphs of expectation error and cortex outputs. The first and fourth are related to search task, the second and fifth are related to recognizing food

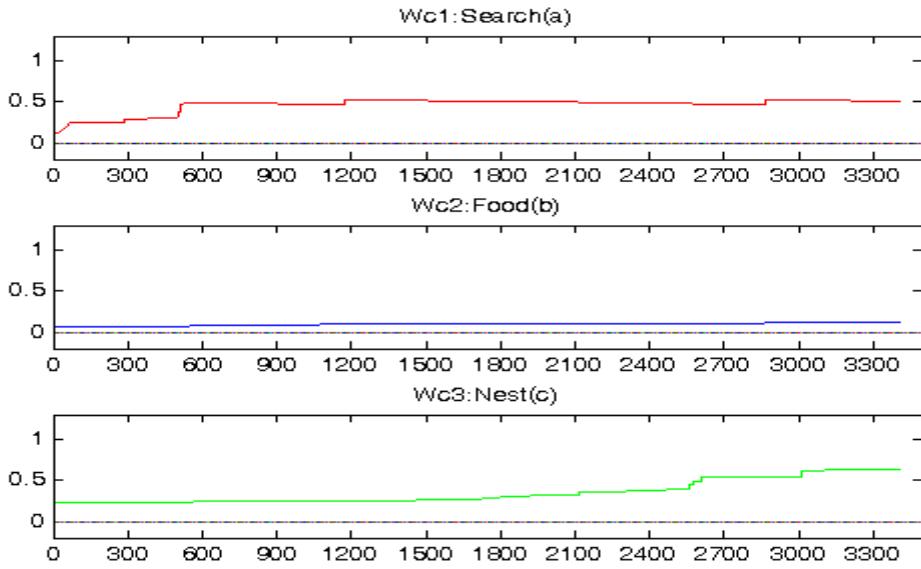


**Figure 4.3:** Test1:(a),(b),(c) shows cortex outputs of BTC which shows whether the robot choose an action.

and the third and sixth are related to recognizing nest. The change of expectation error gives information about learning process while cortex outputs give the information of selected action. Zero expectation error means learning has accomplished. Also, when the value of cortex is above 0.67 it means that the action is selected. In the first test, error is one at the beginning. The learning process is accomplished while the error decreases to zero. At the same time in Figure 4.2(d-e-f) it is seen that the first cortex output is chosen meanwhile search task is also selected. As seen in Figure 4.2(a) robot makes wrong decisions for search task which can be followed by the increases to one in error. However, through learning process the error falls down approximately to zero. As it is seen in Figure 4.2(d) the search task is selected generally. In a step time close to 300 learning begins for food. Through learning process, the second cortex output tries to select the lifting food action, but cannot make decisions until nearly 800th step which corresponds to the end of learning for food (Figure 4.2(e)). The same process is valid for nest. After 800th step in Figure 4.2(f) robot tries to select to set the food to the nest. At approximately step size of 1200 robot is able to make the right decision. On a natural foraging task, only one action can be selected at a time. So, a cortex output of one action decreases to below 0.67 while the other rises up to 0.67. The values of diagonals of  $W_c$  matrix can be seen in Figure 4.3. Through learning process all values rises. A true decision makes the values decrease a little.

