ADAPTIVE WINDOW CHECKPOINT - ROLLBACK METHOD FOR RESPONSE AND RECOVERY MECHANISM IN ROBOTICS

By

DEEPAK MURALIDHARAN

Bachelor of Science

Anna University

Chennai, India

2006

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

ADAPTIVE WINDOW CHECKPOINT - ROLLBACK

METHOD FOR RESPONSE AND RECOVERY

MECHANISM IN ROBOTICS

Thesis Approved:

Dr. Johnson P. Thomas

Thesis Adviser

Dr. Venkatesh Sarangan

Dr. Nohpill Park

Dr. A. Gordon Emslie

Dean of the Graduate College

ACKNOWLEDGMENTS

I wish to express my gratitude to my graduate advisor Dr. Johnson P. Thomas for his inspiration and guidance, He provided sound advice, good ideas, good teaching and taught me the art of technical writing.

I would like to thank the many other people who taught me the basics of computer programming, in particular my undergraduate teachers Mr.Madhusudhan, Mr.Jagan, Ms.Geetha, Ms.Mehjabeen, and my graduate teachers, Dr.Venkatesh Sarangan, Dr.Nophill Park. I would also like to thank our systems manager Mr.Terry Wright for his assistance during system failures.

I would like to thank my friends Santosh, Arvind, Suresh, Ashwin, Siva, and Srikanth for supporting me during the difficult times and for their emotional support and the care they provided. I wish to thank my entire family for their loving support and encouragement. In particular, my nephews, niece, and cousins were very much supportive of me.

Lastly and most importantly, I would like to thank my parents for their continuous 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.

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 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. Robots need to respond themselves to such failures and they should be able to recover from the encountered failure. Increasingly more artificial intelligence is being added on them 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 and so on. The human body has multiple subsystems, all working independently of each other all the time. The human *immune sub-system* is one such sub-system. It is responsible for recovering the human body from any kind of invasion or an attack or a failure. Emulating the immune system in robots form the basis for the robot to recover from attacks and failures. In this thesis we propose a robotic architecture based on the human immune model to develop a robot that can self detect of failures and furthermore recover from failure back to a normal state.

The artificial immune system proposed for robots contain three subsystems namely, a recognition unit, an activation unit and response and a recovery unit. The recognition unit detects the failure and 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.

This thesis focuses on developing a self response recovery mechanism for a robot based on the human body model. We have proposed an approach for recovery that uses the checkpoint rollback mechanism based on an adaptive window scheme to recover from a failure. Chapter 2 presents the literature review of the robotics and the human body model and chapter 3 gives a detailed description about the human body immune system. Chapter 4 provides the problem specification addressed in this thesis. Chapter 5 gives a detailed description about the implementation and simulation results. Chapter 6 concludes the thesis.

CHAPTER II

REVIEW OF LITERATURE

In this chapter we review the literature in the area of self recovery models for robotics.

Baydor and Saitou [1] proposed an error recovery system that heavily relies on Bayesian Inference for error diagnosis and Genetic Programming for recovery. Mahdavi and Bentley [2] proposed an online evolutionary algorithm to automatically recover behavior from a failure. This algorithm is dependent on the number of hardware trials and the time taken to recover is very high. Srinivas [3] was one of the early researchers to study error diagnosis and recovery, but the algorithm that he has proposed requires repeated testing on the physical robot. Moreover, the algorithm that he proposes does not handle unanticipated errors.

Automated recovery algorithms have been proposed by Josh Bongard and Hod Lipson for remote robotics applications [4]. These algorithms help the robot to recover from unanticipated failures. A two-stage evolutionary algorithm, the estimation-exploration algorithm, is proposed. This algorithm evolves a damage hypothesis after every failure or damage. It then recovers by using a neural controller to restore its original functionality.

