ARTIFICIAL IMMUNE SYSTEM BASED DYNAMIC OBSTACLE AVOIDANCE ROBOT

Robotics Lab.
Department of Computer Science and Engineering
Indian Institute of Technology Guwahati
Guwahati (Assam)

Guided By:

Prof. S. B. Nair
(sbnair@iitg.ernet.in)
Shashi Shekhar Jha
(j.shashi@iitg.ernet.in)

Submitted By:

Aakash Bharti
(a.bharti@iitg.ernet.in)
Mayank Gupta
(mayank.2013@iitg.ernet.in)
Pulkit Verma
(v.pulkit@iitg.ernet.in)

TABLE OF CONTENTS

1. INTRODUCTION

1.1. Abstract

This AIS technique is based on Clonal Selection in which eligible antibodies are selected using positive selection while ineligible are discarded if pain generated by current antibody is more as compared to previous one or antibody was not able to complete all its behaviours (basic actions) due to some obstacle. The AIS is based on navigation of autonomous robots where unfavourable situations can occur comprising static, dynamic obstacles and dynamic robots. The robot tries to learn how to avoid obstacles and other robots by continuously eliminating ineligible antibodies. The antigen comprise of environmental conditional taken by ultrasonic sensors.

1.2. Basic Concepts

1.2.1. Antibodies :

The biological immune system consists of antibodies and lymphocytes. Millions of lymphocytes are generated in bone marrow each of which some molecular structure and is producing of antibodies from its surface. In our project we are initially generating antibodies with randomly generated fields taken by ultrasonic sensors. These antibodies recognize specific foreign substances (antigens) that invade living creatures.

1.2.2. Epitopes and Paratopes

Paratope is structure on antibody that makes contact with closest (high bond affinity) structure of antigen called as epitope. The Clonal selection helps in improving response to antigen over time. Simultaneously death of inactivated antibodies occurs with time while concentration of eligible antibodies increases.

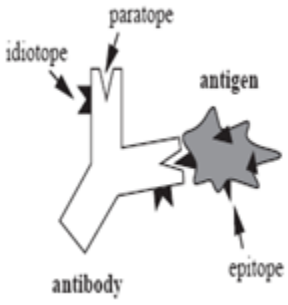


Fig 1: Structure of Antigen and Antibody in Biological World

1.2.3. Clonal Selection

According to Clonal Selection [1], selection and reproduction of antibodies is inspired by the *affinity* of a particular antibody to following characteristics: selection and cloning of the most activated antibodies, death of inactivated antibodies and an antigen. Clonal selection establishes the idea that only those antibodies that recognize the antigen with higher affinities proliferate, thus being selected against those that do not [2]. The selection model is characterized by selection and cloning of the most activated antibodies, death of inactivated antibodies, maintenance of a memory cell and maturation of antibodies by selection, reproduction, and mutation through generations.

1.3. Related Work

The AIS has broadly two models - The Clonal Selection model wherein the more useful antibodies are filtered using positive selection while the others are discarded using negative selection. Antibodies that have high performance become memory cells thus aid in boosting the secondary response when the same antigen strikes again. The second model is the Idiotypic or the Immune network model suggested by Jerne [2] where the antibodies autonomously form a network and provide act as a defense mechanism when attacked by an antigen. An AIS based mechanism for the navigation of an autonomous mobile robot by blending both these models was developed by Nair et al.[3] This hybrid mechanism was tested on a real robot situated in an unknown environment comprising obstacles and unfavorable conditions.

2. PROPOSED MODEL

2.1. Salient Features

All the works related to the topic of Autonomous Robot Navigation Systems developed till date have mainly dealt with static obstacles, hence the environment of the robot does not change in such cases. In most practical situations such ideal environment is not available. The robots must be adaptable enough to work in such situations. Also in many cases it is not possible for a single robot to perform a task. In such cases where multiple robots are involved it is very important that robots do not collide with each other. Also in similar scenarios a lot of time is spent on collision avoidance. Hence the robots must be capable of performing tasks in such dynamic environments.

2.2. Antigen and Antibody Structures

In the proposed system for autonomous navigation, the robot corresponds to an organism. It tries to explore the most comfortable or zero pain regions in the given environment while avoiding the unfavorable conditions encountered in between. The incoming set of values obtained from the sensors play the role of an antigen. Whenever the robot enters a vulnerable area, the robot is said to be attacked by an antigen. This results in an innate pain within the robot. To decrease the pain and to rescue the robot from the vulnerable areas, back to the normal condition, the immune system is triggered.