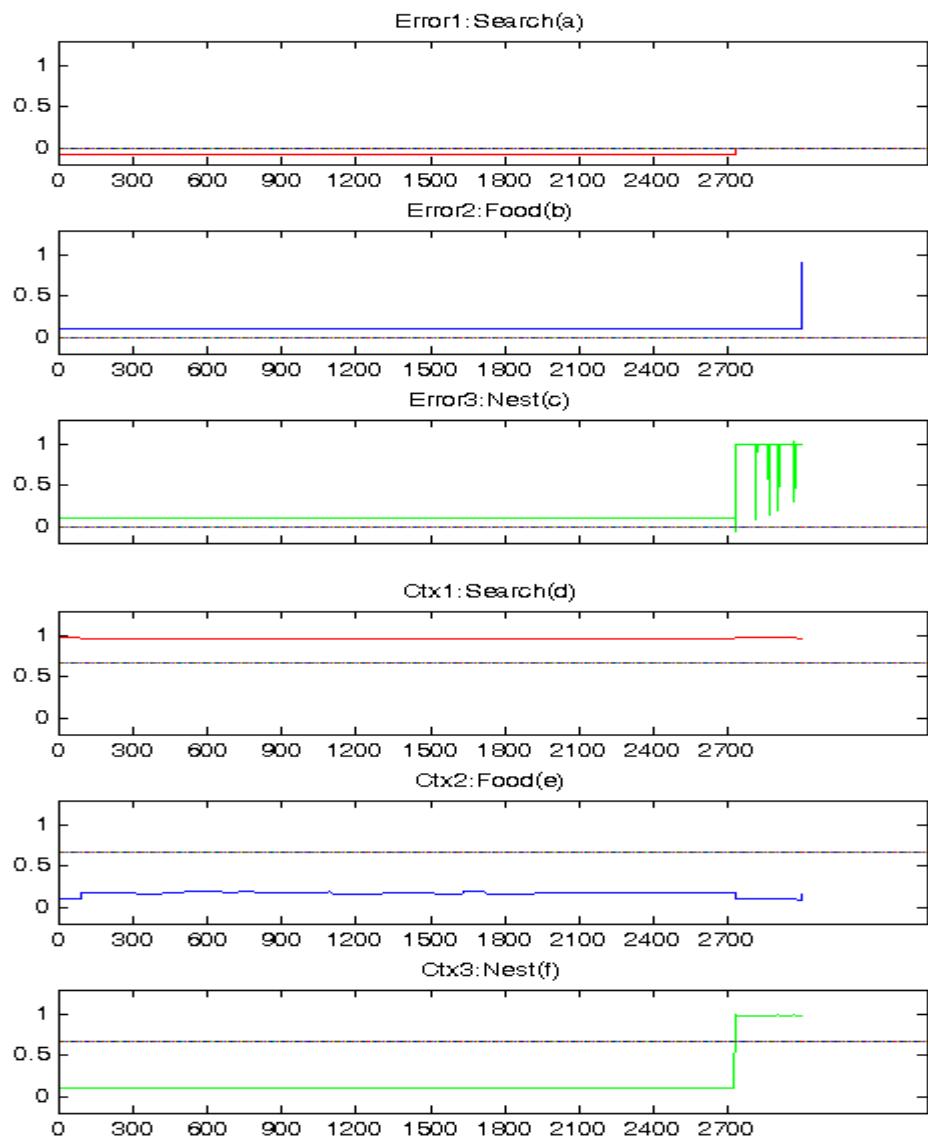


**Figure 4.4:** Test2:(a),(b),(c) shows expectation errors and (d),(e),(f) shows variations on diagonal's  $W_c$  on the learning of search, food and nest respectively.

