A THESIS

FOR THE DEGREE OF MASTER OF SCIENCE

Simulations and Experiments of collision avoidance algorithm for Silvermate robot

Minh-Ngoc Nong

Department of Mechatronics Engineering

GRADUATE SCHOOL

CHEJU NATIONAL UNIVERSITY

2008.05

Simulations and Experiments of collision avoidance algorithm for Silvermate robot

Minh-Ngoc Nong

(Supervised by Professor: Kyung-Hyun Choi)

A thesis submitted in partial fulfillment of the requirement for
the degree of Master of Engineering
2008. 05.

This thesis has been examined and approved.

Thesis director

Choi, Kyung Hyun, Professor of Mechatronics Department.

Lim, Jong Hwan, Professor of Mechatronics Department.

Kang, Chul Ung, Professor of Mechatronics Department.

May,2008

Date

Department of Mechatronics Engineering

GRADUATE SCHOOL

CHEJU NATIONAL UNIVERSITY

ABSTRACT

Robots have been being applied to our life. They have been replaced for human beings in the dangerous working, noxious environment and so on. In Robotics, ***"Collision avoidance"*** is one of the key issues to successful applications of mobile robot systems. Recently, many studies on real time collision avoidance have been achieved and all mobile robots feature some kind of collision avoidance which requires the robot to detect the obstacle; take the measurements, and resume motion with real time system.

This thesis proposes the Modification Elastic strips method for mobile robot to avoid obstacles with a real time system in an uncertain environment. The method deals with the problem of robot in driving from an initial position to a target position based on elastic force and potential field force. To avoid the obstacles, the robot has to modify the trajectory based on signal received from the sensor system. The method based on the elastic force repulsed of the nearest obstacles: one on the left, one on the right to calculate new position of robot.

In addition, the application of Pseudomedian filter is proposed to process the non-linear signal from the sensors. It has been removed the noises of these signal.

The simulations and experiments of these methods are carried out. The trajectory of robot is smoother and more adaptive in an environment including multiple obstacles with the combination of Modification Elastic strips and Pseudomedian filter.

ACKNOWLEDGEMENTS

This thesis is the most significant scientific accomplishment in my life and it would be impossible without people who supported me and believed in me.

First of all, I would like to express my deep gratitude to my supervisor Professor Kyung Hyun, Choi for his valuable advices during the time of my studying in Korea. Perhaps, I cannot thank him enough but I promise that I will try to do with the best in the future.

I am also very grateful to Professor Cho, Kyung Hoo, Professor Lim, Jong Hwan, Professor Kim, Sang Jae and Professor Kang, Chul Ung for giving me a chance to study here.

My sincere thanks are extended to my classmates and all members of the Advance Micro Mechatronics Laboratory (AMM lab): my lab's leaders Chang Jong, Kim, Jeung Beom, Ko; my lab mates: Huang Chan, Yuang, Bae Jin, Trung Thanh, Tran, Khalid Rahman, Asan Rahman, Asif Ali Rahmani, Huang Chan, Kim and Su Jin, Kim who are help me so much during the time I study in our laboratory.

Especially, I would like to give my special thanks to my wife for her support and encouragement. To my son, Hoang Anh, only one thing I want to tell him that no word can express how much I love him.

To my grandparents, parents and grandmother-in-law, parents-in-law, from the depth of my heart I wish to thank them for everything that they are doing for me. Thanks to my brothers, sisters and my friends for their help during my studying.

Finally, I would like to give my best regards to Associate Professor Dang Binh, Nguyen, Associate Professor Nhu Khoa, Ngo, Associate Professor Huu Cong, Nguyen and Mr. Tien Duy, Nguyen at Thainguyen University of Technology, Vietnam for their advices and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

Robots have been being applied to our life. They have been replaced for human beings in the dangerous working, noxious environment and so on. Hence, the mobile-manipulator robots should be developed, which can cope freely with uncertain environment in addition to transfer function and manipulation. Basic functions in a mobile-manipulator robot have to follow collaboratively. There is function that can judge naturally through realization for given environment, function that can create path and trajectories by real time to move. Development of collision avoidance algorithm to recognize and avoid surroundings obstacles by real time should be obtained.