The estimation evolutionary algorithm (EA) evolves a hypothesis about the actual failure to the physical robot. It records the forward displacement of the physical robot along with the controller that is acting upon that robot. When the EA is terminated, it returns the best fit damage hypothesis to the exploration EA. The exploration EA evolves a controller for the physical robot. The algorithm generates a controller for the current state of the robot. Further passes generate a controller for the damaged robot using the best damage hypothesis generated by the estimation phase.

Barnhard, McClain, Wimpey, and Potter [5] proposed a system that solves the Honey-Bee task. The task of a honeybee is to direct other honeybees to their destination. Honeybees have the special ability to find flowers that produce pollen. Once a bee finds the food source, it goes to the location of the other bees and performs some kind of dance movements to communicate the food location to the other honeybees. Thus, the other bees are able to find the source without any further search. This task is implemented in robotics using Bluetooth communication to lead the other blind robot towards the target.

Two robots are used here namely Odin and Hodur. Odin is the guiding robot and Hodur is the blind robot. Blind robot refers to the robot that does not have any knowledge of the target location. It follows the instructions given by the guiding robot. The guiding robot Odin explores the environment and finds the specific target. It then communicates the target location to the blind robot Hodur through the Bluetooth device. Hodur does not have any sensors to sense the target. It relies completely on the information provided by Odin.

In the past, research has been done in the form of Genetic Programming, Bayesian Inference, Evolutionary Algorithm, and Neural Controller. Checkpoint-Rollback method is a different approach to handle response and recovery process for robot failures. Our approach emphasizes in preserving the actions committed before the failure had occurred, this has not been looked upon in the existing approaches.

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 Base station. When a Robot encounters a failure, it should use some failure detection mechanism to detect the possible failure, for the 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. 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. 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 produces unlimited number of antibodies (antigen-specific) that kill 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 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 ARCHITECTURE

The proposed immune system in a robot is shown below:

