ACTIVATION MECHANISM

IN ROBOTS

By

LAKSHMI PRASAD BANALA

Bachelor of Technology in Computer Science and

Engineering

Jawaharlal Nehru Technological University

Andhra Pradesh, India

2005

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 2009

ACTIVATION MECHANISM

IN ROBOTS

Thesis Approved:

Dr. Johnson P. Thomas

Thesis Adviser

Dr. Venkatesh Sarangan

Dr. Nohpill Park

Dr. A. Gordon Emslie

Dean of the Graduate College

ACKNOWLEDGMENTS

First, I thank my advisor Dr. Johnson P. Thomas, for his continuous support during my master's thesis. He helped me in my thesis writing and provided me with ideas as needed. I would like to thank him once again for offering his invaluable time and help in implementing the ideas which drove me to a successful completion of my graduate degree. I would also thank my committee members Dr. Venkatesh Sarangan and Dr. Nohpill Park for their suggestions.

I would also like to thank Dr. Ramesh Sharda (Regents Professor in MSIS, OSU) for introducing me to the real world work culture. He taught how to think the future impact of current issues, the importance of personal behavior, presentation skills and attitudes in deal with any problems. He also helped me to improve my working pace and programming skills through directly involving me in a real time project. I would also thank Dr. Joyce Lucca (Post Doctorial Fellow) for providing me with moral support and strength through my tough times. Dr. Joyce Lucca taught me the importance of people and resource management.

I thank Pradeep Koduru, Satish Aduri and many others for sharing my happy times. Their friendship leaves me with many fond memories of my graduate life at OSU.

I would like to express my affections and love towards my sister, Gayathri, brother-in-law, Srinivas and to my sweet niece “Sunny” (Sheershita).

Last and most importantly, I would like to express my deep love and affection to my parents for providing unconditional support and encouragement throughout my life. My each and every achievement is because of them. I whole heartedly dedicate this thesis to them.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Robotics is the science and technology of robots, their design, manufacture and application. Robots have been defined as a mechanical device that can perform complex task. Robots are being extensively used in wide range of applications such as deployment in demolition areas, fire fighting, bomb diffusion, nuclear site inspection, deep sea exploration and so on. In a dynamic environment, robots are more likely to encounter failures while executing their instructions. It may not be possible for humans to intervene and handle these failures [1] [4] [5] [9]. Robots need to respond themselves to such failures and they should be able to recover from the encountered failure. By adding artificial intelligence to a robot, it becomes an unsupervised worker, who deals with the changing environment on its own.

An ideal robot would imitate the human in every manner. Humans can make decisions to react to different situations and so on. The human body has multiple subsystems, all working independently of each other all the time [7]. The human immune system is one such subsystem. It is responsible for recovering the human body from any kind of invasion or an attack or a failure. The human immune system provides a model that can be applied to the field of robotics to address the issue of failures in robots. Emulating the immune system in robots would form the basis for a robot to recover from attacks and failures. Architecture based on the human immune model to develop a robot that can self detect failures and furthermore recover from failure to normal state.

The artificial immune system proposed for robots contains three subsystems, namely a recognition unit, an activation unit and a response and recovery unit. The recognition unit detects the failure. The recognition unit sends the failure information to the activation unit which then recommends a recovery action to be taken by the response and recovery unit to solve the encountered failure. The response and recovery unit checks the feasibility of the solution sent by the activation unit and implements the action, if it is feasible. If the action send by the activation unit is not feasible, the response and recovery unit devices its own recovery action. The response and recovery unit then sends a feedback to the activation unit. Based on the feedback the activation unit learns and adapts thereby providing more probable and feasibly correct solutions for the future problems.

In this thesis, we focus on the activation unit. The activation unit is placed between the other 2 units (the recognition and response and recovery). This unit is responsible for the robot’s learning mechanism on failure recovery. The activation unit also maintains storage for previously encountered problems and actions taken at that time. After detecting a failure, the recognition unit passes the information about the failure to the activation unit. The activation unit first analyzes the information received from the recognition unit. Based on the stored information and the analyzed information, the activation unit generates plausible solutions for the encountered problems. The best action among the pool of actions is selected and sent to the response and recovery unit. Based on the feedback sent by the response and recovery unit, the activation unit updates the stored information and learns dynamically. The updated information is used for future failures.

The detailed explanation about the proposed model and its implementation is discussed in the following sections. Chapter 2 presents the literature review of the robotics and the human body model. In chapter 3 a detailed description about the human body immune system is given and chapter 4 defines the problem specification. Chapter 5 gives a detailed description about the implementation and provides simulation results and finally chapter 6 concludes the thesis.

CHAPTER II

REVIEW OF LITERATURE

In this chapter, we give a review of earlier implementations of robotics for failure recovery based on human body model.

Tian F., Deng Q., Zuren Feng and Ping Jing [5] compared the results of behavior network based and artificial immune network based methods on robot navigation, behavior coordination and communication strategy. They found that the artificial immune network based method shows better performance.

Richard Canham, Jackson A.H. and Andy Tyrrell [6] proposed a system for error detection scheme in object avoidance while the robot is moving around based on artificial immune system. They used two robots Khepara robot and BAE SYSTEMS RASCAL ™ robot for their research. They implemented the same strategy in both the robots. Robot movement is controlled by a set of sensor values generated from the sensors attached to the robot. Artificial immune system is used to find the best set of sensor values that help robot not to collide with the object.

Steven A. Hofmeyr and Stephanie Forrest [2] implemented the artificial immune system (AIS) for computer network security to avoid illegal intrusions. They implemented the AIS in terms of detecting the abnormal set of mask digits which represents recognition of intrusion. The AIS is also self-learning for acting effectively to any future intrusions.

Kim J. and Peter Bentley [3] also worked on a network intrusion detecting system. Internal and external intrusion on the computer system is being detected through implementing an immune system.

Chingtham Tejbanta Singh and Shivashankar B. Nair [1] proposed an application for an autonomous learning mechanism for robots through a natural immune system. Their application deals with two Lego robots with trail tracking the given route. Two robots were considered, one is a helper robot, works in damage-control mode for guiding the other robot. An artificial immune system is implemented in guiding and tracking the follower robot. The helper robot sends an Infrared signal to the other robot. When the follower robot receives that signal it follows the helper robot. Sometimes the helper robot moves fast enough since it assumes that the follower robot is performing well. But when it finds the follower robot is not following, it searches for the follower robot and re-establishes the connection with the follower robot.

Human immune system is implemented in the earlier researches by many people. They used the human immune model for robot navigation, error/intrusion detection in networks, in guidance (robots helping each other) etc. Our area of implementation is on the recovery procedure on detection of a failure. We also look at how to analyze the *fail* situation and determine possible solutions. Our main area of interest is on adaptive learning that keeps the information up-to-date inside the robot and provides plausible solutions to the failure.

CHAPTER III

PROBLEM DESCRIPTION

Typically robots are employed to work in a hostile environment where human intervention is not possible [4]. These work as a group to achieve a common task in which robots are dependent on each other to be successful. A base station gives directional and other instructions to the group of robots. Since the robots are mobile, possible failure can be due to obstacles resulting in communication breakdown, failure of sensors, energy depletion etc. We focus primarily on communication failure between the base station and the robot which could be due to noise, obstacles or the robots moving beyond the communication range in the network. At this point the robot which has encountered failure should not come to standstill; instead the robot should be able to predict or detect the possible failure and take the necessary action to recover back to a safe state and continue in a normal way.

3.1 Application:

Consider a Base station which gives instructions to a network of robots which work together to find information in an area affected by earthquake. Each robot has its own task to sense information about the destruction that has occurred in a particular area and send information back to Base station.

When a robot encounters a failure, it should use some failure detection mechanism to detect the possible failure, the cause of failure and recover back to the safe state so that normal execution can resume.

CHAPTER IV

HUMAN BODY IMMUNE MODEL

The Human body is an excellent complex module that is built with many subsystems that work independently. Each subsystem is called based on the experienced situation. In case of any kind of intrusion from an antigen (substances such as toxins or enzymes in the microorganisms or tissues that the immune system considers foreign) the subsystem that responds is the immune system.