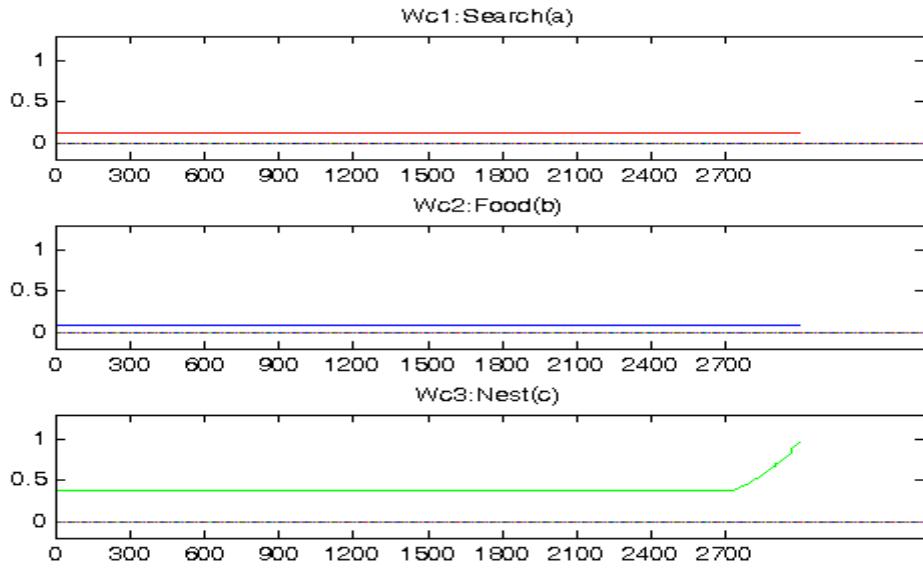


**Figure 4.5:** Test2:(a),(b),(c) shows cortex outputs of BTC which shows whether the robot choose an action.

In Test2, only the initial values of  $W_c$  is changed. The initial values are decreased to investigate the role of initial conditions on duration of learning process(Table4.2). It is expected that the duration of learning will increase. Duration of learning search is almost same with the first test. Test2 has also more steps then Test1 because learning processes of food and nest take much more time. In Test1 the operation of learning food is completed at approximately 600 steps which is seen in Figure 4.2(e) as the pulses that are lower than 0.67. However, in Test2 the robot takes 1200 steps to learn as seen in Figure 4.4(e). Concerning the recognizing nest operation of the robot, it takes approximately 300 steps to learn while in Test2 it takes about 1100 steps. Looking at the Figure 4.5  $W_c$  of Test2 has nearly same values as the values in Test1. Since, the tests have the same structure, being close of the last values of  $W_c$  matrixes is expected. Therefore, changing the initial values of  $W_c$  for different operations changes the duration of task. In Test3 and Test4 the effect of dopamine on learning process is investigated. The dopamine is represented by diagonals of  $W_r$  matrix. Increasing the values of  $W_r$  means rising dopamine. In Test3 dopamine value of search task is rised to 0.9 from 0.1. As it is seen in Figure 4.6(a) error value of search task is close to zero. However, error value of recognizing food task is 0.1 because the RL never works for learning food. The situation is same for detecting nest until the 2700th step after then RL begins for recognizing nest. Looking at cortex values (Figure 4.6), it is seen that only search task is chosen until the 2700th step. At that point the robot encounteres



