SELF DETECTION OF FAILURE IN ROBOTS

By

SURESH VENKATESAN

Bachelor of Engineering

Computer Science

Anna University

Chennai, 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

SELF DETECTION OF FAILURE 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 of all I would like to thank my parents and my sister Vijayalakshmi for their extended support, both financially and morally, which made it possible for me to come to the United States to pursue my graduate degree. Without their support, I would not have achieved this feat. I whole heartedly dedicate this thesis to them.

.

I express my gratitude to my graduate adviser Dr. Johnson P. Thomas. His inspiration, guidance and ability to clarify complex concepts enabled me to successfully complete my graduate degree in Computer Science. During my thesis writing period, he provided me with sound advice, good ideas and taught me the art of technical writing. The Department of Computer Science provided the support and equipment I needed to produce and complete my thesis.

I would like to thank Dr. Venkatesh Sarangan, Dr. Nohpill Park for their stimulating suggestions and valuable hints.

TABLE OF CONTENTS

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 tasks. Robots are being extensively used in a wide range of applications. They are deployed in demolition, fire fighting and bomb defusion, 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. Robots need to respond to such failures and they should be able to recover from the failure. Increasingly, more artificial intelligence is being added to enhance their thinking abilities. 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 learn, make decisions to react to different situations. The human body has multiple subsystems all working on its own, independently of each other all the time. The human immune system is one such system.

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 field of robotics to address the issue of failure in robots. Emulating the immune system in robots would form the basis for a robot to recover from attacks and failures. In this thesis, we propose a robotics architecture based on the human immune model to develop robots that can self detect failures and furthermore recover from the failure to a 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 it is not feasible, the recovery unit devises its own recovery action. The recovery unit then sends feedback to the activation unit. Based on the feedback, the activation unit learns and adapts thereby providing more probable and feasibly correct solutions for future problems. In this thesis we propose a self failure detection mechanism for robots to detect possible failure and determine the probability of cause for the failure. A mathematical model is proposed to determine whether robot’s failure is temporary or permanent.

Chapter 2 presents the literature review of the robotics, chapter 3 describes the problem under study, chapter 4 outlines the human body immune model, chapter 5 outlines the description about the proposed architecture, chapter 6 correlates the proposed architecture with the human immune model, chapter 7 describes the proposed solution, chapter 8 presents the simulation, results and chapter 9 concludes the thesis.

CHAPTER II

REVIEW OF LITERATURE

In this chapter we review previous work in the area of fault detection in robots. D.H Barnhard, J.T.McClain, B.J Wimpey and W.D.Potter [1] proposed a system that uses bluetooth communication to guide the robots to a specific target in the environment. The target is a source of light. Guide robot contains a light sensor that can be used to find the target. First the guide uses the greedy search and scans for the regions where there is a high intensity of light and then takes its path towards it.

When the guide robot moves, the light sensor takes periodic measurement of light intensity to make sure that the robot is on a correct path. If the intensity of light decreases the robot makes a turn and once again scans for a higher intensity. Once it reaches the target it tries to identify its location by means of the coordinate points. Then these coordinate points are sent to blind robot via bluetooth radio link. Once the blind robot receives the coordinate points it starts to adjust its position along the vector coordinate till it matches the distance supplied by the guiding robot.

Stergios I.Roumelioutis, Gaurav S. Sukhatme and George A.Bekey [2] proposed a method to detect faults in mobile robots. Adaptive estimation was used to predict the outcome of faults. The system behavior of each faults were embedded into different Kalman Filters that are set to a particular fault. Each of the filters gives the predicted sensor values. These values are compared to the actual sensor readings. The difference between them gives the performance of the filter. This method was implemented in a Pioneer robot.

Christian Plagemann, Dieter Fox, Wolfram Burgard [3] proposed a model which uses a Gaussian process, classification and regression techniques for studying the distributions of a filter which were used to find the particular state of the system. This method was implemented on robot and the proposed system was able to detect collision with obstacles.

Christian Plagemann, Cyrill Stachniss, Wolfram Burgard [4] dealt with the problem of isolation and fault detection of mobile robots. They proposed a system which used mixed abstraction particle filter to effectively handle the failures in robot. Explicit assumptions were made and a model abstraction hierarchy was built. This model increased the efficiency of the systems thereby reducing the computational load on it.

C Foulston, A. Clare [5] proposed a framework for grid based failure detection system for a robot system. The basic features are agent base monitoring, selective reporting and report dispatch monitoring. Reporting was done by means of e-mail, text messaging and PC alerts and the report dispatching allows humans and agents to sign up for latest reports for further analysis. It uses the concept of detecting and recording errors which are very important for a complex system.

Roger L. King, Aric B. Lambert, Samuel H. Russ, Donna S. Reese [6] described the human immune system and its functionalities from a computational viewpoint. Their work gives a brief way in which the biological systems can be studied and the inferences made can be used in intelligent systems.

Sudha chinni, Johnson Thomas, Gheorgita Ghinea, Zhengming Shen [7] proposed a model that allows trust to be built over time, when the number of interactions between the nodes increases. Trust also includes the quality of service it provides. Bayesian networks form the basis for this model.

In the past, research has been done in the form of Bayesian Inference, filters, report dispatch monitoring and so on. Using observation graphs to detect a failure is a different approach. Our approach emphasizes on detecting the failure and sends the failure information to the other subsystem that recommends a recovery action.

CHAPTER III

PROBLEM DESCRIPTION

Typically robots are employed to work in a hostile environment where human intervention is not possible. 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 a 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 a 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 the Base station. When a robot encounters a communication failure it should use some detection mechanism to detect possible failure, cause of the failure and recover back to a safe state so that normal operation can resume.