Immunity is defined as inherited, acquired or induced resistance to an infection. Human body is in-built with two types of immunity. They are:

1. Innate immunity: This is the first line of defense mechanism in the human body that acts against any kind of invasion. This immunity is antigen-independent.
2. Adaptive immunity: This is a learning process inherited in the human body which creates antibodies (protein that neutralizes an antigen) specific to an antigen on its own.

The main components of the human immune system are White Blood Cells (WBC), fibroblasts and blood platelets. WBC plays an important role in the immune system by providing necessary defense (antibodies) against foreign bodies. Fibroblasts help in remodeling the damaged tissues. Platelets avoid further blood loss in case of any wounds or cut parts.

Lymphocytes are the principle components of immune system that are present in WBC. Lymphocytes are constituted of T-cells and B-cells [7]. T-cells are produced in bone marrow but mature in the thymus. Unlike T-cells, B-cells are produced and mature in bone marrow. T-cells will be circulating in the blood stream all through the body. They scan the body surface to find the foreign antigens or foreign behaviors. So they are also known as Immune Surveillance. B-cells produce antibodies for an antigen.

The macrophages of WBC's are located on the surface of the body cells. These are the primary contact for the invaded antigen. Whenever any foreign body comes in contact with the human body cells, the macrophages engulfs the foreign body and decomposes them to release their amino acids. The T-cells in the blood stream gets activated and differentiate the foreign body by comparing the chemical structure of the self cells with the foreign body amino acids. If the comparison fails, T-cells alarm the other cells by releasing a chemical substance in to the blood stream. This chemical substance activates the T4 killer cells and B-cells in the blood stream. T4 Killer cells weakens the amino acid structure of the foreign body. While the B-cells produce unlimited number of antibodies (antigen-specific) that kills the foreign body cells.

The antigen-specific antibodies that are left remaining after killing the foreign body cells get transformed into memory cells. The memory cell holds the structure of the foreign amino acid and the antibody used to destroy it. These cells reach cell mature stations (bone marrow for B-cells and thymus for T-cells) through the blood stream. Also the memory cells help in mounting a strong attack next time, if the same antigen invades.

Wound (internal or external) healing process [11] will come into action after killing the foreign bodies. This process includes 4 steps. They are haemostasis, inflammation, proliferation or granulation and remodeling or maturation. Blood platelets cover the wound to avoid further blood loss, this phase is called haemostasis. The defense mechanism against the invaded antigen comes under the inflammation phase. The basic skin provided by the fibroblasts comes under proliferation phase. Finally, covering the wound with original skin and cleaning the dead cells by scavenger macrophages comes under remodeling phase.

CHAPTER V

PROPOSED APPROACH

The proposed immune system in robot is shown below:

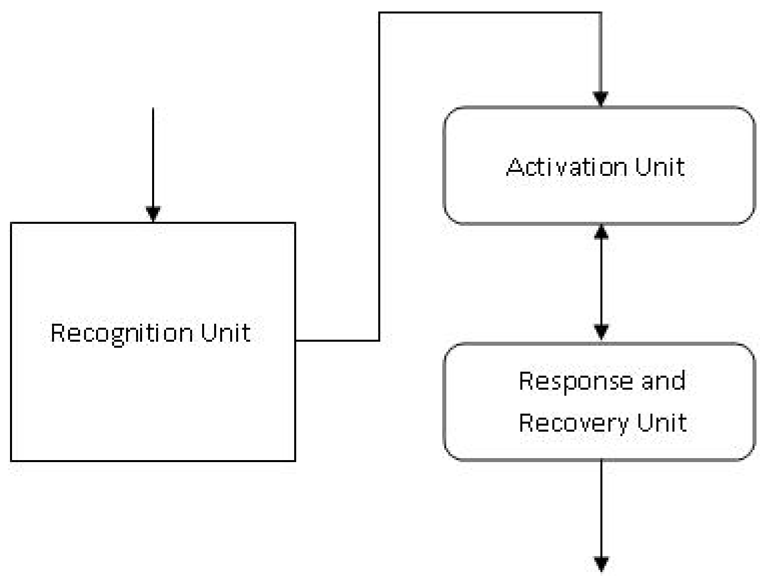

Fig 1: Robot Immune System

The recognition unit is to sense a communication failure. Communication failure might occur due to an obstacle, message loss or due to a robot moving to an unreachable position from neighboring robots in a network. This unit keeps checking the robot continuously for a communication failure. Once it detects a failure, it notifies the activation unit by sending the probability of cause for the failure and nature of the failure.

The two main functionalities of the activation unit are providing more probable solution to the encountered problems and updating the knowledge repository. The knowledge repository maintained by the activation unit consists of previously encountered problems and actions considered to those problems.

When the activation unit receives information from the recognition unit, the activation unit analyzes the information and sends an action to the response and recovery unit. After executing an action by recovery and response unit, it sends a feedback to the activation unit. Based on the feedback, the activation unit updates its repository and uses this information when a failure happens again.

The final unit is the Response and Recovery unit. This unit is responsible for taking actions that would recover the robot back to a normal state. The response and recovery unit checks the feasibility of the solution sent by the activation unit and implements the action, if it is feasible. If the action send by the activation unit is not feasible, the response and recovery unit devices its own recovery action. It sends a feedback to the activation unit after recovering.

CHAPTER VI

CORRELATION WITH HUMAN IMMUNE MODEL

6.1 Recognition Unit:

Initially the observation graph is defined for every robot in the simulation. This is similar to the amino acid structures that are present in the T-cells of WBC [7]. The robot waits for some time unit (say 30 units) and checks the connection back to

Antigen recognition	Recognition unit
1. Human body has predefined amino acids. T–Cells look for changes in patterns of amino acid of the self and foreign bodies. 2. Release chemicals when an antigen is detected which signals the other cells for further action against the foreign body.	1. Every robot has its own initial observation graph. Current observation graphs are determined whenever there is an input to the robot. Recognition unit looks for changes in the initial and the current observation graphs to detect a failure. 2. Sends related information about the failure to the activation unit for further action.

Table 1: Correlation of Human Immune System's Antigen recognition process with proposed Recognition Unit

the base station. This is being done to check whether the robot is within the communication range of the base station or not. If the robot receives acknowledgement from the base station then it assumes that it is connected to the base station. If not, the robot is not in the base station’s communication range or the robot is isolated.

This process is similar to the work done by macrophages, a type of cell present in the human body, which continuously checks for foreign behavior inside the human body. It alarms T-cells on finding a new behavior. The T-cells then check the foreign body’s amino acid structure with self cells, those that exists within the human body. Similarly, when a robot encounters the communication failure under study, using the approach that we have proposed, an observation graph is created for that robot and compared with the robot’s initial observation graph.

After studying the newly found amino acids, if the T-cells confirm a foreign behavior then the surrounding cells are alarmed and they will come to the aid of the damaged cell. Similar to this, our proposed recognition unit invokes the activation unit by sending the information about the failure. Table 6.1 depicts the correlation of antigen recognition with the proposed recognition unit.

6.2 Activation Unit

B-cells store the information about amino acid structures and antibodies that are used to kill the antigens that had invaded earlier [7]. These B-cells provide defense mechanisms against the invasion by foreign bodies. Similarly, a knowledge repository is maintained by each robot to store information about failures that had occurred earlier and the actions that were taken to recover from those failures. This information is used whenever a similar kind of failure happens to the robot in future.

Activation in human body	Activation unit in robot
1. T4 killer cells are responsible for initiating action on the foreign body. B–cells produce antigen-specific antibodies.	1. Information retrieval technique to analyze the information from the recognition unit. This technique also helps in finding the best possible solution for the current problem through ranking them.
2. Memory cells stores the structure of the antigen and the antibody, which is used to destroy them. This helps to act better next time whenever the same antigen is encountered.	2. Uses learning mechanism which improvises the problem specific learning in the robot.