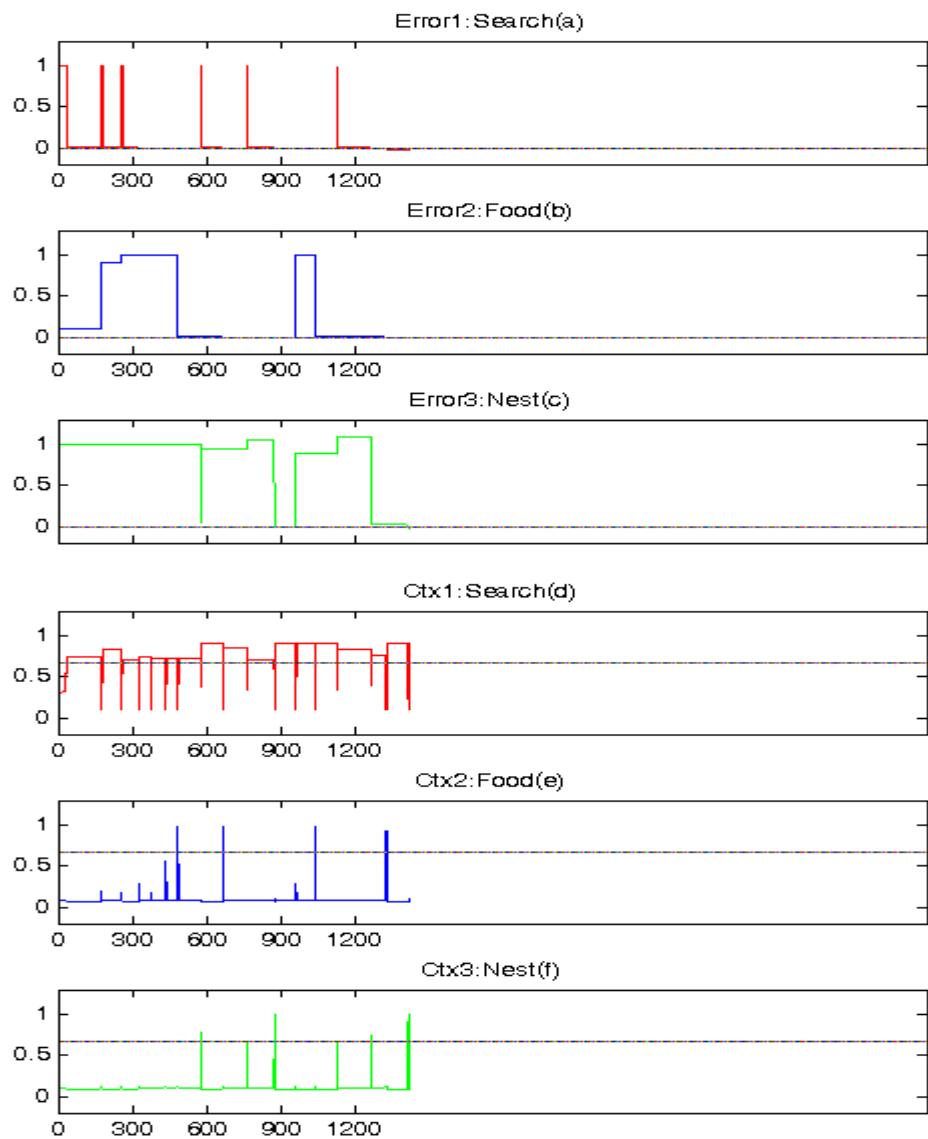
**Figure 4.6:** Test3:(a),(b),(c) shows expectation errors and (d),(e),(f) shows variations on diagonal's  $W_c$  on the learning of search, food and nest respectively.



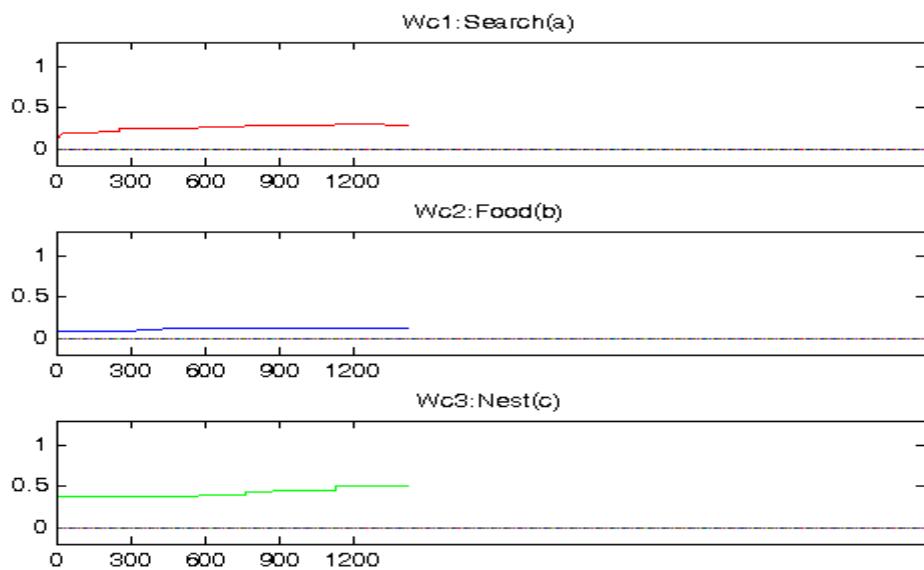
**Figure 4.7:** Test3:(a),(b),(c) shows cortex outputs of BTC which shows whether the robot choose an action.

with the nest, and begins to learn. Even error value of nest high, robot tries to choose depositing food action. The robot never give up the search action and tries to choose both of searching and placing food. Thus, it cannot decide one of them. Looking at the Wc values, the values related to search and food donot change while the value of recognizing nest rises after encountering with the nest. Even though the Wc value of search task doesnot change, robot can choose the action from the beginning. Changing values of Wr affects all learning process and makes the robot always choose the first action. This situation causes robot not to work on tasks properly.