CHAPTER IV

HUMAN BODY IMMUNE MODEL

The Human body has multiple subsystems that work independently. In case of any 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. There are two types of immunity. One is innate and the other is adaptive. Innate immunity is a non-specific immunity. The response provided by this immunity is antigen-independent. Adaptive immunity is a learning process inherited in the human body and the response provided is antigen-dependent.

The main components of human immune system are WBC (White Blood Cells), fibroblasts and blood platelets [8]. WBC plays an important role in the immune system by providing the necessary defense 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 principle components of the immune system that are present in WBC. Lymphocytes are constituted of T-cells and B-cells. 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 circulate in the blood stream all through the body. They play an important role in detecting foreign antigens or foreign behaviors. B-cells produce antibodies (protein structures) for an antigen.

The macrophages of WBC's are located on the surface of the body cells. Whenever any foreign body comes in contact with the human body cells, the macrophages engulfs the foreign body and decompose them to release their amino acids. T-cells differentiate the foreign body by comparing the chemical structure of the self cells with the foreign body amino acids. Then T-cells alarm the other cells by releasing chemical substance. This chemical substance activates the T4 killer cells and B-cells in the blood stream. T4 Killer cells weaken the amino acid structure of the foreign body and the B-cells produce unlimited number of antibodies that kill the foreign body cells.

The excess antibodies that remain after the defense process 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. These memory cells help in mounting a strong attack next time if the same antigen invades.

Wound (internal or external) healing process 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 [9]. 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. The final wound covering comes under remodeling phase.

After finishing all the stages and killing the foreign body cells, the dead cells are cleared up by the scavenger macrophages to rehabilitate normal body condition.

CHAPTER V

PROPOSED APPROACH

The proposed immune system in a robot is shown below:

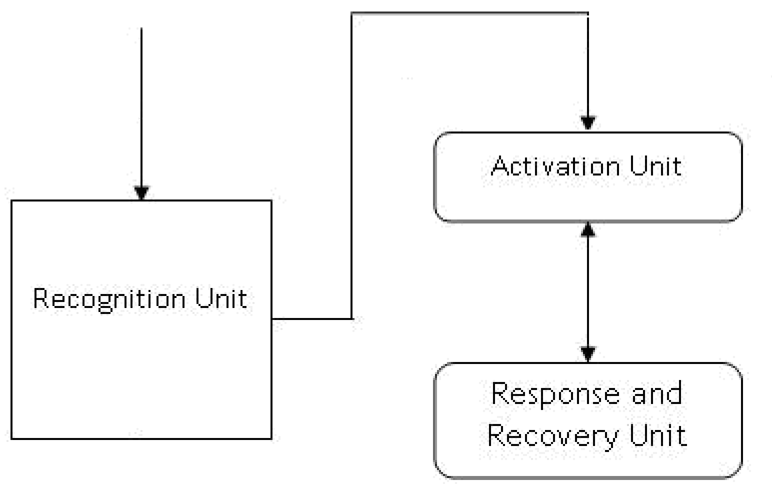

Fig. 1: Robot Immune System

5.1 Recognition Unit:

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 out of the defined environment. 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 the nature of failure.

5.2 Activation Unit:

The Activation unit is responsible for providing a more probable solution to the encountered problems and updating the knowledge repository. The knowledge repository stores a list of previously encountered problems and corresponding recovery action taken. When the activation unit receives the failure information from the recognition unit, it sends a recovery action to the response and recovery unit. It uses the feedback from the response and recovery unit to update its repository.

5.3 Response and Recovery Unit:

The final unit is the response and recovery unit. This is responsible for taking action 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 sent by the activation unit is not feasible, the response and recovery unit devises its own recovery action. After taking the necessary action, it sends a feedback to the activation unit.

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. 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.	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. Release chemicals when an antigen is detected which signals the other cells for further action against the foreign body.	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 the 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 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. 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 store 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 the proposed Activation Unit.

The activation unit recommends an action to the response unit to bring the robot back to 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 table 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. 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.	1. Executes the recovery mechanism to prevent further failure.
2. The surrounding cells come to aid the damaged cell and provide some kind of defense mechanism against infections.	2. Receives action from activation unit and implements its own failure checking conditions with the recommended action to act against failure.
3. Fibroblasts cells are used to remodel the tissues	3. Response unit make sure that robot resumes to normal execution.

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

The Scavenger Macrophages clean 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. The table 3 shown above gives the correlation of human body recovery with the proposed response and recovery unit.

CHAPTER VII

PROPOSED SOLUTION

We propose a self failure detection mechanism for robots to detect possible failure and determine the probability of cause for the failure. A mathematical model is proposed to determine whether a robot's failure is temporary or permanent based on the failure factor. Changes in observation graphs are used to detect a failure. Once a failure is detected the probability of cause for a failure is calculated.

7.1 Algorithm:

The steps in our approach are:

Step1: Determine the Initial Observation Graph

Determine the observations needed to detect the cause for the failure in robot. These observations capture the state of the robot. A state also includes relational information such as the position of the robots and other objects such as the base station. A Bottom up approach is used in generating the observation graph from observations. An Observation graph is hierarchical with 4 levels. The first level represents the object under consideration, the second level represents the nature of failure, the third level represents the cause of failure, and fourth level represents the observations made of a robot. There exists an AND/OR function on the observation node.