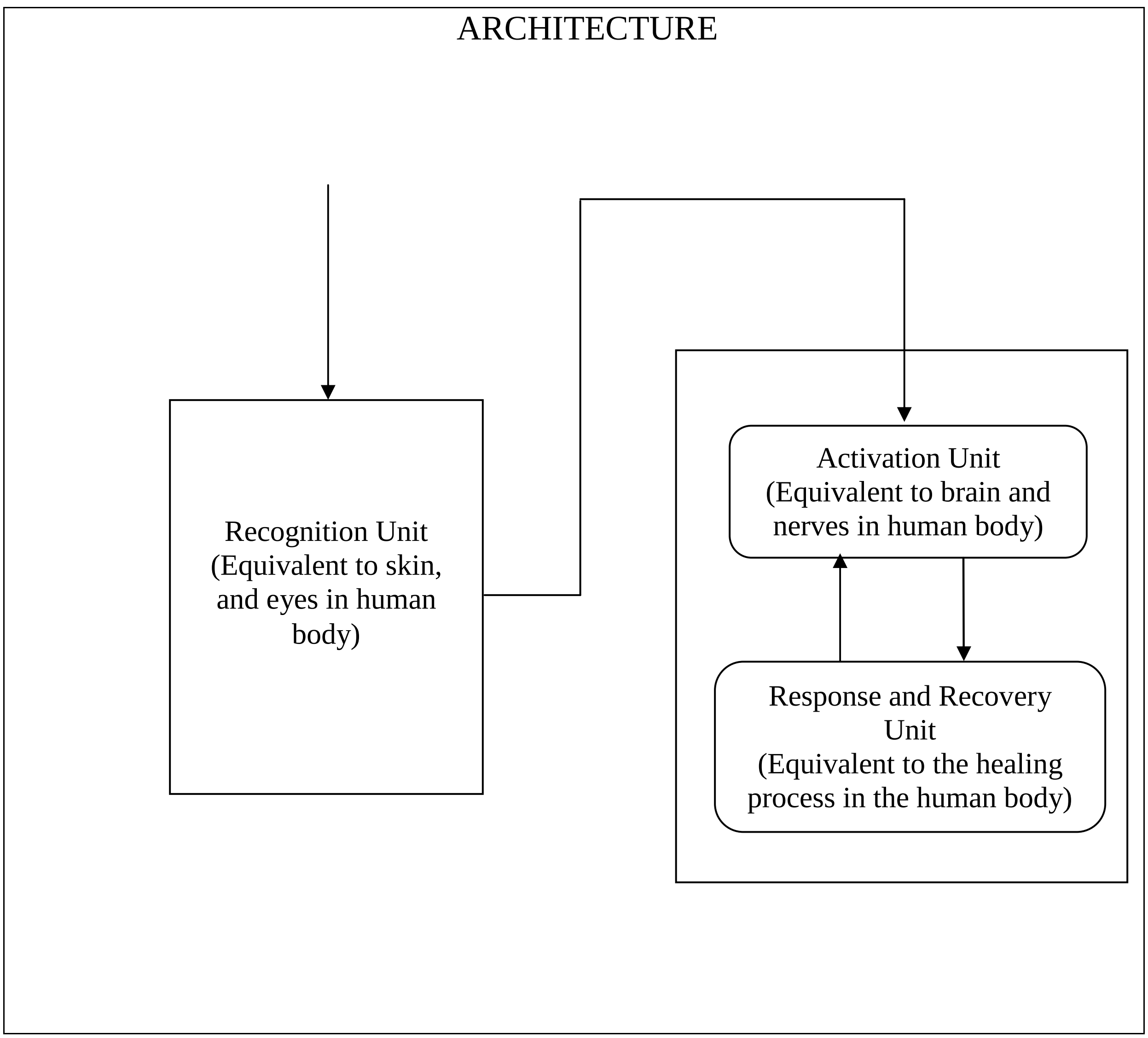

Fig 5.1: Proposed Robotic Architecture

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 the 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-Recovery Unit:

The final unit is the Response and Recovery unit. This is responsible for taking actions according to the parameters setup by the activation unit. This unit receives a solution from the activation unit and executes the solution. It also decides whether to take a new action or to execute the solution that is sent by activation unit based on the feasibility of the solution sent by the activation unit. After taking the necessary action, it sends a feedback to the activation unit to update its repository. This feedback indicates whether the solution sent by the activation unit has been successfully executed or not.

CHAPTER VI

PROPOSED APPROACH 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. 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 6.1: 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. 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 6.2: 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 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 activation unit. If the action sent by the activation unit is not feasible, it implements its own action to recover from its current situation. 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 6.3: 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 covers 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

The input for Response-Recovery unit is received from the activation unit. The input is a tuple format <Problem, Cause, and Action>. Problem represents the problem as identified by the detection unit, cause states the cause of the problem and action is the solution that is recommended to recover from the problem.

We propose an algorithm for the Response-Recovery mechanism of robots. The Response-Recovery mechanism checkpoints the instructions in the time intervals defined by an adaptive window mechanism. If a failure occurs, based on the checkpoint, Rollback and Recovery takes place. This algorithm is based on an Adaptive Window Checkpoint-Rollback scheme

7.1. Checkpoint

During the check pointing process, issues two issues that needs to be addressed are,

- What to checkpoint?
- When to checkpoint?

7.1.1. What to checkpoint:

A Checkpoint consists of the following:

1. Robot ID
2. Current Position (In terms of (x, y) co-ordinates)

The information that is used in check pointing is shown in the table below:

ID_STACK	LOC_STACK
5	140, 133
5	161, 164
5	181, 188
5	177, 176

Table 7.1: Checkpoint for Robot 5

- ❖ ID_STACK: This column shows robot ID that is transmitting the checkpoint. For example, 5 indicates robot 5.
- ❖ LOC_STACK: This column shows the current location of the robot in the environment in terms of (x, y) co-ordinates.

7.1.2 When to checkpoint:

Since a large number of instructions will be generated and transmitted by the base station, it is not feasible to checkpoint all the instructions transmitted by the base station. This will result in a huge memory overhead. There will be substantial overheads in message transmission.