Initially the robot generates multiple antibodies against the antigenic attacks to discover the zero pain regions inside the environment. The correct antibody is chosen during an attack using clonal selection. Antibodies are generated randomly before the start of the experiment. Antibodies that contribute to an increase in pain are decreased in concentration (negative selection) while those which decrease it are increased in concentration (positive selection). Whenever the concentration reaches zero, the antibodies are removed.

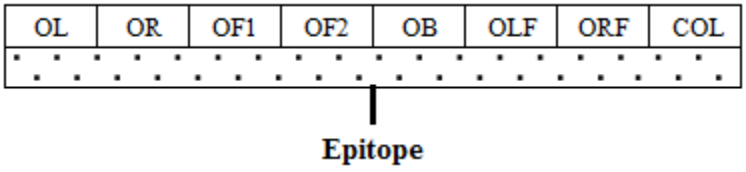


Fig 2: Structure of Antigen

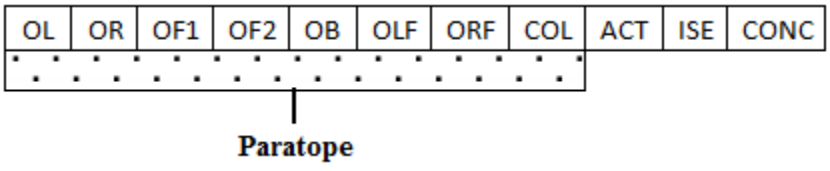


Fig 3: Structure of Antibody

An antigen in the robot's world constitutes the environment perceived by it via its sensors. It thus provides information about the current environment around the robot which includes the presence of obstacles, pits, slopes, dark regions, etc. based on its sensing capabilities. As the antigen is a direct derivative of the environment, the same representation could be used for any environment. The external and internal conditions of the robot as sensed by sensors mounted on-board constitutes the antigenic epitope as shown in Figure 2.

OL : Obstacle in left
OR : Obstacle in right
OF1 : Obstacle in front 1
OF2 : Obstacle in front 2
OB : Obstacle in back
OLF : Obstacle in left forward
ORF : Obstacle in right forward
COL : Colour of path

- OF1 is the first reading of the front sensor; let us say at time t.
- OF2 is the second reading of the same sensor at time t+x.

As can be seen from this figure the antigen vector contains the internal and external conditions of the robot. Figure 3 depicts the structure of the antibody. The structure of the antibody is similar to that of an antigen except for the fact that it has a structure called the paratope in place of an epitope. In addition there are three extra fields:

ACT : Action (String of behaviours of length 5)
ISE : Is eligible
CONC : Concentration of antibody

In Figure 3, the Action field contains a set of rules or programs which guides the robot to perform a particular task to circumvent (or destroy) the problem (antigen) at hand. The robot is embedded with a basic set of behaviors which include:
1. Move forward
2. Move backward
3. Rotate left (45 degrees)
4. Rotate right (45 degrees)

The antibody action sequences generated are a random combination of these 4 behaviors. The antibodies whose action sequence leads to increase in pain are rejected by the system.

2.3. Antigen-Antibody Interactions

An affinity function defines the degree of interaction between the epitope of an antigen and the paratope of the antibody. We have used a linear distance function as the affinity measure to calculate the affinities between antigens and antibodies. An antibody is stimulated by an antigen if the distance between their paratopes and epitopes in all dimensions lie within respective threshold regions defined separately for each dimension. The lesser this distance the more is the stimulation.

2.4. Goals

2.4.1. Static Obstacle Avoidance

i. We have used two robots, there are static obstacles in the arena present randomly and robots learn to avoid each other using Clonal selection.

ii. The antibodies are generated randomly consisting of eight fields which try to match with antigen taking seven ultrasonic sensor values in respective directions from given environment namely front (by fixed sensor), backward (by other fixed sensor), 5 other values generated by rotating sensors, colour of path mounted in front. These values are compared with all antibodies having same type of fields, one concentration and one eligibility field, and action sequences.

iii. The closest antibody is executed in response to antigen generates pain which decides eligibility of antibody. Low pain means more eligible. When antibody becomes eligible its concentration increases, positive selection. Others are deleted, negative selection.