Table 2: Correlation of Human Immune System's Activation process with proposed Activation Unit

The activation unit recommends an action to the response unit to bring the robot back to a normal position. This is similar to the B-cells that produce a tremendous amount of antibodies while the T4 Killer cells weaken the antigens. These antibodies are generated from previous knowledge stored in the memory cells and they will eventually kill the invading antigen.

The memory cells store the information about the antigen. Similarly, the knowledge repository will also update its database with the new information based on the feedback obtained from the response recovery unit after it executes the solution. The block diagram of activation unit is shown in section 6.3. Table 6.2 shown above correlates the memory cells and B-cells with the proposed approach.

6.3 Response and Recovery Unit

The response and recovery unit is responsible for bringing the robot back to a normal position to resume its normal execution. This is similar to the scavenger macrophages and B-cells in human body [7] [11]. The B-cells produce antibodies (if they are not in memory cells) specific to antigens. Similarly, the proposed response and recovery unit will implement the action specified by the activation unit. If the action sent by the activation unit is not feasible, it implements its own action to recover from the failure. In the human body, the newly created antigen-specific antibodies are stored in the memory cells for future reference. Similarly, the new action taken for the problem is sent back to the activation unit for the generation of future actions.

.

Response and recovery in human	Response and recovery unit
1. The platelets seal the blood vessels preventing further damage. 2. The surrounding cells come to aid the damaged cell and provide some kind of defense mechanism against infections. 3. Fibroblasts cells are used to remodel the tissues.	1. Executes the recovery mechanism to prevent further failure. 2. Receives action from activation unit and implements its own failure checking conditions with the recommended action to act against failure. 3. Response unit make sure that robot resumes to normal execution.

Table 3: Correlation of Human Immune System's Response and Recovery process with proposed Response and Recovery Unit

The Scavenger Macrophages cleans up all the dead cells and fibroblasts cover the area with skin which is a process of getting back to normal health. Similar to this process, after implementing the action the proposed unit recovers the robot from failure and resumes its normal operations. Table 6.3 shown above gives the correlation of human body recovery with the proposed response and recovery unit.

CHAPTER VII

PROPOSED SOLUTION

In this chapter, we will focus on the activation unit in detail. We describe the modules that make up the activation unit and present algorithms to choose a probable solution for a failure and for the robot's learning mechanism.

7.1 Input:

The input for the activation unit is received from the recognition unit. The input is a 3-tuple format

<*Problem, Cause, Probability*>

The *Problem* represents the problem as identified by the recognition unit, *Cause* states the reason for the problem and the *Probability* value is the probability of the problem occurrence.

7.2 Block Diagram:

The pictorial representation of the activation unit is shown below.

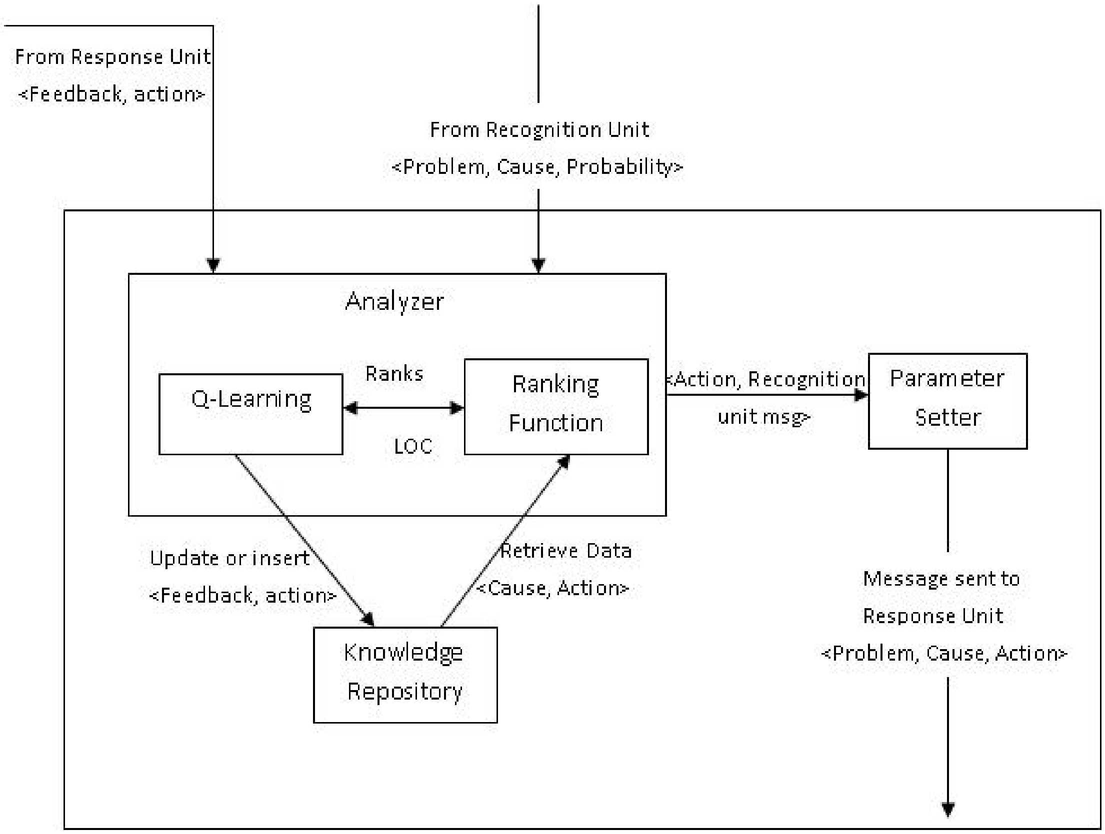

Fig 2: Block diagram for Activation Unit

The activation unit is subdivided into 3 different units:

1. Knowledge Repository: It stores the information about previously encountered failures and the action used to overcome that problem.
2. Analyzer Module: This module interacts with the knowledge repository. Its function is to choose an action for the failure and improve the robot's learning mechanism.

3. Parameter Setter (PS): This module's function is to set the parameters that are to be sent to the response and recovery unit based on the received message from the recognition unit and the action selected by the analyzer module.

The three parts of the activation unit work in coordination. The message from the recognition unit that invokes the activation unit is received by the *Analyzer* module. The *Analyzer* module requests the actions based on the failure specified in the message from the knowledge repository using the format *<Cause, action>*. The retrieved results and the message from the recognition unit are sent to *Parameter Setter (P.S)*. The P.S sets the message parameters *<Problem, Cause, Action>* and sends it to the response and recovery unit.

The activation unit works differently when it receives message *<Feedback, Action>* from the response and recovery unit. The *Analyzer* module updates the knowledge repository based on *Feedback*, if the *Action* is the action sent by the activation unit to the response and recovery unit earlier. If the *Action* received is different to the action sent by the activation unit earlier, the *Analyzer* module inserts in to the knowledge repository. The format used by Analyzer module to update or insert a new action in to the knowledge repository is *<Feedback, Action>*.

The activation unit receives input from both recognition and response units. The input from the recognition unit is for solving the current failure and the input from the response and recovery unit is used for updating the repository. The updated data is input to a learning mechanism which recommends a recovery action for a future failure.

7.3 Knowledge Repository:

The activation unit maintains a repository of actions for the previously encountered failures. The repository looks as follows:

Unit Name	**Problem**	**Details**	**Related to**	**Action**	**LOC**
Bluetooth	Communication	no response	Low power	Dock	0
Sensor	Range	Not in range	Distance	Roll back	0
Bluetooth	Communication	Message loss	Distance	Missed Message Request	0
Bluetooth	Communication	Isolation	Obstacle	Roll back	0
Sensor	Communication	Improper function	Low power	Dock	0

Table 4: Data in the knowledge repository (LOC – Level of Confidence)

Data is inserted in to the repository whenever the activation unit receives a new action that is not in the list, from the response unit. The format would look as follows:

<Unit name, Problem, Cause, Action>

7.4 Analyzer Module:

This module receives messages from both the recognition and the response and recovery unit. It has access to retrieve or update data from the knowledge repository. The probable action is selected when a message is received from the recognition unit. An insert or update takes place in case of a message received from the response and recovery unit.