To overcome this problem, we use an adaptive window scheme for storing the instructions in the checkpoint table. A counter keeps track of the message ID in the robot. Whenever the robot receives an instruction from the base station and executes it, the instruction counter gets updated. Every time when the instruction counter hits the adaptive window level, the checkpoint gets synchronized between the robots and the base station i.e. the checkpoint is transmitted from robot to the base station. After the robot synchronizes its checkpoint with base station, the adaptive window gets doubled. This process is continued until a failure is detected. When a failure occurs, the adaptive window is reset and it will start incrementing from the point at which it encountered failure.

The overhead of storing the instruction is therefore greatly reduced as the checkpoint table stores instructions with message ID that matches the current value of the adaptive window counter.

7.2 Rollback

When a failure occurs, the robot has to roll back to a location from where it can communicate with the base station. The locations based on the checkpoint stacks will provide such data. Based on the location indicated by the checkpoint stack, the robot will rollback. This rollback process continues until the robot establishes a connection with the base station.

7.3 Recovery

After the robot rolls back to a location within the communication range of the base station, it will transmit a message to the base station, requesting the base station to send the instructions that it had missed during the time of failure. On receiving this request, the base station will issue the missed instructions to the robot and the robot will send back the acknowledgement after executing all the issued instructions. When the robot receives a message that is not in sequence, it compares the current message ID that it has received with its predecessor and requests the base station to resend the missing messages.

7.4 Algorithm

// Function for storing the checkpoint

Function Store_Check_Point

begin

1. Push (Current_Location, LOC_STACK)
2. Push (Message_ID, MSG_STACK)

end

//Function for check pointing

Function Check_Point

begin

1. Message_Count ← Message _Count + 1
   a. **if** Message _Count = Check_Point _Counter **then**
      i. Call Update_ Adaptive_Window
      ii. Call Store_Check_Point
      iii. Transmit(Current_Location, BaseStation)
      iv. Transmit(Message_ID, BaseStation)
   b. **end if**

end

//Function for rolling back

Function Roll_Back

begin

1. **while** failure = true **do**
    a. Location ← Pop(INS_STACK)
    b. Move (Location)
2. **end while**

end

Steps for Recovery:

1. For robot isolation problem,
    a. If there exists a connection to the base station, send a request to the base station to continue sending instructions.
    b. Start executing the instructions
2. For out of order message problem,
    a. Send a request to the base station to resend missed messages.
    b. Start executing the instructions in order

CHAPTER VIII

IMPLEMENTATION AND RESULTS

8.1 Objective

We have proposed an algorithm for robots to respond to failure and recover from the failure by themselves 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 the total number of messages transmitted, message overhead, and failure factor.

8.2 Framework Description

8.2.1 Scenario

The simulation environment is in the form of a rectangle. This environment 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, the robots could lose some messages. We have proposed architecture based on the human body model which can detect and recover from failures.

8.2.2 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.

8.2.3 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 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 only one way communication exists. This can be either from the base station to robot or from the robot to base station. Failure is not considered when a robot stops by observing an obstacle in its path.

8.3 Instruction Format

As mentioned earlier, 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.

8.3.1 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.

8.4 Addressed Failure

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.

8.4.1 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 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. This time the communication is from robot to base station.

8.4.2 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 a previously unreachable position 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 the presence of an obstacle on the communicating path.

8.5 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, 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 to the missed instruction list. 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 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. Obstacles are considered as failure only when they are present in robot’s path during communication.

8.6 Integrated System

The 3 main components of the integrated system are

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

8.6.1 Recognition Unit

The Recognition sub system 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. This unit is independent of the other units as it keeps checking continuously for failures all the time. Failure factor is calculated for failures on a robot and the type of failure (permanent or temporary failure) is decided based on the value of the factor.