There could be more than one observation for a cause. In a similar way there is an OR function on the cause node. If one of the causes is true it is said to be a failure. An external input such as a command from a base station will result in a new state that is independent of the previous state. The initial observation graph is defined as a graph model G_I

Definition: The state of a robot is modeled as a graph $G = (V, E)$ where:

- V is a set of nodes $V = V_{ID} \cup V_F \cup V_C \cup V_O$ where
    - V_{ID} is the root ID node of the graph
    - V_F is the a node representing the nature of failure
    - V_C is a set of cause nodes which specify the cause of failure.
    - V_O is a set of observation nodes specifying the observations that are made of a robot
    - For nodes Vc, V_F, $V_O \in V$ *there* exists a function *Cond:* $e \rightarrow$ Boolean where $e \in \{Vc \cup V_F \cup V_O\}$. The Boolean on the node indicates whether a value associated with the node is true
- E is a set of directed edges such that

  $E = E_I \cup E_C \cup E_O$ where $E_I = (V_F \times V_{ID})$, $Ec = (V_C \times V_F)$

  $Eo = (V_O \times V_C)$

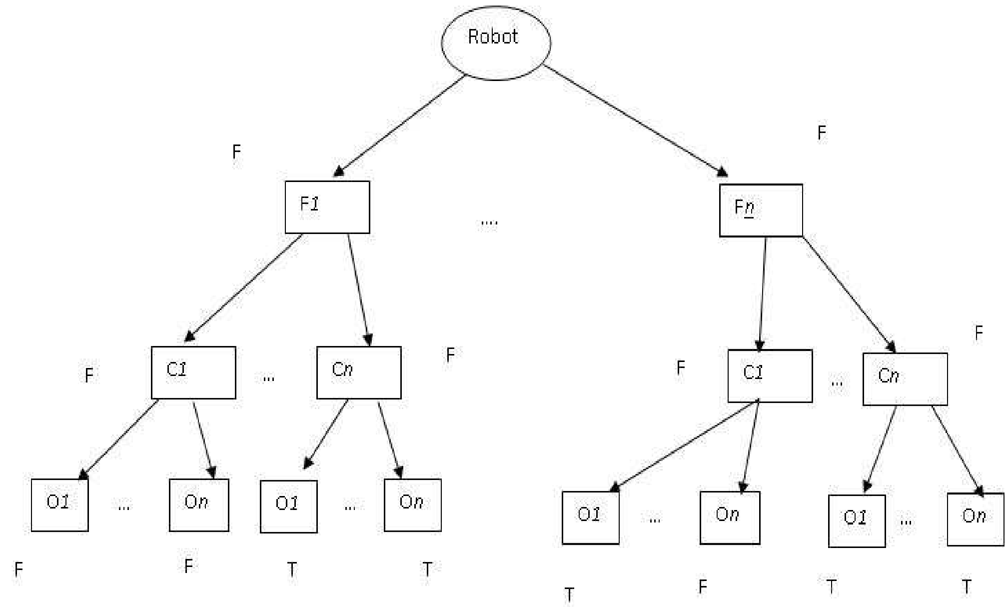

Fig 2: Initial observation graph

O*n* represents observations made of a robot

C*n* represents the cause of failure

F*n* represents the nature of failure

T , F represents True , False respectively

Initially we assume that there are no failures.

Step 2: Determine the Current Observation Graph

Derive the new state as a result of an external input such as executing an instruction received from the base station. Represent the new state as a graphical model G_N.

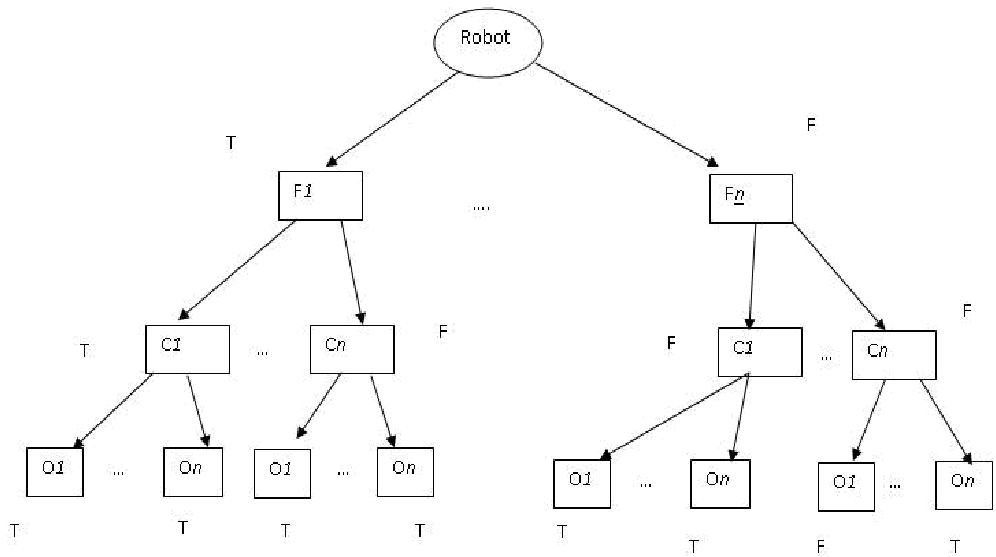

Fig 3: Current observation graph

Step 3: Compare the two Observation Graphs

Compare the Graphs G_I and G_N in terms of the boolean values on the identical nodes in V_F and V_C. The nodes V_F , V_C whose boolean values differ on the two observation graphs represent the nature of failure and cause of failure respectively.