The message from the recognition unit is passed to the Ranking function in the analyzer module. The ranking function analyzes the message and requests the plausible actions list for the encountered failure from the knowledge repository. The probable action is selected from the retrieved actions list using the message from the recognition unit. The detailed explanation about the ranking function is given in section 7.4.1.

The message from the response and recovery unit is passed to Q-Learning function in the analyzer module. The format would look as follows:

<Feedback, Action>

The Q-Learning function updates or inserts in to the repository based on the *Feedback* shown in the above format. If the *Feedback* is positive, the Q-Learning function just updates the knowledge repository with the value calculated using the Q-Leaning algorithm (shown in section 7.4.2). If the *Feedback* is negative, then the Q-Learning function inserts a new action for the problem in the knowledge repository. The new action is specified in the *Action* field shown in above format. The format of inserting data in to the knowledge repository is shown in section 7.3

The detailed explanation about the Q-Learning algorithm is given in section 7.4.2.

The overall flow of execution:

1. If the message is from recognition unit, then start from step 2 else start from step 7
2. Based on the message from the recognition unit, request the related actions from the Knowledge Repository
3. Rank the output using Information Retrieval technique
4. Select the best action from the ranked actions list
5. Send the action and state information to the Response and Recovery Unit
6. Store the remaining action/state list in temporary storage to find the reward (level of confidence) later using the Q-Learning algorithm.
7. Receive the input from the Response unit
   a. If the sent action is a success. Then assign the reward 100 to that action and calculate the level of confidence for the remaining actions in the temporary stored list (step 6) and update the repository
   b. If a new action is taken by the response and recovery unit then:
      1. Add the new action to the list (from step 6) and rank it 1
      2. Increment the other actions rank by 1.
      3. Create the current state and actions diagram

4. Calculate the level of confidence for all the other actions based on the new ranking

8. Repeat from step 1 through step 7 for any failure detected

The pseudo code for the above flow is shown below:

Calculating Ranks:

Initialized LOC_0=0 for all the actions

Function Information_Retrieval (problem, cause, P)

If $LOC_{(i-1)}$ has no values then $LOC_{(i-1)}$=0

Else LOC_i=$LOC_{(i-1)}$

Do for all the possible solutions

Ranking Algorithm + (LOC_i *P)

Rank the action based on the score got from the ranking algorithm

Repeat

Send the best rank parameters (problem cause, action) to the response unit

End Function

Calculating LOC:

Function QLearn (problem, cause, action, result)

If (result = = false)

Re-arrange the ranks. Give rank 1 to the new action

Send the previously selected action to the end of ranks list (last rank)

End if

Do for all actions in the list

Calculate LOC_i value for all the actions based on the rank list

Repeat

End Function

7.4.1 Ranking Function:

The proposed ranking function is Okapi BM25 [10][12], which is an information retrieval algorithm used in internet searches. This is used to search documents based on a key word or a search term. The ranking function is modified according to requirements. The modified ranking function considers the previously calculated level of confidence (LOC) value in selecting actions. This helps in giving priority for the actions that are considered for previously encountered problems.

The level of confidence values are initialized to 0. The ranking algorithm that is implemented is based on the following:

$$Score(D,Q)=\sum_{i=1}^{n} IDF(q_i).\frac{f(q_i,D)+(K_1+1)}{f(q_i,D)+K_1.(1-b+b.\frac{|D|}{avgdl})}+(LOC_i*P)$$

$$IDF(q_i)=\log\frac{N-n(q_i)+0.5}{n(q_i)+0.5}$$

Score (D, Q) = score/rank of the search terms in the document D

Q= $\{q_1, q_2 \ldots\}$ = list of search terms where q_i=Search term

$f(q_i, D)$ = frequency of the search term in the record D

K1 and b are free parameters and their values are taken as K_1=2.0 and b=0.75

|D| is the length of the record

avgdl is the average length of all the records in the document

LOC_i = previous Level of confidence for that action when considered earlier. It is calculated using Q-Learning algorithm.

P is the probability of the problem because of the cause. This is sent by recognition unit.

$IDF(q_i)$ is Inverse Document Frequency, calculated using the second equation shown above

N = total number of records in the document

$n(q_i)$ = total number of records that contain the search term q_i

Based on the above equation, the rank is decided for all the considered actions. Then the best action (rank 1) is sent to the recovery and response unit, suggesting that this action would solve the current failure.

7.4.2 Q-Learning Algorithm:

The Q-Learning Algorithm [8] [9] suits for the implementation of the self-learning or adaptive learning mechanism in the robot. This will help in updating the Knowledge Repository. This algorithm is implemented in the Analyzer module.

Q Learning algorithm comes under the Reinforcement learning process. This depends on a penalty-reward mechanism of our natural leaning process. This emphasizes the learning from interaction with the environment. This creates a tradeoff between exploration and exploitation. The agent (here robot) has to exploit the known things to get a reward and also to explore the better action sequence to maximize the reward. However, finally it should not fail at the task.

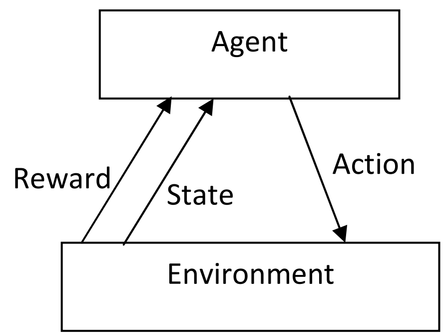

Fig 3: An agent action-state-reward diagram

In the above diagram

An agent (here robot) is at some state in the current environment. If it applies some selected action from the available action list, it will in turn receive a reward and a state transition in the same environment.

Algorithm:

For state $S=S_1$ to S_n

For action $a=a_1$ to a_m

Initialize Q'(S, a) = 0

End For;

End For;

Observe the current state S;

Repeat

Select an action a Є $\{a_1, a_2 \ldots\ldots a_m\}$ and execute it;

Receive an immediate reward r(S, a);

Observer the new state S'=(S, a);

Update: $Q'(S, a) = r(S, a) + \gamma \text{ Max } Q' (S', a')$;

S=S'

Forever

An agent always has option of selecting an action from the available actions list to go to a new state in the act of achieving the goal state. After performing any selected action, the agent receives a reward and state transition.

The following is the Q-Learning algorithm's notation-meaning table:

Notation	Meaning
S	State
A	Action
Q	Agent matrix
Q(S, a)	Agent moved to state S on action a
r(S, a)	Reward for moving to state S on action a
S'	New state
a'	New action
γ (gamma)	Discount factor

Table 5: Q-Learning algorithm notation-meaning table

This algorithm first initializes Q'(S, a) for a state in S and action a_i to 0. Then depending on the observed state, the algorithm selects an action from the given set of actions $\{a_1, a_2 \ldots a_m\}$. A reward r(S, a) is received based on the transition state. The values are updated and S is assigned to the new state S'. The algorithm continues looping until the agent calculates the LOC value for all the available actions.

CHAPTER VIII

IMPLEMENTATION AND RESULTS

In this chapter, we will discuss the implementation issues involved in the proposed model. A simulation tool was developed to validate the proposed activation unit's algorithm. The simulation model is used to measure the performance metrics such as the message overhead and learning activity in robots.

8.1 Framework Description:

This section gives in detail explanation about the considered scenario, environment, assumptions, instruction format and execution for simulation.

8.1.1 Scenario:

The simulation area is defined as a rectangle. This area comprises of a base station, a group of robots, and obstacles surrounded by a wall on all the four sides. The base station is responsible for sending instructions to all the robots. These instructions are sent one by one in a sequential manner to different robots. These instructions and the destination robots are generated randomly at the base station and the base station does not have any prior information about the environment in which the robots are moving. While the robots are moving, there is a possibility for the robots to get isolated from other robots and the base station. Apart from isolation, there could be message losses for robots. We have proposed an architecture based on the human body model which can detect and recover from failures.

8.1.2 Environment:

The simulation environment consists of base station, robots, obstacles and walls on all four sides. The Environment is assumed to be in a two dimensional co-ordinate system. The Base station and robots are considered as (x, y) points. Each robot moves in (x, y) co-ordinates. Obstacles are represented as lines with different orientations with co-ordinates (x1, y1) and (x2, y2). Walls are considered as borders for the environment.