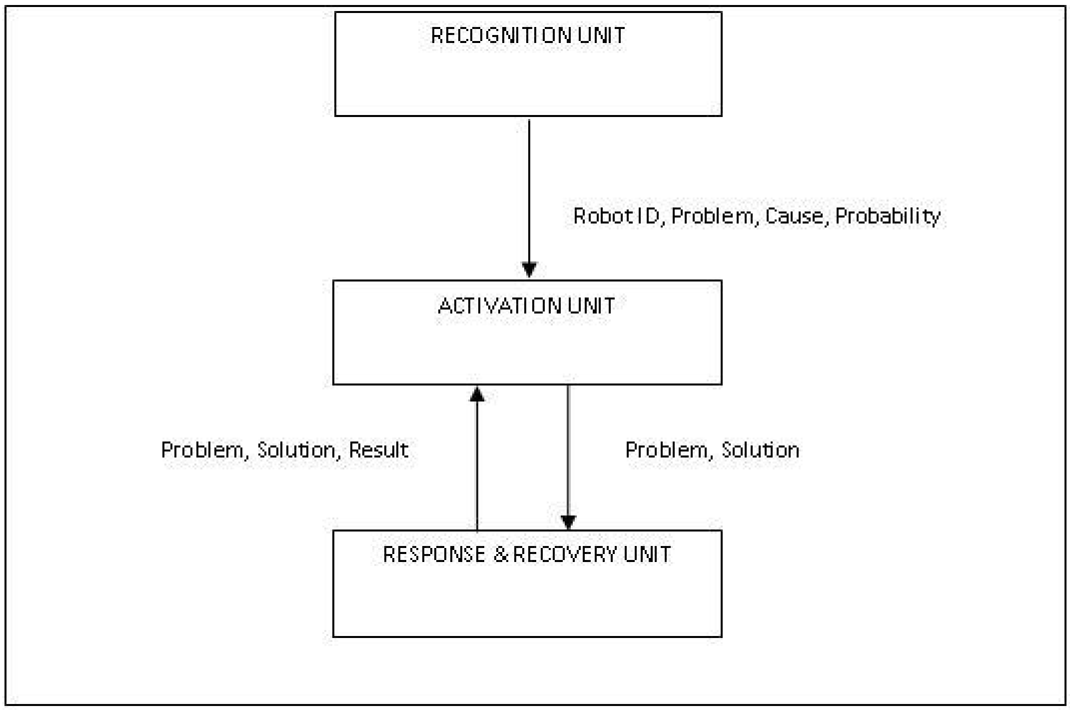

Fig 8.1: Data flow among the components

This unit calls the activation unit with 3-tuple format.

(Problem, Cause, Probability of cause of failure) ----Equation 8.6.1

8.6.2 Activation Unit

The activation unit is invoked by the recognition unit. The activation unit houses the knowledge repository. The Knowledge Repository consists of problem, cause, action. It stores actions for all the problems that it had encountered. It is also responsible for the self-learning mechanism. It has a database that maintains a series of actions for the foreign behavior that it encountered previously. This unit ranks the actions in terms of their efficiency. Based on the type of failure and its associated action's rank, this sub-system instructs the response unit to take appropriate action.

This unit calls the Response and Recovery Unit with 3-tuple format.

(Problem, Cause, Action) ----Equation 8.6.2

8.6.3 Response-Recovery Unit

The Response and Recovery subsystem is responsible for recovering from the failure. As the robot receives the instructions from the base station, it executes them by moving the distance indicated by the instruction. After executing the instruction, the robot waits for a constant wait time. After the wait time expires and the robot has not received any instruction, then it checks whether it is currently ble to communicate with the base station or not. If it is not able to communicate, then it comes to a conclusion that it is failed.

After moving the distance specified by the instruction, if the robot is able to communicate with the base station and if the robot's current instructions counter equals the adaptive window level, then the robot checkpoints the current location. The robot then sends the checkpoint to the base station and updates its position in the base station. For example, if the robot 5 receives an instruction 5R7M4 when it is in a location (142, 151), then it moves 7 unit towards its right and updates its current location in the base station as (149, 151).

During the time of a failure, the robot rolls back to the position indicated by the stored checkpoint. For example, if the checkpoint for the robot 5 is stored as (156, 151), then the robot will move from (149,151) to (156,151). This process will continue until robot is in a location from where it can communicate with the base station.