Step 4: Determine the Probability for Cause of Failure

Bayes theorem is used to calculate the probability of cause of failure for a robot given a failure.

Bayes theorem $P(h \mid e) = \dfrac{P(e \mid h) * P(h)}{P(e)}$

Where

P (e) is the prior probability of evidence

P (h) is the prior probability of hypothesis

And P (h | e) is the probability of h given e.

Calculation of probability takes earlier experiences into consideration.

Failure is taken as evidence and the probability of cause for failure is calculated.

$$P(Cause \mid Fail = 1) = \frac{P(Fail = 1 \mid Cause) * P(Cause)}{P(Fail = 1)} = Fcause \rightarrow \quad \text{Equation (1)}$$

Calculating individual components in Equation (1)

Probability of a failure given a cause is

$P(Fail = 1 \mid Cause) = 1$

Probability of isolation

$P(Cause) = \dfrac{Ccause}{C_{Total}}$ And

Probability of failure

$P(Fail = 1) = \dfrac{C_{Fail}}{C_{Total}}$

Therefore $F_{cause} = \frac{Ccause}{C_{Fail}}$ → Equation (2)

Where

C_{Cause}= Number of time robot failed due to cause

C_{Fail} = Number of times robot has failed (Includes all possible failures)

C_{Total} = Total number of instructions the robot has received

Each robot maintains a table which has these values. Corresponding entries in the table are updated each time the robot fails.

Step 5: Send failure information to the Activation Unit

Once a failure is detected the recognition unit sends failure information to the activation unit for further action. The information sent is 3-Tuple.

<Failure, Cause, Probability of cause for failure>

Failure is defined by the node V_F whose value is true and cause is defined by the node *Vc* whose value is true and probability is defined by equation (1).

Step 6: Failure Factor

Failure factor is calculated to find whether the failure for a robot is permanent or temporary over a period of time. For every failure in each robot, the failure factor is calculated.

Failure factor is defined as

$$FF = (Wcause * Fcause)/(Tcurr - Told)$$ → Equation (3)

Where,

W*cause* is the weight of cause usually a value between (0, 1)

F*cause* is the probability of cause for the given robot

T*curr* is the current time unit when the robot has failed.

T*old* is the previous time unit when the robot had the same type of failure.

While calculating the failure factor of each failure in a robot the previous value is taken into consideration.

Cumulative failure factor $_{n} = \alpha\ FF_{n} + (1-\alpha)\ FF_{n-1}$

Where

α - Constant that takes higher value for the current failure factor and smaller value for the old failure factor $(1-\alpha)$.

Trials are run on the robot and the average failure factor is set as the threshold. When the robot's failure factor is above the threshold for a period of time it is considered as a permanent failure, else it is considered as a temporary failure.

CHAPTER VIII

IMPLEMENTATION AND RESULTS

We have proposed an algorithm to self-detect failure in robot when they get isolated from the base station and the other robots. A simulation tool was developed to validate the proposed algorithm. The simulation model is used to measure performance metrics such as total number of failures in a robot, message overhead and the number of messages lost by each robot.

8.1 Framework Description

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 to 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 a 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 common radial communication range is predefined for robots and the base station. The communication range is also a variable simulation parameter. Instructions 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 one way communication exists. This can be either from the base station to robot or from the robot to base station. If the robot stops moving, on observing an obstacle it is not considered as a failure.

8.1.4 Instruction Format

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 in 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

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 it is considered as a 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 due to isolation or obstacles.

8.3 Obstacles

Obstacles are predefined 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 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 the instruction generated by the base station 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 to the missed instruction list since there is no path to that robot from the base station. These instructions are the message loss to that robot.

- The Robot looks for a connection back to the base station whenever it exceeds the waiting time. If the robot could not transmit the acknowledgment to the base station because of a communication breach due to the factors such as the presence of an obstacle, or absence of neighboring robots, then it is considered to be isolated.

- On detecting an obstacle in the robot's path, the robot stops at that point and does not move 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 an obstacle ahead, it stops at that position. Obstacles are not considered as a failure if they are physically in the way. Obstacles are considered as failure if there are two robots that try to communicate and there is an obstacle between them.