2.4.2. Single Lane Following

i. “Single Lane” it is one of the task in which two robots try to move towards each other in single lane path they will learn such that out of two robots only one will go while the other will back off. This is done by using black colour path which decreases the pain as compared to non-black surface that has more pain.

ii. Since black colour decreases the pain hence the robots will try to search black line rather than moving randomly in non-black surface.

iii. Line following is done on the right edge of black line so when robot deviates from right edge it tries to find the same edge.

2.4.3. Dynamic Obstacle Avoidance (Different kind of Robot)

i. In this task a separate robot is made to follow a black line back and forth continuously. There are two other robots moving randomly in the arena which learn to avoid this separate robot following black path.
ii. Again this uses Clonal selection where the antibody which do not avoid obstacle or do not complete all its behaviours gets rejected, while others are preserved.

2.4.4. Dynamic Robot Avoidance (Same kind of Robot)

i. In this task a separate robot is made to follow a black line back and forth continuously. There are two other robots moving randomly in the arena which learn to avoid this separate robot following black path.
ii. Again this uses Clonal selection where the antibody which do not avoid obstacle or do not complete all its behaviours gets rejected, while others are preserved.

2.5. Pain Function as Feedback

We have introduced a pain function which will act as feedback for the learning phase. The basic idea is that whenever an antibody action leads a robot into an unfavourable condition (generally a location too close to an obstacle) then the pain value increases. Hence the robot aims to decrease its pain value. This concept of pain function acts as motivation as well as feedback for the system.

The pain function is calculated as follows:

$$Pain = \frac{kc}{x2-x1} + \sum_{i=1}^{7} \frac{c}{(distance)_i} \quad , if\ x2 > x1$$

$$Pain = \sum_{i=1}^{7} \frac{c}{(distance)_i} \quad , otherwise$$

where, c is constant,
k is weight constant,
$(distance)_i$ is i^{th} field of the antigen.
x2 and x1 are reading of front sensor.
x2 is first reading, x1 is second reading.

- k is used to increase the pain more when an obstacle is approaching towards the robot.
- x2 and x1 are 2 readings taken by the front sensor so as to differentiate between static and dynamic obstacles. First x2 is read then after some time x1 is read.
- c is used so that pain will have proper values. If c =1 then the function effectively becomes $1/x$ and for x>1 the function gives very less values.
- i=1 to 7 because there are seven sensor values as mentioned in the structure of antigen and antibody.

2.6. Colour Field in Antigen

The colour field in antigen is used when we have the robot has to follow the lane in Goal 2. So whenever the black colour is detected the path following antibody action comes into effect.

2.7. Choosing the correct antibody

Antibodies are chosen from the set of eligible antibodies according to the following formula:

$$\text{Chosen Antibody index, i} = \min\left\{\frac{distance}{1+\log(conc)}\right\}$$

The main idea here is to choose the antibody whose affinity is largest, i.e. distance is minimum and concentration is largest. We have taken log of concentration as distance is taken as more important factor than concentration while performing the experiments.

2.8. Removing the bad antibodies

- We have given the concentration of all antibodies as 2 initially.
- Now whenever the pain of the system increases after executing an antibody action, the concentration is decreased by 1, and whenever the pain decreases the concentration is increased by 1.
- When the concentration becomes 0, the antibody is tagged ineligible and is never taken into consideration after that.

3. TECHNICAL DETAILS

3.1. Explanation of Code

- The code for the project was written entirely in Java. It is client-server architecture.
- There are three entities involved:
  1) **Relay server** (Not developed under the project): This is a server written in Java, which listens to the requests made by a client, gets the opcode from the request, and sends this opcode to the robot.
  2) **The Java Client**: This program sends the opcode of the operation to be performed on the robot to the relay sever. It uses socket programming for the task.
  3) **Nxt_bin** (Not developed under the project): This program resides in the robot brick, listens to the relay server, gets the opcode, and executes the equivalent instruction.