In Test4 two initial values of Wr are changed while the last one is constant. It is seen at Figure 4.8 that robot is able to realize a normal process. It learns to search for food, lift the food and set to the nest respectively. It chooses only one action at a time and its Wc values, which are given in Figure 4.9, are close to ones at the first test(Figure 4.3). Thus, even with the changed initial Wr values, robot is able to make right decisions and learn them. It is seen in Figure 4.8(e) that robot can choose to lift the food at the 1300th step after it learnt to do that about at the 1150th step. Changing Wr and Wc means mathematically changing the place of equilibrium points of dynamical system on weight space. Since the equations of RL and decision making, which are the equations from 3.1 to 3.7, are difference equations of a dynamical nonlinear deterministic system, the bifurcation analysis for equilibrium points is investigated in [6]. If the robot learns



**Figure 4.8:** Test4:(a),(b),(c) shows expectation errors and (d),(e),(f) shows variations on diagonal's  $W_c$  on the learning of search, food and nest respectively.



**Figure 4.9:** Test4:(a),(b),(c) shows cortex outputs of BTC which shows whether the robot choose an action.

then it means that the system can converge to the right equilibrium point according to the inputs of system which are the sensory data.

## **5. CONCLUSIONS AND RECOMMENDATIONS**

In this project, the BTC model which is a decision making model of brain circuit including cortex, basal ganglia and thalamus and RL model, which is used for robot to learn, are coded in embedded C for a real time application of decision making task. The proposed model for the basal ganglia circuit in [5] is implemented on a real mobile robot with embedded C programming to show the applicability of this approach. The BTC model's implementation had been already discussed on the simulation platform using Khepera II [12]. In this graduation study a simple robot, Bioloid, is used to show the applicability of the proposed model.

It has been emphasized that the computational model and learning method proposed in [5] are used for realizing the foraging task on a mobile robot. Instead of focusing on the model and learning method, the proposed method is realized on Bioloid robot with a real-time application. It is shown that even with a simple robot, the computational model is capable of realizing high-level learning tasks. Thus, here the main goal is to show the proposed model's applicability rather than robot's abilities. Since the learning process considered is reward-modulated learning and indicates that the basal ganglia particularly striatum [16] is responsible for accomplishing the action selection task, the method also embraces the biological realism. Based on this, one of the main goal is testing the biological substructures of brain to understand how they operate and how diseases show up. Thus, testing these kind of models are both advantageous concerning time and requirements in comparison to human tests. This graduation project is presented in the 11. National Neuroscience Congress [17], to improve it further with feedback from neuroscience group.

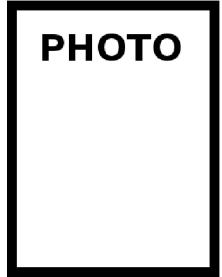
At a later stage on this graduation process,  $W_r$ , which represents dopamine effect on decision making task, can be evaluated by RL to find an appropriate decision space

automatically or to see how it behaves on a changing environment. Also, the BTC model can be changed to a more biologically realistic with a more realistic task.