8.4 Integrated System

The 3 main components of the integrated system are

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

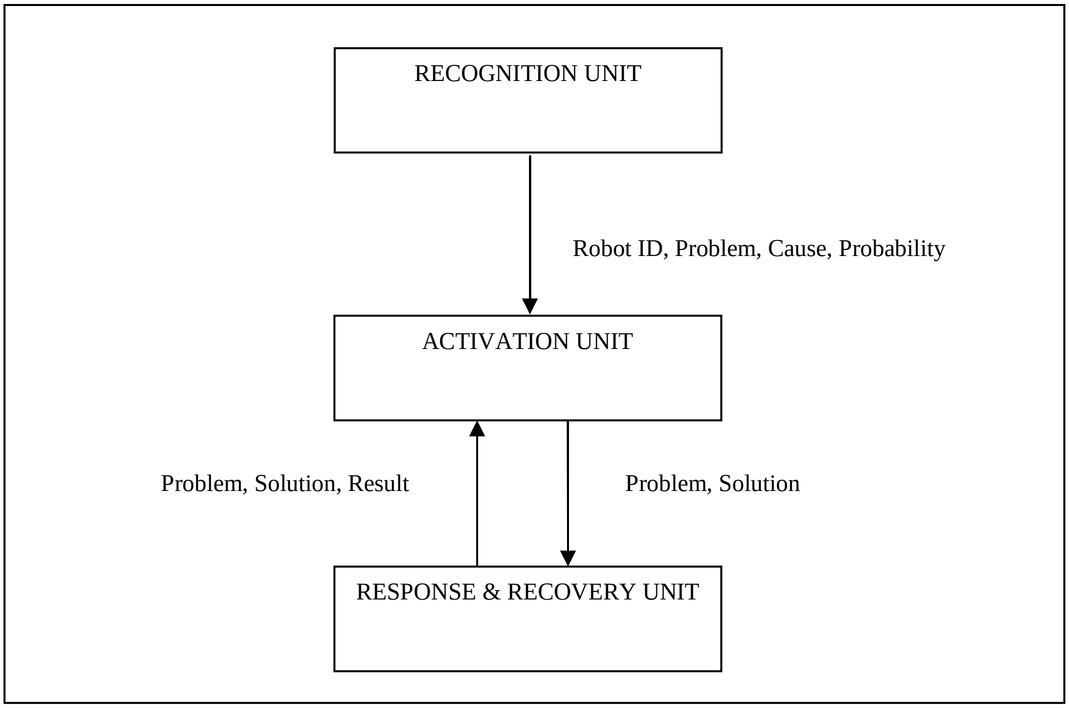

Fig 4: Data flow among the components

8.4.1 Recognition Unit

The Recognition subsystem 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 for the failure is calculated based on previous experiences. This unit is independent of the other units as it keeps checking continuously for failures. Failure factor is calculated for failures on a robot and the type of failure (permanent or temporary failure) is decided based on the values of the factor.

Observations:

The observations that are considered for isolation are as follows

1. Path back to the Base station
2. Obstacle in path
3. Waiting time

The observation that is considered for message loss is as follows

1. Message sequence

Initial Observation Graph:

We simulated communication failure due to isolation or message loss. When the robots are initialized in the environment initial observation graphs are determined for each robot in the system. We assume that there are no failures initially. With these observations a bottom up approach is used to generate the graph. (Refer section 7.1)

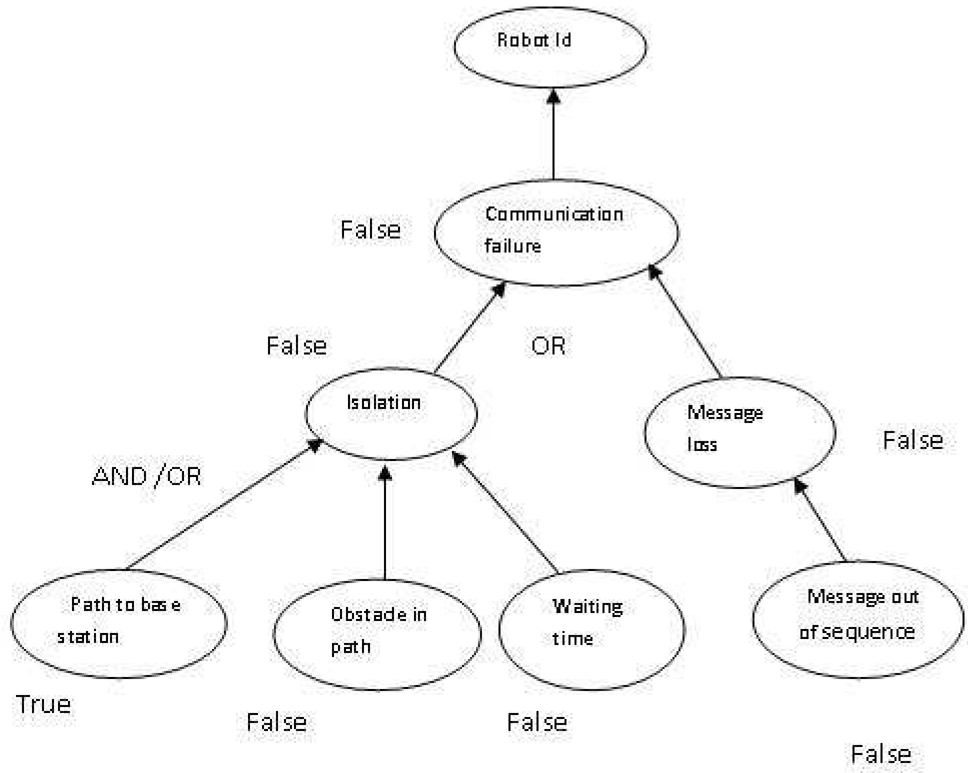

Fig: 5 Initial Observation graph for communication failure

Current observation graph:

For every instruction from the base station to the robot, the observations are made for message loss and observations are made for isolation whenever the robot exceeds the waiting time. These observations are used to complete the graph in a bottom-up approach. When the waiting time has exceeded and there is no clean path (without obstacles) to the base station it is considered to be isolated. When the message received by the robot is out of sequence it is considered as message loss.