8.1.3 Assumptions:

A total of 15 robots, 5 obstacles, and 1000 instructions are considered for the simulation. The number of robots, instructions and obstacles are simulation parameters that can be varied. The Base station is fixed at the center of the environment. Obstacles are stationary and have predefined positions. A failure is not considered when a robot stops by observing an obstacle in its path. A common radial communication range is predefined for robots and the base station. The communication range is also variable simulation parameter. Only the base station generates instructions for all the robots available in the environment. An instruction will be sent from the base station to a robot. After executing the current instruction, a robot receives another instruction. No parallel execution of instructions is considered for this simulation, as the base station needs to update the robot's new location after executing each instruction. At any given point only one way communication exists. This can be either from the base station to robot or from the robot to base station.

8.1.4 Instruction Format:

As mentioned in section 8.1.3, instructions are generated at the base station for every 2 virtual time units.

The Instruction format consists of 4 fields:

| Robot Id | Direction | Distance to move in units | Message id for that particular robot |

For example: 1R5M1 is an instruction for robot 1, to move right for 5 units with a message ID 1.

8.1.5 Instruction Execution:

After a robot receives an instruction, the robot checks for obstacles before moving every unit in the co-ordinate system till it executes the instruction or observes an obstacle its path. During the movement, if the robot encounters an obstacle or a wall, the robot stops at that position. The positions are updated at the base station either on successful instruction execution or on observing an obstacle or wall.

8.2 Addressed Failure:

A failure is defined as a situation where a robot could not perform the given task. Communication failure can be defined as the situation in which neither the robot can communicate with the base station or with the neighboring robots. In this simulation, communication failure could be due to

1) Robot Isolation: A robot is unable to communicate back to the base station by itself or through any other robots.

2) Message loss: This happens when a robot receives a message that is not in order because of an obstacle or unreachable position from base station.

8.2.1 Robot Isolation:

The communication will always take place either between a robot to the base station or from the base station to a robot. Consider that the base station sends an instruction to the robot; after executing the instruction the robot sends its updated position as acknowledgement back to the base station. Here the communication is from base station to robot. The robot waits for some time unit (say 30) after executing the instruction, and then checks its connection with the base station by sending a message. If it does not receive any acknowledgement back from the base station, the robot assumes that it is isolated.

8.2.2 Message Loss:

When a robot receives a message that is not in order due to message loss caused by the existence of an obstacle or in an unreachable position earlier from base station, then it is considered as message loss for that robot. For example, consider the robot R1 has executed the instruction, 1R5M1 that is sent by the base station. After sometime, it again receives an instruction, say 1L8M3, from the base station. The robot always checks the message id of current instruction with the instruction that has been executed and finds that message is not in sequence. This indicates that robot has lost a message. This may be due to presence of an obstacle or the robot is in unreachable position from the base station.

8.2.3 Obstacles:

The obstacles are predefined in the environment and are represented as lines with different orientations having co-ordinates (x1, y1) and (x2, y2). Obstacles can be present in any orientation within eight degrees of freedom. Walls are predefined boundaries in the environment and are also considered as obstacles.

Obstacles are addressed in the simulation as follows:

- When the base station tries to communicate with a robot, the presence of an obstacle might block the communication between them. In this case, a communication path will not be generated by the base station to the destination robot and the instruction will be pushed into the missed instruction list. Consider an example say base station generates an instruction 1R4M1. The robot R1 takes 4 virtual units to execute this instruction. Here the base station will wait for 4 units to expire before sending the next generated instruction for R1. Since the base station did not receive any acknowledgement from the robot R1, it pushes the instructions generated after 1R4M1 in to the missed instruction list. These instructions are the message loss to that robot.

- A variable waiting time is defined for each robot (say 30). Each robot checks its connection with the base station by sending an acknowledgement after the variable waiting time. If that robot could not receive the acknowledgment because of a communication breach due to the factors such as the presence of an obstacle, or absence of neighboring robots then this is considered as Robot Isolation.

- On detecting an obstacle in the robot's path, the robot stops at that point and does not proceed further. For example, when the base station sends an instruction 1R5M4 to the robot R1, the robot looks for obstacles before moving each unit. If it finds any obstacle ahead, it stops at that position. Obstacles are not considered as a failure while executing the instruction from the base station. The obstacles are considered as failure only when a robot checks its connection to the base station through its communication range but not the physical movement.

8.3 Integrated System:

The 3 main components of the integrated simulation system are

1. Recognition Unit
2. Activation Unit
3. Response and Recovery Unit

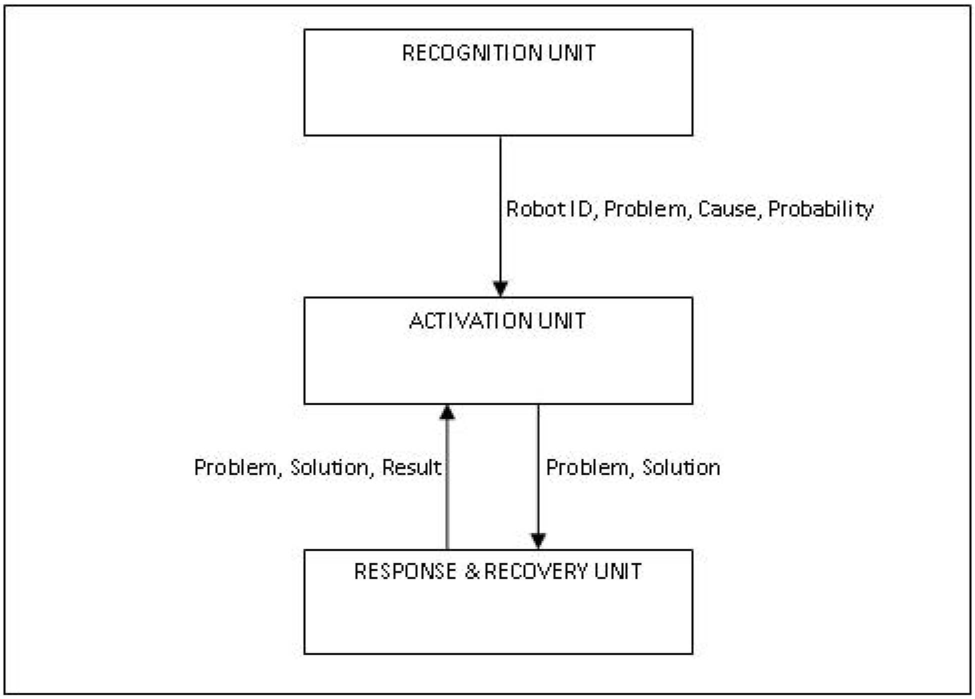

Fig 4: Data flow among the components

8.3.1 Recognition Unit:

The Recognition unit in a robot performs the task of identifying the failure and the cause of failure. Examples of such causes of communication failure are robot isolation, message loss, etc. Probability of cause of failure is calculated based on previous experiences. Failure factor is calculated for failures on a robot and the type of failure (permanent or temporary failure) is decided based on the failure factor value.

8.3.2 Activation Unit:

The activation unit is invoked by the recognition unit or by the response and recovery unit. The activation unit holds the knowledge repository that contains information about previously encountered problems and considered actions (See Table 4). The activation unit helps in robot learning mechanism.

The activation unit invokes the response and recovery Unit with a 3-tuple:

<*Problem, Cause, Action*> →Equation 1

The implemented activation unit diagram with respect to other two units is shown below.