## REFERENCES

- [1] **Prescott, T.J., Gonzales, F.M.M., Gurney, K., Humphries, M.D. and Redgrave, P.**(2006). A robot model of the basal ganglia: Behaviour and intrinsic processing,*Neural Networks*,**19**, pp.31-61.
- [2] **Montes-Gonzalez, F., , F.M.M. and Contreras, C.M.**(2013). The Evolution of Motivated and Modulated Robot Selection, *Int J Adv Robotic Sy*, **Vol.10**, 125.
- [3] **Gurney, K.N.**(2009). Computational Models in Neuroscience, From Membranes to Robots, *Computational Modeling in Behavioral Neuroscience*, pp.107-136.
- [4] **Bullock, D., Tan, C.O. and John, Y.J.**(2009). Computational perspectives on forebrain microcircuits implicated in reinforcement learning, action selection, and cognitive control, *Neural Network*,**22**, pp.757-765.
- [5] **Sengor, N.S., Karabacak, O. and Steinmetz, U.**(2008). A Computational Model of Cortico-Striato-Thalamic Circuits in Goal-Directed Behaviour, Proceedings of International Conference of Artificial Neural Networks(ICANN'08), *Springer-Verlag*, pp.328-337.
- [6] **Denizdurduran, B.**(2012). Learning How to Select An Action: From Bifurcation Theory To THE Brain Inspired Computational Model, I.T.U. Graduate School of Science Engineering and Technology, *M.Sc. Thesis*, [www.simmag.itu.edu.tr/yayinlar/denizdurduran\\_Msc\\_Thesis.pdf](http://www.simmag.itu.edu.tr/yayinlar/denizdurduran_Msc_Thesis.pdf)
- [7] **Alexander, D.E., Crutcher, M.D. and DeLong M.R.**(1990). Basal gangliathalamocortical circuits: Parallel substrates for motor, oculomotor, "prefrontal" and "limbic" functions. *Progress in Brain Research*,**85**, pp.119-146.
- [8] **Haber, S.N.**(2010). The reward circuit: Linking primate anatomy and human imaging. *Neuropsychopharmacology Reviews*,**35**, pp.4-26.
- [9] **Dayan, P. and Balleine, B.W.**(2002). Reward, motivation, and reinforcement learning,*Neuron*,**35**(2), pp.285-298.
- [10] **Krichmar, J.L.**(2008). The Neuromodulatory System: A Framework for Surcial and Adaptive Behavior in a Challenging World, *Adaptive Behavior*,**Vol.16**, No.6, pp.385-399.
- [11] **Girard, B., Tabareau, N., Phama, Q.C., Berthoz, A. and Slotine, J.J.**(2008). Where neuroscience and dynamic system theory meet autonomous robotics: A contracting basal ganglia model for action selection,*Neural Networks*,**Vol.21, Issue 4**, pp.628-641.

- [12] **Denizdurduran, B. and Sengor, N.S.**(2012). Learning How to Select an Action: A Computational Model, *International Conference on Artificial Neural Networks (ICANN)*, Lausanne, Switzerland.
- [13] **Robotis e-Manuel**, <http://support.robotis.com/en/>, Retrieved on 01.04.2013.
- [14] **Schultz, W., Dayan, P. and Montague, P.** (1997). A Neural Substrate of Prediction and Reward, *Science*,**275**, pp.1593-1599.
- [15] **Sutton, R.S. and Barto, A.G.**(1998). Reinforcement Learning: An Introduction, MIT Press, Cambridge, MA, A Bradford Book.
- [16] **Chuhma, N., Tanaka, K.F., Hen, R. and Rayport, S.**(2011). Functional connectome of the striatal medium spiny neuron, *J. Neurosci.*,**31(4)**, pp.1183-1192.
- [17] **Ercelik, E., Denizdurduran, B. and Sengor, N.S.** (2013). A Bioloid Robot Implementation of Computational Basal Ganglia Circuit, *Proceedings of 11. National Neuroscience Congress*,**P-031**, p.141.



**PHOTO**

## **CURRICULUM VITAE**

**Name Surname:** Emeç Erçelik

**Place and Date of Birth:** Denizli/07.07.1990

**Address:** ITU Golet Ogrenci Yurtlari 7.Blok Maslak/Istanbul

**E-Mail:** ercelike@itu.edu.tr

**B.Sc.:** Electronics Eng., Istanbul Technical University (expected date, June 2013)

### **PUBLICATIONS/PRESENTATIONS ON THE THESIS**

- **Ercelik E.**, Denizdurduran B. and Sengor N.S., 2013: A Bioloid Robot Implementation of Computational Basal Ganglia Circuit *11. National Neuroscience Congress*, Apr.28-May 1, 2013 Izmir, Turkey.