There are two kinds of failure. One is robot isolation and the other is message loss. For both the failures, different solutions should be taken. The two solutions that are considered in this study are rolling back to a previous checkpoint and sending a missed message request to the base station. The Activation unit sends the message to the response-recovery unit for executing the action. The input will look like

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

Or

For isolation (Communication, Isolation, Roll back)

But there is no guarantee that the solution suggested by the activation unit will solve the problem. If the solution provided by the activation unit is incorrect, then the response-recovery unit will execute its own action that is suitable for the current problem. After taking the action, it will send a feedback to the activation unit.

The feedback format is a 4-tuple format:

(Problem, Cause, Action Taken, Feasibility)----Equation 8.6.3

Ex1: Action from activation unit:

(Communication, Isolation, Rollback)

Feedback from response unit:

(Communication, Isolation, Rollback, True)

Ex2: Action from activation unit:

(Communication, Isolation, Missed Message Request)

Feedback from response unit:

(Communication, Isolation, Rollback, False)

In the first example (Ex1), for the problem, the activation unit's recommended action is "Roll back". The response unit executes the action and sends the result as “true”, implying that the action suggested by the activation unit solved the problem. In the second example (Ex2), the action recommended by the activation unit is a missed message request to the base station. On receiving a missed message request from the activation unit, the response recovery unit checks for an acknowledgement from the base station. If it had not received any acknowledgement from the base station, then it comes to the conclusion that the robot needs to rollback to get within communication range of the base station.

Hence, the response recovery unit sends false as feedback to the activation unit. This means that the action sent by the activation unit does not solve the problem, and it therefore takes “Roll back” to solve the problem and sets the result to false. The feedback from the response unit will contain the action taken (it may be different from action sent by the activation unit) and the result (**true**-for solving problem after implementing action sent by the activation unit or **false** – for taking the new action instead of action sent by the activation unit). This is considered by the activation unit to improve its recommendation for future actions.

8.7 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 connections and obstacles. If a robot is not reachable from the base station, the instructions that got missed out for that robot is stored in the missed instruction list and they are sent to the robot whenever the robot gets connection and requests for the instructions. Instructions are stored in the base station's instruction queue. After the robot receives the instruction and after the wait timer in the robot expires, it requests the base station for the missed instruction.