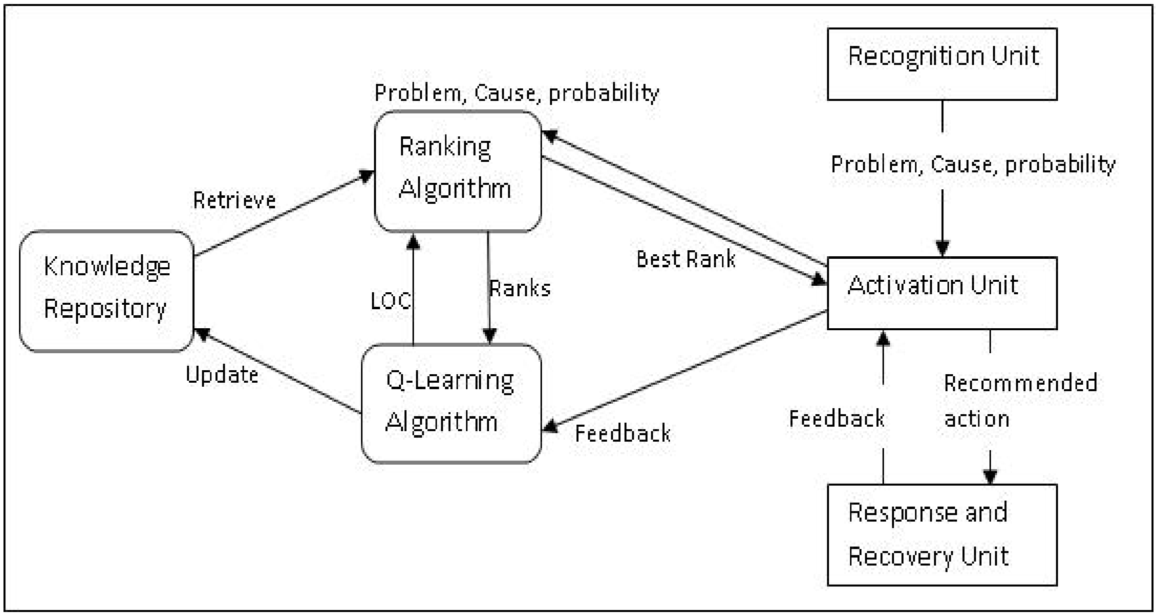

Fig 5: Integration of activation unit with other units

The recognition unit sends a message to the activation unit. The ranking algorithm in the activation unit analyzes the information and requests the action list from the knowledge repository for the encountered problem. The actions list is ranked using the ranking algorithm and the best of the available actions is sent to the response and recovery unit. After recovering from problem, the response and recovery unit sends feedback to the activation unit. The feedback from the response and recovery unit is passed to the Q-Learning algorithm to calculate the LOC values for the previously ranked action list. The calculated LOC values for the actions are updated in the repository and used for the future failure. The procedure repeats for every failure.

The detailed explanation of the format is sent to the response and recovery unit and the feedback from the response and recovery unit is shown below.

The example message format (equation 1) that is send from the activation unit to response and recovery unit will look like as follows:

For message loss: (Communication, message loss, Request Missed Message)

Or

For isolation: (Communication, isolation, Roll back)

The activation unit expects a feedback from the response and recovery unit after recovering the robot from a failure state. This feedback helps in the robot's learning process. The calculated values for actions are used by the robot if a failure occurs in future. The feedback format for the message from the response and recovery unit to the activation unit is a 4-tuple:

(Problem, Cause, Action Taken, Feasibility) →Equation 2

Ex1: Action from the activation unit to the response and recovery unit:

(communication, isolation, rolls back)

Feedback from the response and recovery unit to the activation unit:

(communication, isolation, Roll back, true)

Ex2: Action from the activation unit to the response and recovery unit:

(communication, isolation, missed message request)

Feedback from the response and recovery unit to the activation unit:

(communication, isolation, Roll back, false)

In the first example (Ex1), for the problem, the activation unit sent action is “Roll back”. The response and recovery unit executed that action and sends the *Feasibility* (equation 2) value as “true”, since the sent action solved the problem. But in the second example (Ex2), the response unit sends *Feasibility* (in Equation 2) as “false”. This means that the action sent by the activation unit did not solve the problem, so the response and recovery unit took “Roll back” action to solve the problem and made the result as false. The value “false” for the *Feasibility* parameter refers to the solution sent by the activation unit could not solve the problem and the response and recovery unit has taken a different action.

Based on the feedback the level of confidence values are calculated using the Q-Learning algorithm which helps in ranking.

If the *Feasibility* value is “true” in the feedback, the ranks are considered as it is and their level of confidence value is calculated.

If the *Feasibility* value is “false” in the feedback (refer Ex2 above), then the action taken by the response unit is given high priority (rank 1) and the previously selected incorrect action’s rank value is send to the end of all the ranks. Then the Q-Learning is implemented on the modified rankings.

If the *Feasibility* value is “true”, the virtual diagram for the ranked solution would look as follows:

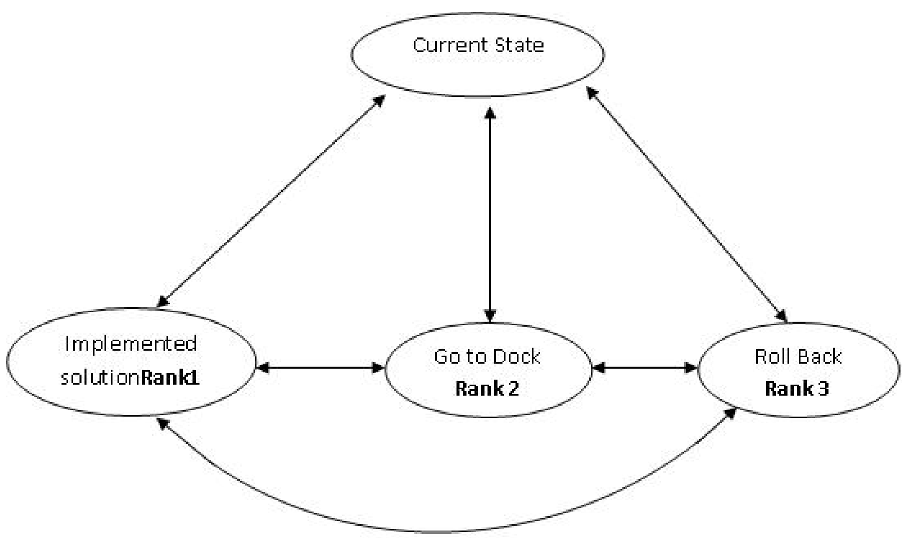

Fig 6: Internal virtual diagram in Q-Learning for calculated ranks using ranking algorithm

If the *Feasibility* value is “false”, the ranks are re-arranged and the level of confidence values of the corresponding ranks are calculated using QLearning.

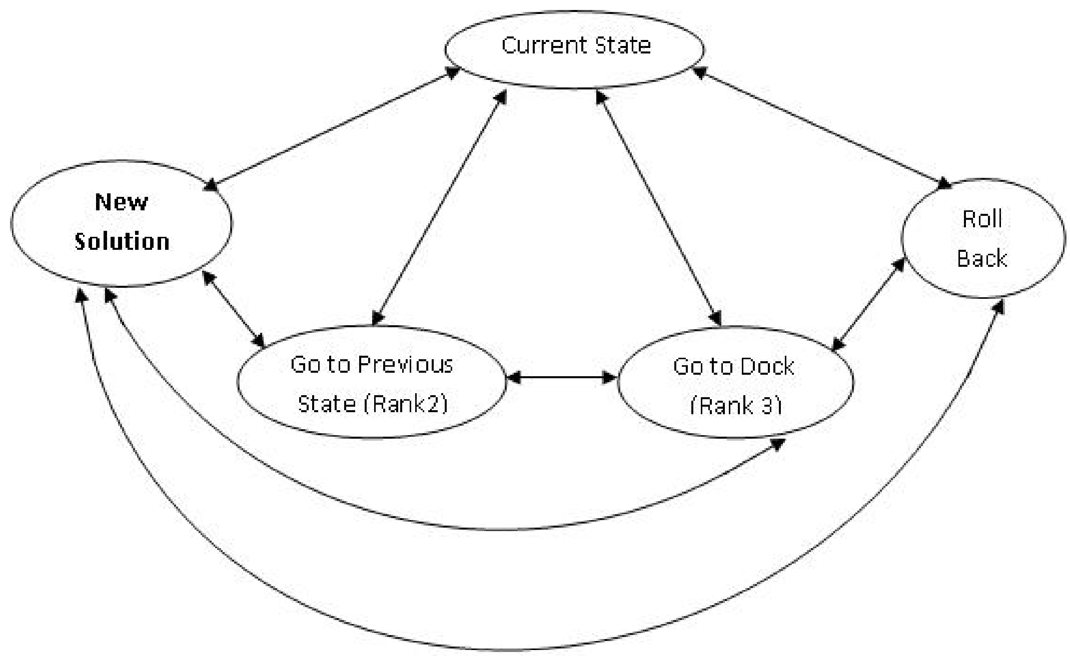

Fig 7: Modified internal virtual diagram for Q-Learning when a new solution is added

After modifying the internal virtual diagram for *Feasibility* value “false”, the LOC values are calculated using the Q-Learning algorithm (See Section7.4.2).