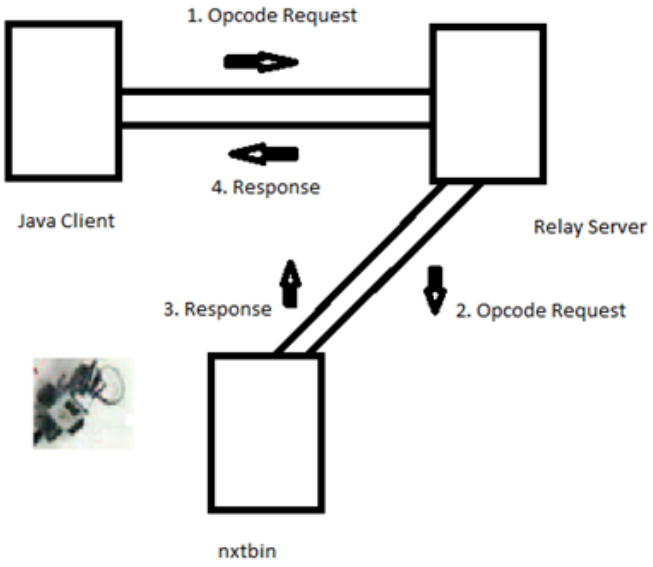


Fig 4: Client-Server Communication

- The Java Client is a new interface that we have written, as the opcodes need to be generated for the desired actions. These opcodes were present in the PROLOG code, but not in Java. It uses following files:
  1) *LejosInterfaceJava.java*: This class takes the request for the operation, converts to opcode and sends it to RobotConnect.java.
  2) *RobotConnect.java*: This class establishes connection with a brick, and communicates the opcodes received.

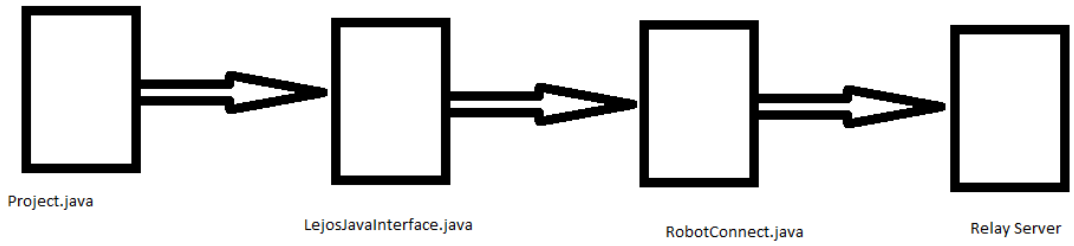


Fig 5: Program Flow

- The different classes present are:
  1) *Antigen.class*: This contains the fields that constitute the antigen.
  2) *Antibody.class*: This contains the antigen fields, and other antibody attributes, like concentration, action, eligibility.
  3) *Main.class*: This contains methods for:
     i. Reading the sensor values and returning the created antigen.
     ii. Selecting the most appropriate antibody for the given antigen.
     iii. Logging feedback.
     iv. Saving and getting the antibody collection from the disk.
  4) *Behaviour.class*: This class defines the basic actions that can be performed by the robot. *Antibody.action* field contains a permutation of numbers between 0 and 3. This class defines which number corresponds to which action.

3.2. Program Compilation

The files can be compiled using the standard *javac* command.
$ javac Project.java
$ javac RobotConnect.java
$ javac LejosJavaInterface.java

3.3. Program Execution

1. Execute the *nxtbin* file on the brick.
2. Start the relay server on a command prompt:
   nxjpc relay_server <port number>
3. Execute Main class on another command prompt:
   java Main <brick name> <port number on which the relay server is running>

4.RESULTS AND ANALYSIS

4.1. Experimental Setup

4.1.1. Static Obstacle Avoidance

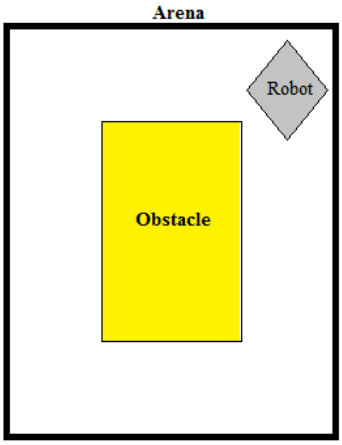


Fig 6: Arena for Static Obstacle Avoidance

We started with 400 antibodies and finally stopped the learning phase when 175 antibodies were remaining. The robot performed better after the learning was complete. It was able to find low pain regions more easily. The learning phase lasted about 2.5 hours.