8.7.1 Message Overhead:

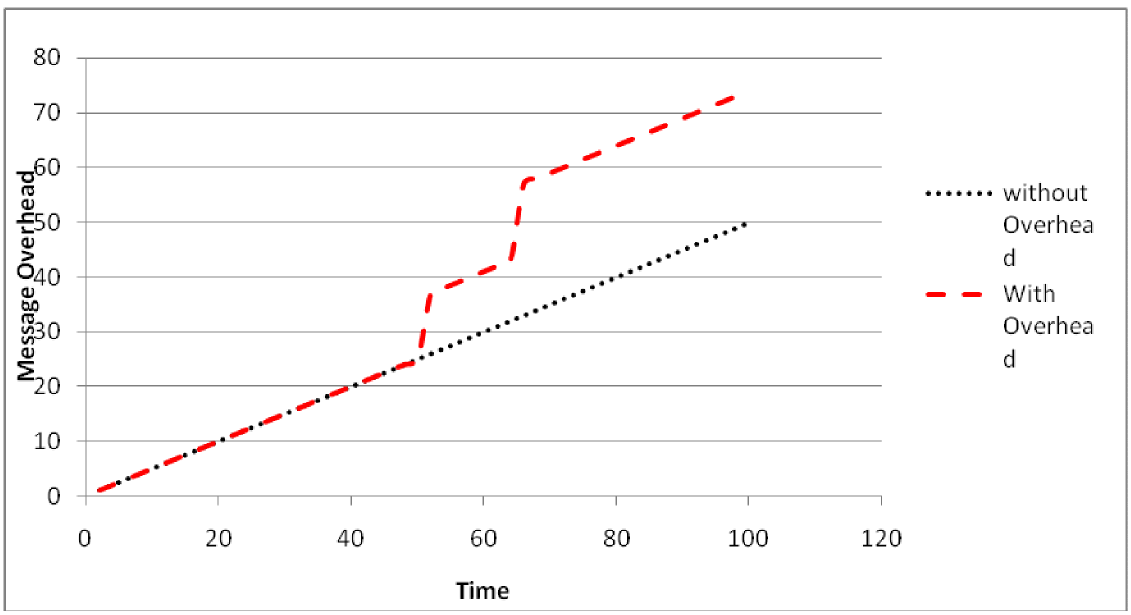

Fig 8.2: Message Overhead

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 while testing condition for isolation and message loss. If there is no instruction from the base station to the robot till the wait time expires, the robot checks for a connection with the base station costing two acknowledgements. Also, after rolling back to a previous position, it checks for a connection back to the base station. So the total number of acknowledgements increases to four. This acknowledgement consists of Robot ID, and a Boolean value for connection. Message overheads can be controlled by using a variable wait time.

8.7.2 Total Number of Rollbacks and Successful Connections

The graph shown below is plotted between the Robot ID in X axis and the total number of rollbacks and successful connections in Y axis. From the graph we infer that, when there is an increase in the total number of rollbacks for a robot, the probability for the robot to recover from failure also increases. In the graph shown above, Robot 10 has rolled back 37 times out of which it has recovered 36

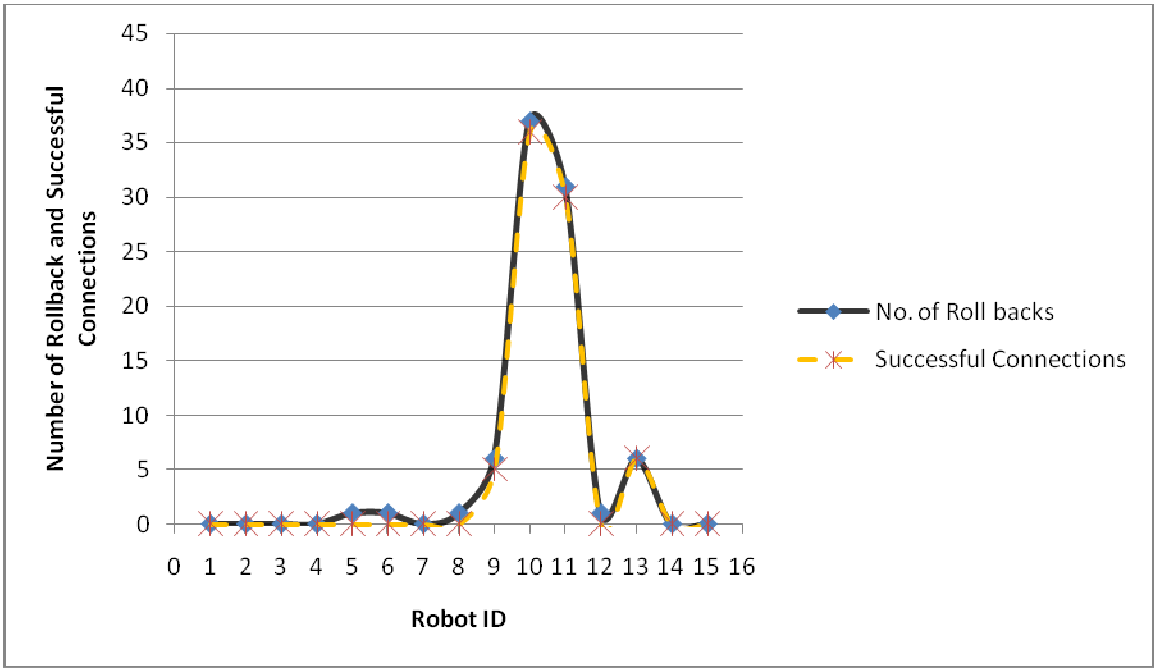

Fig 8.3: Total Number of Rollbacks and Successful Connections

times. But Robot 5 has rolled back only once and it failed to recover. There might be multiple roll backs for a failure. The robot rolls back to a previous checkpoint and checks for a connection with the base station. Even after rolling back, if the robot is still in an isolated state, then the robot rolls back again by executing its previous instructions in the reverse order until it finds a connection to the base station. This graph shows that the chances of getting a successful connection increases when the robot rolls back more number of times.