8.3.3 Response and Recovery Unit:

The Response and Recovery subsystem is responsible for recovering the robot from a failure. This unit stores the robot check points based on an adaptive window scheme. It also resets the adaptive window after implementing a recovery mechanism to get the robot back to a safe state. The response and recovery unit checks the feasibility of the action sent by the activation unit whether it helps to recover the robot or not and then implements the action, if it is feasible. If the action sent by the activation unit is not feasible, the response and recovery unit devices its own solution to recover the robot from failure. A feedback is sent to the activation unit after the recovery.

8.4 Results:

The simulation is run for 1000 instructions that are generated randomly at the base station. These instructions are sent from the base station to the robot depending on connection and obstacles. If a robot is not reachable from the base station, the instruction is pushed to a miss instruction list at the base station. The missed instructions were sent to the robot whenever the robot gets re-connected and sends a request to the base station. The robot's isolation is found by checking the connection back to the base station based on a robot variable waiting time counter (See Section 8.3.1). If there is no acknowledgement back from the base station, the robot assumes that it is isolated and executes the proposed three units (Recognition, Activation and Response and Recovery).

The results for the activation unit, which implements the ranking and Q-Learning algorithms, are shown in the graphs below. The ranking algorithm is used to rank the retrieved possible actions for failure from the knowledge repository. The Q-Leaning algorithm is used to improve the ranking algorithm by finding the level of confidence values to the actions selected by the ranking algorithm earlier through the feedback sent by response and recovery unit. These values are stored and used in future rank calculations for actions.

The graphs based on the results from simulation are:

1. Successful Solutions Vs Incorrect Solutions send by activation unit
2. No. of Failures and No. of incorrect solutions sent by activation unit
3. Message Overheads

8.4.1 Successful Vs Failure Solutions send by Activation Unit:

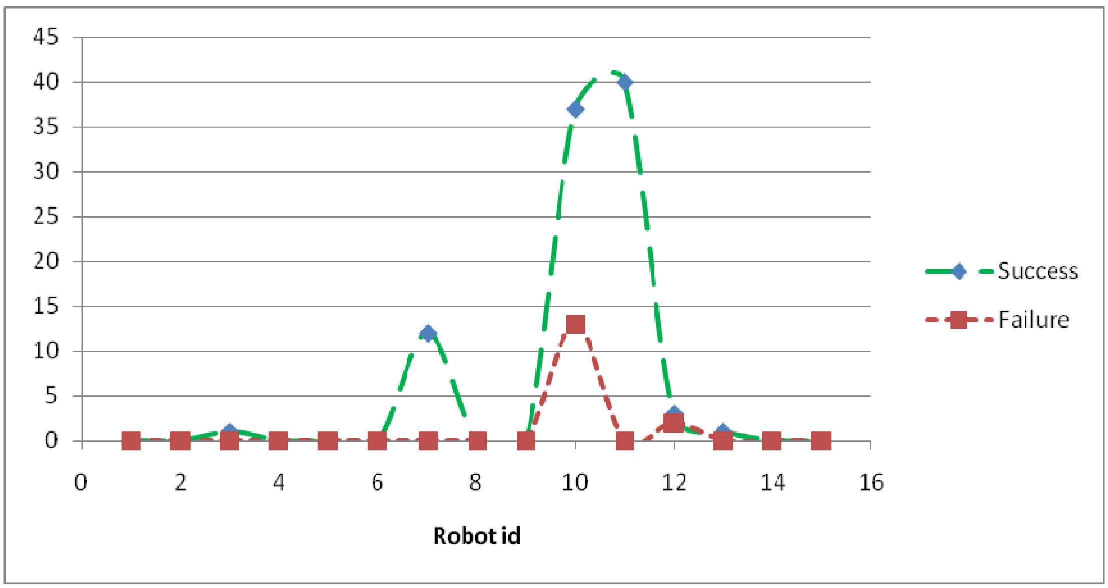

Fig 8: Graph on number of correct Vs incorrect actions from Activation Unit

In the above graph, the number of successes (actions sent by activation unit) is greater that number of failures (actions sent by activation unit). This shows that the robots are learning as the robot communication failures increases. Most of the time, the activation unit is sending the actions that help the robot to recover from the communication problem.

Consider robot id 10 in the above graph; the success rate of using the solution sent by the activation unit gives probable action for a problem and connection back with the base station. Robot 10 received fewer incorrect actions from activation unit. This means when there is a negative feedback from the response and recognition unit, the robot learns and updates its repository which helps in future action section.

8.4.2 No. of Failures and No. of incorrect solutions sent by Activation Unit:

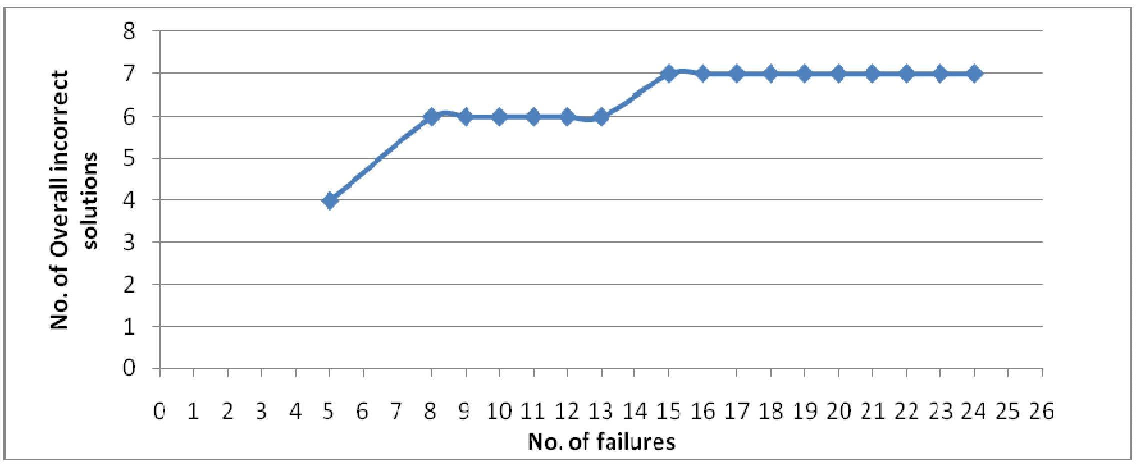

Fig 9: Graph on robot learning mechanism

The graph shown above supports the learning algorithm (Q-Learning) implemented for the activation unit. Here in the graph, x-axis is No. of failures and y-axis is No. of overall incorrect solutions send by the activation unit. The number of failures (on x-axis) refers to the failure count for the whole system.

Since the robot learns over time, the number of wrong actions sent by the activation unit tends towards a constant. The number of failures increases only when the activation unit sends a solution that could not solve the current problem. In this case, based on the feedback sent by the response and recovery unit, the learning algorithm updates the repository.

Consider that case when the number of failures is 5 and 8 on the x-axis. Here the line raise to 2 more points, this means that the learning happened because of the feedback.

Then from 8 till 13 the number of incorrect actions sent by activation unit is the same, which means the activation unit is sending a proper solution for the failure. The similar case occurs when the total number of failures is 13 and 15 on x-axis. The curve is raised, since the robot's response unit found again an incorrect solution and sends a negative feedback to the learning algorithm.