4.1.2. Single Lane Following

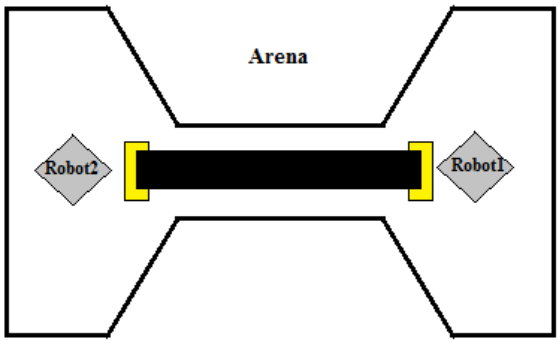


Fig 7: Arena for Single Lane Following

We started with 500 antibodies and finally stopped the learning phase when 30 antibodies were remaining. After multiple learning phases, the robot did not perform any better than before. It was able to find low pain regions more easily but there was a lack of motivation in crossing the lane. Detailed discussing regarding this is mentioned while explaining "Problems Faced". The learning phase lasted about 2-3 hours in each iteration. The yellow strips at the end helps robot in identifying the start/end of path.

4.1.3. Dynamic Obstacle Avoidance (Different kind of Robot)

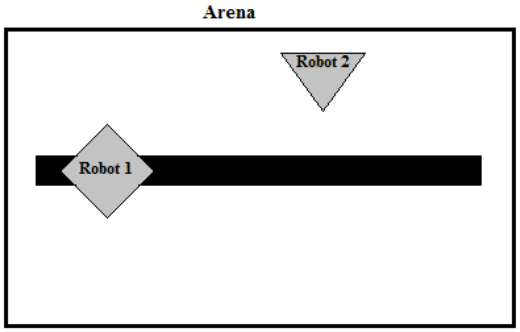


Fig 8: Arena for Dynamic Obstacle Avoidance

We started with 400 antibodies and finally stopped the learning phase when 30 antibodies were remaining. After multiple learning phases, the robot did not perform any better than before. It was able to find low pain regions more easily but Robot 2 was getting hit by Robot 1 constantly as it was not designed to detect an obstacle. Detailed discussing regarding this is mentioned while explaining "Problems Faced". The learning phase lasted about 1-2 hours in each iteration.

4.1.4. Dynamic Robot Avoidance (Same kind of Robot)

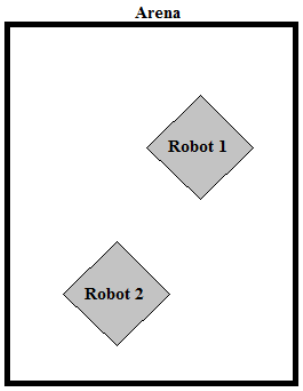
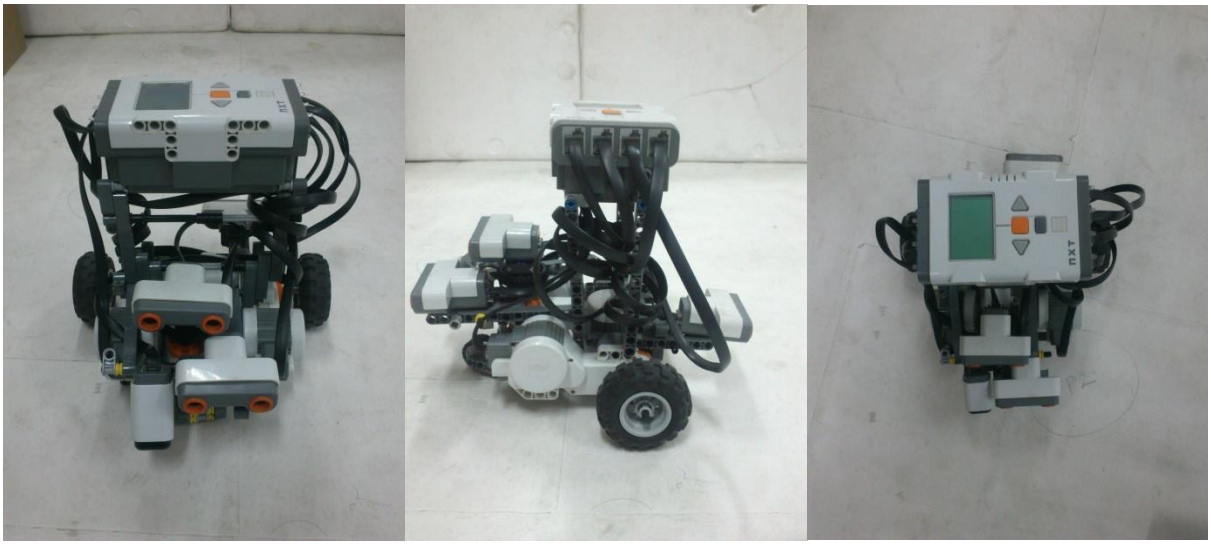