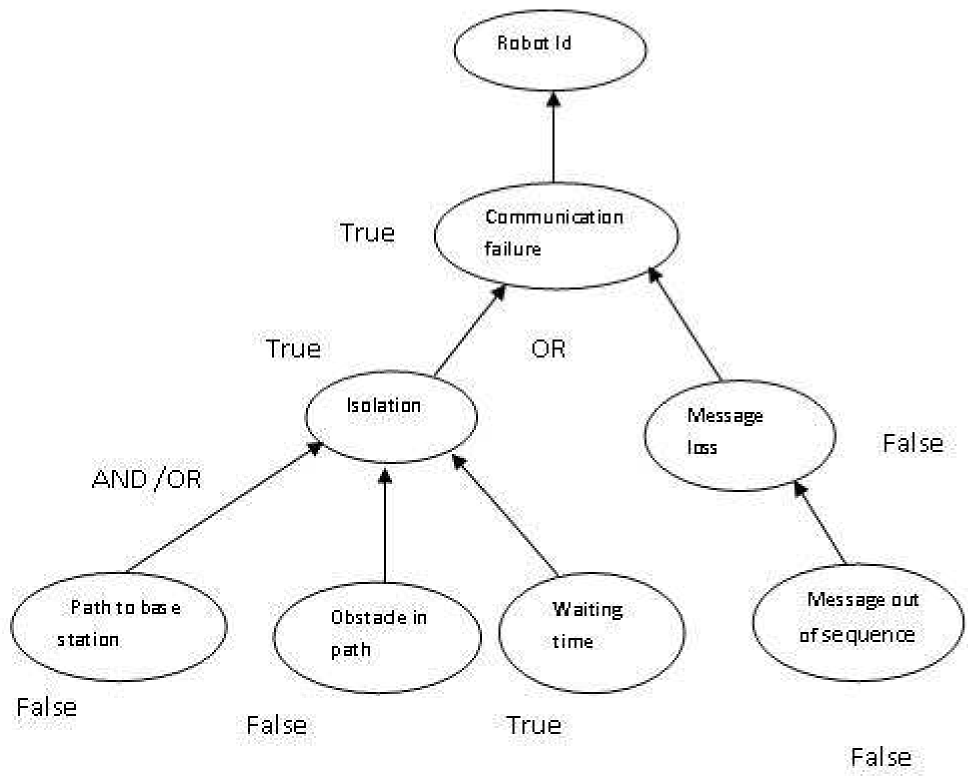

Fig: 6 Current observation graph for communication failure

The initial and the current observation graphs are compared. When there is a difference in the boolean value on the node representing the failure in the two graphs, communication failure is detected. The node V_F represents the nature of failure.

This unit calls the activation unit as a 3-tuple format.

(Problem, Cause, Probability of cause of failure) → Equation (4)

For example it sends

(Communication failure, Isolation, 0.62)

The probability of cause of failure is calculated using Bayes theorem which uses previous experiences for that robot.

$$P(Cause = "Isol" | Fail = 1) = \frac{P(Fail = 1 | Cause = "Isol") * P(Cause = "Isol")}{P(Fail = 1)} = F_{Isol}$$

Here the failure of the robot is taken as evidence and the probability that the robot is isolated is calculated.

$P(Fail = 1 | Cause = "Isol") = 1$

Give the robot is isolated the probability that the robot failed is 1.

Probability of isolation

$P(Cause = "Isol") = \frac{c_{isol}}{c_{Total}}$ And

Probability of failure

$P(Fail = 1) = \frac{c_{Fail}}{c_{Total}}$

Therefore F*isol* $= \frac{c_{isol}}{c_{Fail}}$ → Equation (5)

Where

C_{isol} = *Number of time robot failed due to isolation = 5*

C_{Fail} = *Number of times robot has failed (Includes all possible failures) = 8*

C_{Total} = *Total number of instructions the robot has received =15*

Therefore the probability is 5 /8 = 0.62

This probability is used in calculating the failure factor for a robot.

The failure factor is calculated as follows:

$$\text{Failure Factor} = (W_{cause} * F_{cause})/ (T_{curr}-T_{old}) \quad \rightarrow \text{Equation (6)}$$

For example let's say the robot R1 has failed due to isolation at time 200 and 400 virtual units and its corresponding probability of isolation are 0.4 , 0.2.

Wcause is given a value 1.

Failure factor when the robot fails first time = (1 * 0.4) / (200 - 0) = 0.002 Cumulative value are used to calculate when the robot fails more than once.

Cumulative failure factor $_n = \alpha$ FF $_n + (1-\alpha)$ FF $_{n-1}$

Where α = .6

Current Failure factor = (1*0.2) / (400 -200) = 0.001

Failure factor when the robot fails the second time= .6 (0.001) +.4 (0.002) = 0.0014

8.4.2 Activation Unit

The activation unit is invoked by the recognition unit or by response and recovery unit. The activation unit holds the knowledge repository that contains information about previously encountered problems and considered actions. The activation unit expects a feedback from the response and recovery unit after recovering the robot from failure state. This feedback helps in robot's learning process.

8.4.3 Response & Recovery Unit

The Response and Recovery subsystem is responsible for recovering from the failure. This unit stores robot check points based on adaptive window scheme. Also it resets the adaptive window after implementing a recovery scheme to get the robot back to safe state. The response and recovery unit checks for feasibility of the action that is sent by the activation unit. It executes the action if it is feasible. If the action recommended by the activation unit is not feasible, the response and recovery unit devises its own action and sends the feedback to the activation unit.