We were able to validate the proposed algorithm with the help of the simulation tool. We tested the algorithm with different inputs and calculated metrics like the overall message overhead incurred while transferring the messages between robot and base station (Figure 8.2) and the total number of rollbacks taken by the robots and the total number of successful connections (Figure 8.3).

CHAPTER IX

CONCLUSION

9.1 Summary

Our overall goal is to propose an autonomous architecture for robots for self detection and recovery from a failure. Our model is based on the human immune system. 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.

The problem was defined and a plausible solution has been proposed and simulated. The proposed architecture increases the overhead in terms of acknowledgement between the base station and robots

9.2 Future Work

Future work may investigate implementing some mechanism to reduce the message overhead. We have used the adaptive window scheme to checkpoint the instructions. A much more effective scheme can be used to handle the checkpoint process. This architecture could be extended to different areas of research such as computer security, intrusion detection and error analysis.

REFERENCES

[1] K.Saitou K, C.M. Baydarw. "Off-line error prediction, diagnosis and recovery using virtual assembly systems." Proceedings of 2001 ICRA. IEEE International Conference on Robotics and Automation. ICRA '01,pp. 818- 823, 2001

[2] Bentley, P.J, Mahdavi S.H. "An evolutionary approach to damage recovery of robot motion with muscles." Proceedings of the European Conference on Artificial Life ECAL '03: pp. 248-255. 2003

[3] S. Srinivas, “Error Recovery in Robot Systems,” Ph.D. thesis, California Institute of Technology, Pasadena, CA, 1977.

[4] Hod Lipson, Josh C. Bongard, "Automated Damage Diagnosis and Recovery for Remote Robotics." Proceedings of the 2004 IEEE International Conference on Robotics and Automation, pp. 3545- 3550, 2004

[5] D. H. Barnhard, J. T. McClain, B. J. Wimpey, W. D. Potter, "Odin and Hodur: Using Bluetooth Communication for Coordinated Robotic Search." Proceedings of the 2004 International Conference on Artificial Intelligence IC-AI '04, pp. 291-296, 2004

[6] V. Moulin, G. Castilloux, A. Jean, D. R. Garrel, F. A. Auger and L. Germain, "In vitro models to study wound healing fibroblasts." Burns, pp.359-362, 1996

VITA

DEEPAK MURALIDHARAN

Master of Science

Thesis: ADAPTIVE WINDOW CHECKPOINT - ROLLBACK METHOD FOR RESPONSE AND RECOVERY MECHANISM IN ROBOTICS

Major Field: Computer Science

Biographical:

Personal Data:

Born on December 16th, 1984 in India

Education:

Completed Bachelor of Engineering in Computer Science and Engineering at Sriram Engineering College, Anna University, Chennai, India in May, 2006.

Completed the requirements for the Master of Science in Computer Science at Oklahoma State University, Stillwater, Oklahoma in December, 2008.

Experience:

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

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

Name: Deepak Muralidharan Date of Degree: December, 2008

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: ADAPTIVE WINDOW CHECKPOINT - ROLLBACK METHOD FOR RESPONSE AND RECOVERY MECHANISM IN ROBOTICS

Pages in Study: 44 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 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 response recovery mechanism in robots. Our goal is to make the robot respond to the failure by rolling back to a safer state from where it can communicate with the base station and recover from the failure by resuming its operation from the rolled back location. Simulation results show that the probability of getting a successful connection back to the base station increases when the number of rollbacks for a robot increases.

ADVISER’S APPROVAL: Dr. Johnson P. Thomas