Fig 9: Arena for Dynamic Robot Avoidance

We started with 200 antibodies and finally stopped the learning phase when 30 antibodies were remaining. The robot performed better after the learning was complete. It was able to find low pain regions more easily. The learning phase lasted about 1.5 hours.

4.2. Snapshots



Front View *Side View* *Top View*

Fig 10: Snapshot of the robot used in experiments.

4.3. Problems Faced

1) ***Long time to converge*** performing with large number of antibodies was taking a long time to converge hence we have mainly worked with the antibodies from 500 to 1000.

2) ***Using long actions sequences of antibodies:*** The initial number of behaviours we chose was 15. This resulted in long times for completion of the antibody action sequence, making the system very slow. Also, since the robot's effective movement was more, there were chances that it went to the complete different locality from the one where it started. So, the useful/correct antibodies were getting rejected. So, we changed the length to 5.

3) ***Single lane behaviour without line following by two robots***

   i) Initially we tried this behaviour without any line following but the problem was if they enter the lane at different angles, they were unable to detect each other (due to position of sensors), and it may happen they may not get into the lane itself.

   ii) ***Black Line following for single lane behaviour by two robots:*** To solve the problem we have used "1 extra light sensor" to follow the right edge of black line by both the robots. But still they were not able to learn to cross lane efficiently.

   iii) To make robot to search black line easily we decrease the pain value when robot finds black line.

iv) To solve above problem we have used two types of antibodies, one which works on black line (antibody that gets generated when robot detects the black line) and other that works on non-black line (normal antibody that works on non-black surface). But here the problem was the non-black antibodies are getting use a lot more as compared to black line antibodies because the black path was very small as compared to non-black path.

v) One more problem was that the robots were unable to follow the right edge; the reason found was the position of light sensor which was mounted at very low height. Hence we increased its height and tilted it little bit.

vi) Even with above modifications the task was not done correctly. So we decreased the behaviours of antibodies, we used only two behaviours (a) moving forward (b) moving backward on finding the black line.

vii) Initially the probabilities of these two behaviours were 50-50%. So, the forward movement was mostly cancelling the backward movement. So, later on we change it to 75-25% (forward-backward). But still we were not able to do make it learn well so that only 1 robot out of two crosses the lane while backing off.

viii) There was no *motivation* for the robots to cross the single-lane bridge. So, they preferred to be on the same side in which they started instead of crossing the bridge and going to the other side.

4) ***Rotating ultrasonic sensor was not coming back to original position after rotating*** it was due to the wire which must be loose little bit for smooth rotation but due to continuous rotation it was unable to get back to front direction hence we had to manually fix it.

5) ***In Dynamic obstacle avoidance*** the robots moving randomly were not able to learn initially because of position of ultrasonic sensors which fails to detect line following robot when arrived towards it at some angle.

6) ***Including Concentration for calculation of pain*** initially we were using only affinity (distance between antigen and antibody) for choosing best antibody but for more optimal results we have also included "concentration" in the formula. This led to even slower convergence of the system.

4.4. Results

4.4.1. Static Obstacle Avoidance

As we can see, the deviation in the value of pain decreases with time. That is, the robot become stable as the 'wrong' antibodies get discarded.