8.4.3 Message Overheads:

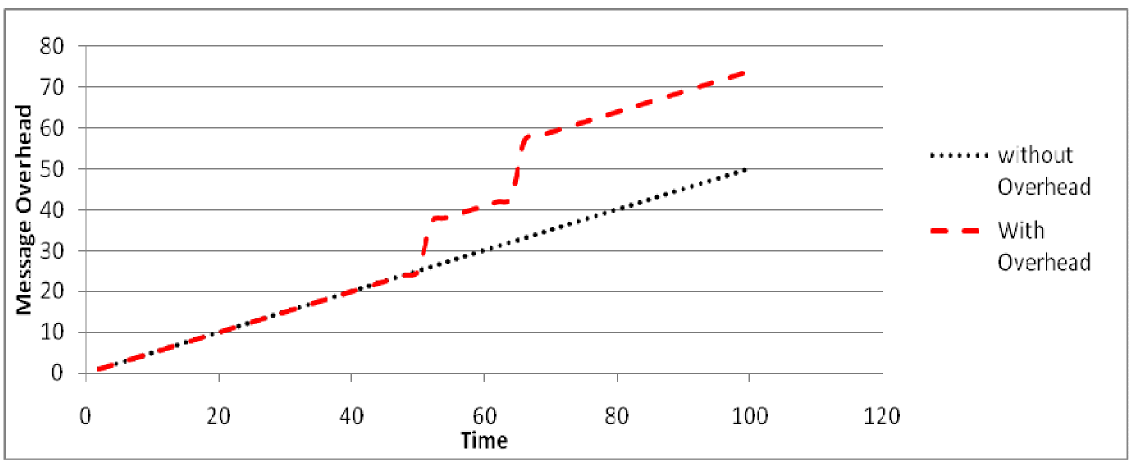

Fig 10: Graph on message overheads in implementation

The x-axis for the above graph is time and the y-axis is message overhead. The message overhead increases, since there is a communication check from a robot to the base station, in other words, it is testing for isolation. The increase or decrease of message overheads could be regulated using the variable waiting time for robot. If there is no instruction from base station to a robot till the waiting time, a robot checks its connection with the base station which costs 2 acknowledgements. Moreover, when a robot is rolling back to a previous position, it again checks its connection. So a total of 4 acknowledgements would exist for a robot to check its connection. The acknowledgement format would look like “<*robotid, connection*>”. Initially the connection value is “false” indicates that there is no connection with the base station. If the robot receives the acknowledgement with connection value as “true” from the base station, then there exists connection between the robot and the base station. If the waiting time is high for a robot, the message overhead could be reduced but there would be many isolated robots. If the waiting time for robots is low, the message overhead is high. By having an optimum threshold on waiting time, the system performance could be improved.

CHAPTER IX

CONCLUSION

In this thesis, we presented an approach to deal effectively with the failures that could occur while robot is executing in a hostile environment. Our overall goal is to propose an architecture for robots which helps in self detection, learning and recovery from a failure. The human immune system provides the basis for our model. The human immune system has a collection of cells which have a coordinated mechanism to protect the human body by identifying foreign bodies, killing them and preserving the information for future use. On a similar note the proposed architecture has three subsystems, namely, a recognition unit, an activation unit and a response and recovery unit which work together in detecting failures and recovering the robot to normal.

In this thesis, the activation unit to recommend actions for robot to recover from a failure based on ranking function and Q-Learning was developed. Simulation results show that the proposed architecture increases the overhead in terms of acknowledgement between base station and robots which is proportional to the number of instructions.

Future work can focus on developing a mechanism to reduce the message overhead. Advanced learning mechanisms could be implemented to improve the learning time in robots. This architecture could be extended to different areas of research such as computer security, intrusion detection, error analysis and many.

REFERENCES

[1] Chingtham Tejbanta Singh, and Shivashankar B. Nair, "An Artificial Immune System for a Multi Agent Robotics System", Proceedings of World Academy of Science, Engineering and Technology, pp. 308-311, June 2005.

[2] Hofmeyr S.A. and Forrest S., "Immunity by Design: An Artificial Immune System", Proceedings of GECCO Conference, pp. 1289-1296, 1999.

[3] Kim J. and Bentley P.J., "The Human Immune System and Network Intrusion Detection", Proceedings of 7th European Conference on Intelligent Techniques and Soft Computing (EUFIT), pp. 39, 1999.

[4] Harmon S.Y. and Gage D.W., "Current Technical Research Issues of Autonomous Robots Employed in Combat", Proceedings of 17th Annual Electronics and Aerospace Conference (EASCON), pp. 215-219, September 1984.

[5] Feng T., QinHua Deng, Zuren Feng and Ping Jiang, "Artificial Immune Network Based Robot Behavior Coordination under Wireless Network", Proceedings of 2nd International Conference on Pervasive Computing and Applications, pp. 72-77, July 2007.

[6] Richard Canham, Alexander H. Jackson and Andy Tyrrell, "Robot Error Detection Using an Artificial Immune System", Proceedings of the 2003 NASA/DoD Conference on Evolvable Hardware, page 209, 2003.

[7] de Castro, Leandro N. and Timmis J., Artificial Immune System: A New Computational Intelligence Approach, Springer-Verlag, 2002.

[8] Sutton, R. S. and Barto, A. G., "Reinforcement Learning: An Introduction". Cambridge, MA: MIT Press, 1998.

[9] Amit Konar, Computational Intelligence**:** Principles, Techniques and Applications, Springer-Verlag, 2005.

[10] Buttcher S, Clarke L.A.C., "Efficiency vs. Effectiveness in Terabyte-Scale Information Retrieval", University of Waterloo Technical Report, School of Computer Science, 2005.

[11] David K., Heather O., The Basic Principles of Wound Healing, http://www.pilonidal.org\pdfs\Principles-of-Wound-Healing.pdf [last accessed - Dec, 2008].

[12] Probabilistic Information Retrieval, http://nlp.stanford.edu/IR-book/html/htmledition/probabilistic-information-retrieval-1.html [last accessed - Dec, 2008].

VITA

LAKSHMI PRASAD BANALA

Master of Science

Thesis: ACTIVATION MECHANISM IN ROBOTS

Major Field: Computer Science

Biographical:

Personal Data:

Born on June 25th, 1984 in India

Education:

Completed Bachelor of Engineering in Computer Science and Engineering at Narayana Engineering College, Jawaharlal Nehru Technological University, Hyderabad, India in May, 2005.

Completed the requirements for the Master of Science in Computer Science at Oklahoma State University, Stillwater, Oklahoma in May, 2009.

Experience:

Student Worker, Student Union, OSU, Stillwater OK. 2006 - 2007.

Graduate Research Assistant, Spears S. School of Business, OSU, Stillwater, OK. 2007 - 2008.

Graduate Teaching Assistant, Computer Science Department, OSU, Stillwater OK. 2007 - 2008.

Name: LAKSHMI PRASAD BANALA Date of Degree: May, 2009

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: ACTIVATION MECHANISM IN ROBOTS

Pages in Study: 52 Candidate for the Degree of Master of Science

Major Field: Computer Science

In robotics, implementing strategies based on human body would help a robot to handle situations in a hostile environment where human intervention is not possible. Multiple robots may work in a coordinated manner to achieve certain tasks. One of the big problems is detection and recovery from failures, since human intervention may not be possible. To this end we propose an autonomic self-detection and self-recovery robotics architecture based on the human immune system. In this thesis, we look at self-detection and self-recovery of communications failure. In particular, we look at two types of communication failures; failures caused by robot isolation and failures caused by message loss. This thesis focuses on one component of the autonomic robotic architecture, namely, the activation mechanism in robots which make the robot respond to the communication failure that it had encountered during its operation by sending some suggested action. This is similar to the work done by the thymus and bone marrow (cell mature stations) in human immune system. The activation unit helps in storing, learning through experience and using the experience for future problems. It also learns through the feedback sent by another unit and uses the experience for future problems. The simulation results show that the proposed architecture helps robot in minimizing the failures by providing more probable actions to make the robot act dynamically.

ADVISER'S APPROVAL: Dr. Johnson P. Thomas