8.5 Results**:**

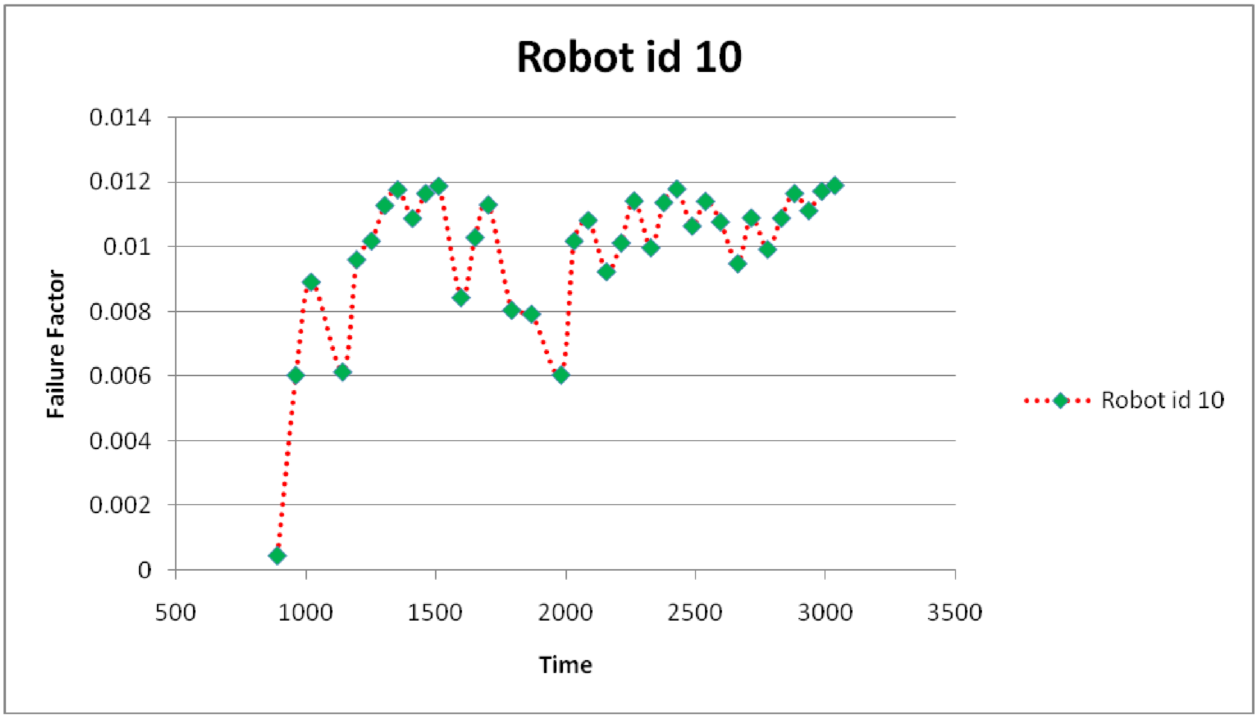

Fig 7: Graph on Time Vs Failure factor for Robot id 10

A Graph is plotted with time on the X-axis and failure factor on the Y-axis. Failure factor is calculated for each failure on the robot. It can be seen from the graph that the failure factor increases whenever the robot fails frequently. A threshold is set to distinguish whether the type of failure is permanent or temporary. Consider the threshold value to be 0.01 (Refer section 7.1). On time (1200 -1500), (2100- 3200) in the X-axis we can see that the failure factor is above the set threshold most of the time which is an indication that the robot has been failing very often. In this case the robot's failure can be considered permanent.

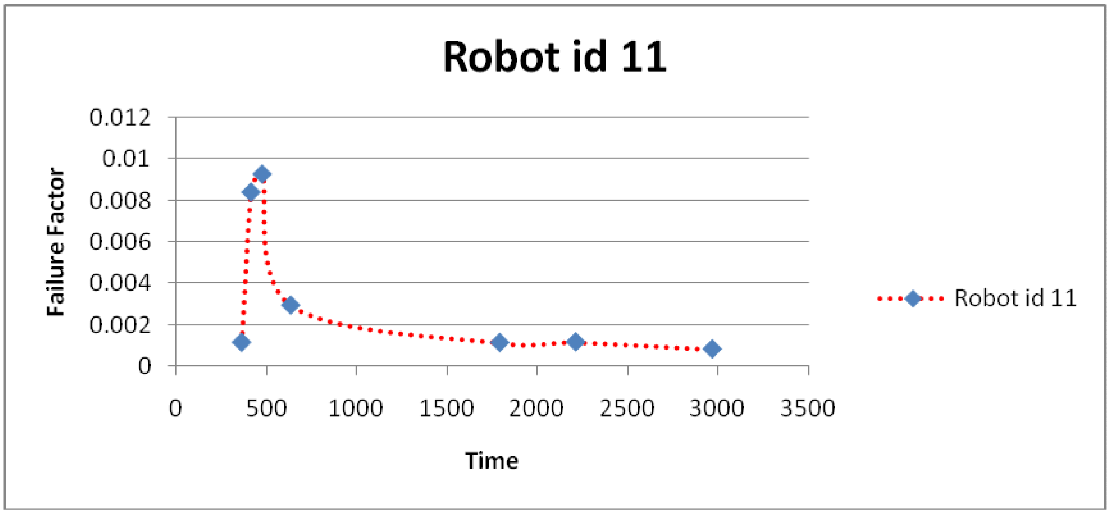

Fig 8: Graph on Time Vs Failure factor for Robot id 11

A Graph is plotted with time in the X-axis and failure factor in the Y-axis for Robot id 11. It could be seen that the failure factor keeps on decreasing over the set threshold (0.01). From the graph we can infer that the robot is failing very rarely. The failure for this robot can be considered as temporary.