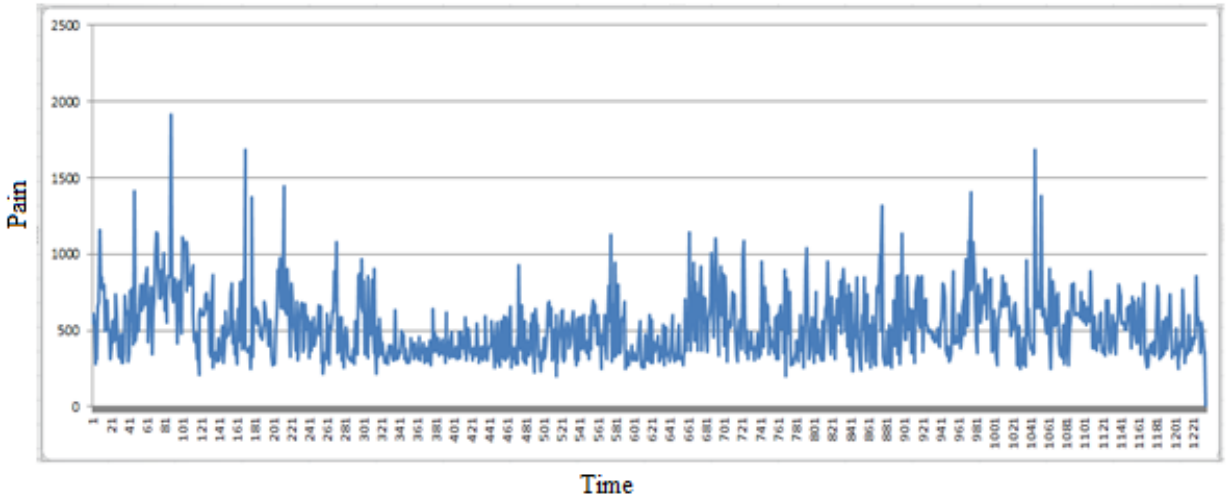


Fig 11: Pain vs Time graph for Static Obstacle Avoidance

We started with 400 randomly generated antibodies and in the end, 175 were remaining.
On analysing the behaviours of the remaining antibodies, we found that

- Almost 60% behaviours were of rotation, and
- Only 40% were of moving forward or backward.

Thus, the robot tends to be in its place instead of moving or traversing the arena once it is in a stable state.

When we moved the robot to a larger arena with different obstacles and noted its performance.

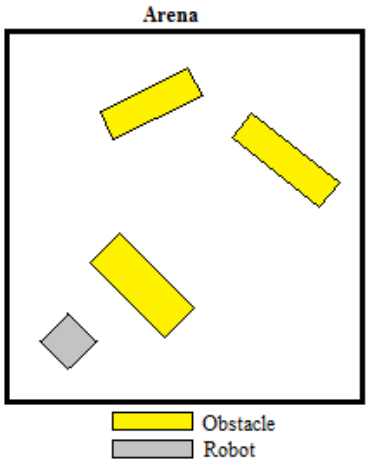


Fig 12: Arena to test the performance of "Static Obstacle Avoidance" antibodies

In this case the learning was stopped and the robot has to navigate using the antibodies that were remaining after the learning phase. It was found that once the robot finds a place where pain is less it tends to stay there and does not explore the whole arena. The reason may be attributed to the fact that the antibody actions having more rotational behaviours were remaining at the end of learning phase.

4.4.2. Dynamic Robot Avoidance (Same kind of Robot)

As in the previous case, the deviation in the value of pain decreases with time. That is, both the robots become stable after some time as the 'wrong' antibodies get discarded.

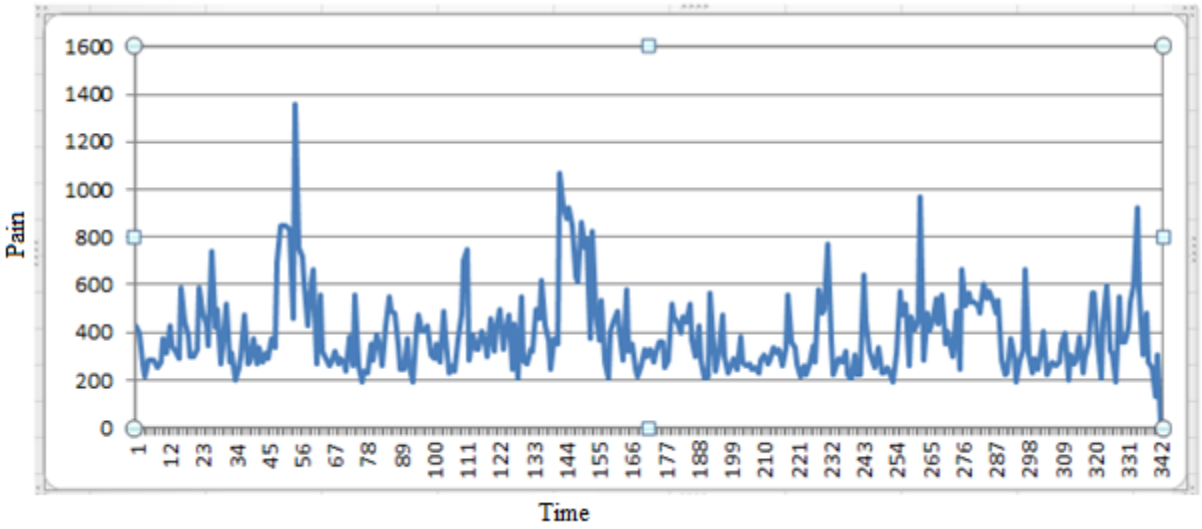


Fig 13: Pain vs Time graph for Robot 1 (Dynamic Robot Avoidance)

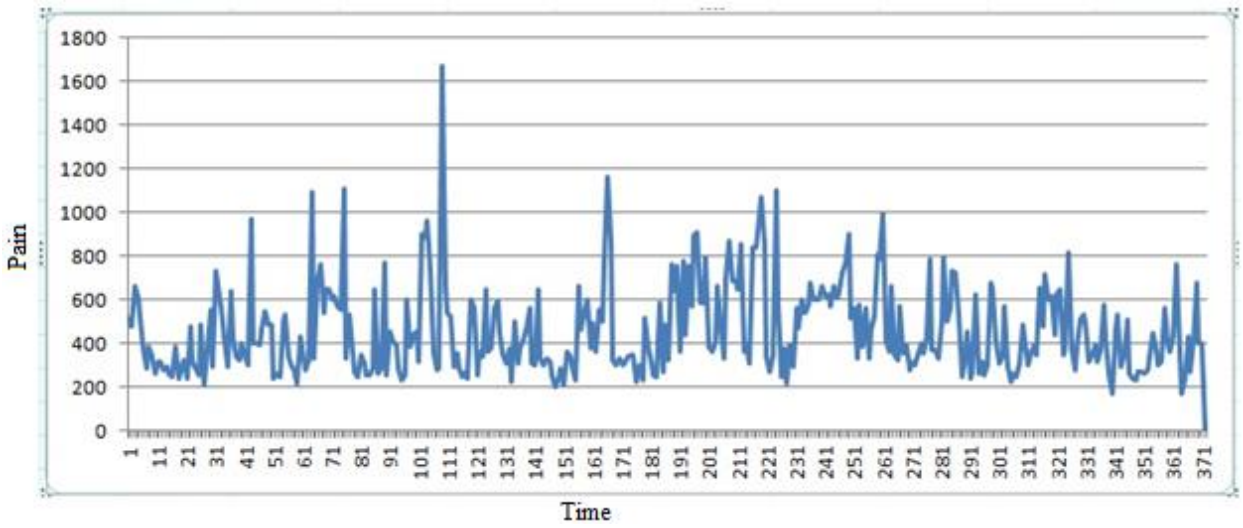


Fig 14: Pain vs Time graph for Robot 2 (Dynamic Robot Avoidance)

5. CONCLUSION

5.1. Discussion

The results of the experiments show that the training of robots to work in an environment where dynamic obstacles are present can be done. The main point to keep in maid is the choice of fields of antibody and antigen.

The experiment gave positive results in 2 cases (Goal 1 and Goal 4) and negative results in other 2 cases (Goal 2 and Goal 3).

5.2. Deployment outside Lab

The 2 cases with positive output can be deployed outside lab if the robots are trained properly. One important thing to note is that the surface should be plane as the current design of the robot does not take into account the rough terrain.

5.3. Future Work

- Antibodies and Antigens could be modelled in a better way to achieve Goal 2 and Goal 3.
- Information sharing between the robots can be used in the case when similar robots try to work in same environment. This may lead to better convergence time during learning phase.

6. REFERENCES

[1] J. D. Farmer, N. H. Packard, and A. S. Perelson. The immune system, adaptation, and machine learning. *Physica D: Nonlinear Phenomena*, 22(1):187–204, 1986.

[2] N. K. Jerne. Towards a network theory of the immune system. *Annales d'immunologie*, 125C(1-2):373–389, Jan. 1974.

[3] K. Shrivastava, S. S. Jha, and S. B. Nair. Autonomous Mobile Robot Navigation using Artificial Immune System. *Proc. of Conf. on Advances In Robotics* (AIR '13), Article 82.