Proposed system vs. Existing system:

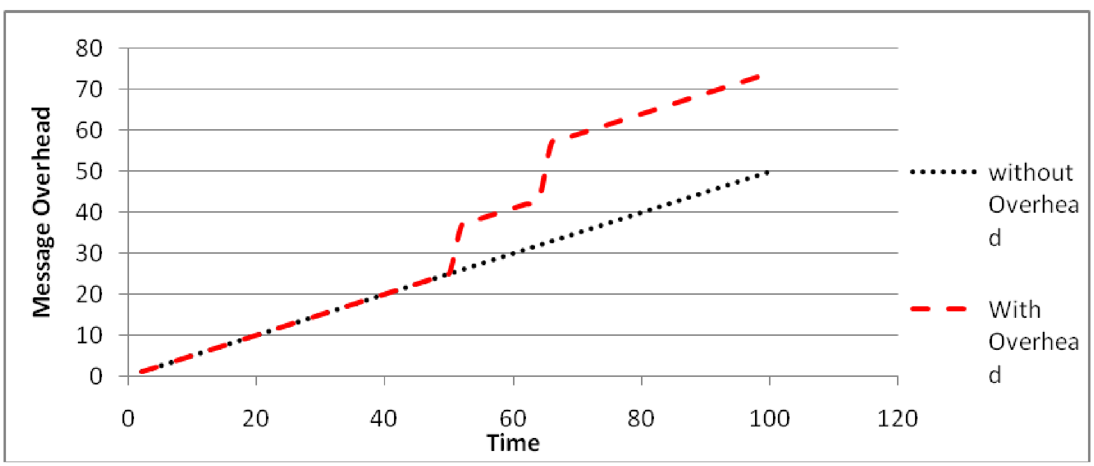

Fig 9: Graph on message overheads in implementation

The x-axis for above graph is time and y-axis is message overhead. By using the proposed model, the overhead increases, since there is communication back from robot to base station when the waiting time expires. If there is no instruction from the base station to a robot till the waiting time expires, the robot checks for a connection back with the base station which costs 2 acknowledgements. Also when a robot rolls back to the previous position it checks for a connection back to the base station .Therefore there would be a total of 4 acknowledgements needed. . The increase or decrease of message overheads could be regulated using variable waiting time for the robot.

CHAPTER IX

CONCLUSION

Our overall goal was to propose an autonomous architecture for robots which helps in self detection and recovery from a failure. The Human immune system has been the source of inspiration for this model. The Human immune system has a collection of cells which have a coordinated mechanism to protect the human body by identifying the 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 a normal state. In this thesis a recognition unit based on the human body model has been proposed and simulated. The proposed architecture increases the overhead in terms of acknowledgement between base station and robots. Future work may focus on implementing an effective mechanism to reduce the message overhead. This architecture could be extended to different areas of research such as computer security, intrusion detection, error analysis and so on.

REFERENCES

[1] D.H Barnhard, J.T.McClain, B.J Wimpey and W.D.Potter, “Odin and Hodur: Using Bluetooth Communication for coordinated robotic search”, Proceedings of the International Conference on Artificial Intelligence, pp. 365-371, 2004

[2] Stergios I.Roumelioutis, Gaurav S. Sukhatme and George A.Bekey, “Fault Detection and Identification in a mobile robot using multiple-model estimation” Proceedings of the International Conference on Robotics and Automation, pp. 2223-2228, 1998

[3] Christian Plagemann, Dieter Foxx, Wolfram Burgard, “Efficient failure detection On mobile Robots using particle filters with Gaussian process proposals” Proceedings of the International Joint Conference on Artificial Intelligence, pp 2185-2190, 2007.

[4] Christian Plagemann, cyrill stachniss, wolfram Burgard, “Efficient failure Detection for Mobile Robots using mixed abstraction particle filters”, Proceedings of the European Robotics Symposium, pp. 93-107, 2006

[5] C Foulston, A. Clare, “A framework for Grid-based failure detection in an Automated Laboratory Robot system”, Proceedings UK e-Science Poster in All Hands Meeting, 2006

[6] Roger L. King, Aric B. Lambert, Samuel H. Russ, Donna S. Reese, “The Biological Basis of The Immune System as a Model for Intelligent Agents” Proceedings of Symposium on Parallel and Distributed Processing Workshops of International Parallel Processing Symposium , pp. 156-164, 1999

[7] Sudha chinni, Johnson Thomas, Gheorghita Ghinea, Zhengming Shen “Trust model for certificate revocation in ad hoc networks”, Ad Hoc Networks Journal, Volume 6, Issue 3, pp. 441-457, May 2008

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

[9] Keast David, Orsted Heather, The Basic principles of wound healing, http://www.pilonidal.org/pdfs/Principles-of-Wound-Healing.pdf [last accessed - Dec, 2008]

VITA

SURESH VENKATESAN

Master of Science

Thesis: SELF DETECTION OF FAILURE IN ROBOTS

Major Field: Computer Science

Biographical:

Personal Data:

Born on July 31, 1984 in India

Education:

Completed Bachelor of Engineering in Computer Science at Bharath Institute of Science and Technology, Anna University, Chennai, 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:

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

Name: SURESH VENKATESAN Date of Degree: May, 2009

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: SELF DETECTION OF FAILURE IN ROBOTS

Pages in Study: 43 Candidate for the Degree of Master of Science

Major Field: Computer Science

Robotics is an emerging field and robots may be employed in places where human intervention is not possible. Multiple robots may work in a coordinated manner to achieve certain tasks. However one of the big problems is the 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 particular we look at two types of communication failure, failures caused by robot isolation and failures caused by intermittent message loss. This thesis focuses on one component of an autonomic robotic architecture, namely, self failure detection mechanism in robots. Our goal is to make the robot recognize the failure encountered during its operation and send related failure information to the activation unit for further action. We propose an approach to self-detection based on observation graphs. Simulation results show that the failures were effectively detected. The proposed recognition unit is similar to T-cells in the human immune system.

ADVISER'S APPROVAL: Dr. Johnson P. Thomas