The collision avoidance is one of the key issues to successful applications of mobile robot systems. All mobile robots feature some kind of collision avoidance, ranging from primitive algorithms that detect an obstacle and stop the robot short of it in order to avoid a collision, through sophisticated algorithms, that enable the robot to detour obstacles. The latter algorithms are much more complex, since they involve not only the detection of an obstacle, but also some kind of quantitative measurements concerning the dimensions of the obstacle. Once these have been determined, the obstacle avoidance algorithm needs to steer the robot avoid the obstacle and proceed toward the target.

Recently, many studies on real time obstacle avoidance have been achieved. The method requires the robot to detect the obstacle; take the measurements, and resume motion with real time system.

1.1 Related works

1.1.1 Edge Detection

One popular obstacle avoidance method is based on edge-detection. In this method, an algorithm tries to determine the position of the vertical edges of the obstacle and then steer the robot around either one of the "visible" edges. The line connecting two visible edges is considered to represent one of the boundaries of the obstacle. This method was used in our very early research [4], as well as in several other works [6,7,11], all using ultrasonic sensors for obstacle detection.

A disadvantage with the implementations of this method is that the robot stops in front of obstacles to gather sensor information. So, it is suitable for planning method. It cannot apply for robot with real time.

1.1.2 Certainty Grid

A method for probabilistic representation of obstacles in a grid-type world model has been developed and proposed by Moravec at Carnegie Mellon University [2,3,13]. This world model, called *Certainty Grid* is especially suited to the accommodation of inaccurate sensor data such as range measurements from ultrasonic sensors.

In the certainty grid the robot's work area is represented by a two-dimensional array of square elements, denoted as cells. Each cell contains a certainty value (*CV*) that indicates the measure of confidence that an obstacle exists within the cell area. With this method, *CVs* are updated by a probability function that takes into account the characteristics of a given sensor. Ultrasonic sensors, for example, have a conical field of view. A typical ultrasonic sensor [5] returns a radial measure of the distance to the nearest object within the cone, yet does not specify the angular location of the object. If an object is detected by an ultrasonic sensor, it is more likely that this object is closer to the acoustic axis of the sensor than to the periphery of the conical field of view [1]. For this reason, this method's probabilistic function *C* increases *CVs* in cells close to the acoustic axis more than *CVs* in cells at the periphery.

In this method, there are many problems were occurred in distance measurement of obstacle by impreciseness of *CV* values.

1.1.3 Artificial Potential Field Methods (APF)

The idea of imaginary forces acting on a robot has been suggested by Khatib [10]. In this method, obstacles exert repulsive forces, while the target applies an attractive force to the robot. A resultant force vector $\vec{R}$, comprising the sum of a target-directed attractive force and repulsive forces from obstacles, is calculated for a given robot position. With $\vec{R}$ as the accelerating force acting on the robot, the robot's new position for a given time interval is calculated, and the algorithm is repeated.

Krogh [12] has enhanced this concept further by taking into consideration the robot's velocity in the vicinity of obstacles. Thorpe [17] has applied the potential field method to off-line path planning and Krogh and Thorpe [20] suggest a combined method for global and local path planning, which uses a *“Generalized Potential Field”* approach. Newman and Hogan [15] introduce the construction of potential functions through combining individual obstacle functions with logical operations. While each of the above methods features valuable refinements, none have been implemented on a mobile robot with real sensory data. By contrast, Brooks [8,9] and Arkin [1] use a potential field method on experimental mobile robots (*equipped with a ring of ultrasonic sensors*). Brooks' implementation treats each ultrasonic range reading as a repulsive force vector. If the magnitude of the sum of the repulsive forces exceeds a certain threshold, the robot stops, turns into the direction of the resultant force vector, and moves on.

Common to these methods is the assumption of a known and prescribed world model, in which simple, predefined geometric shapes represent obstacles and the robot's path is generated off-line.

1.1.4 Virtual Force Field (VFF) method

A method is our earlier real-time obstacle avoidance method for fast-running. This method was proposed by Borenstein [18]. The VFF method allows for fast, continuous, and smooth motion of the controlled vehicle among unexpected obstacles, and does not require the vehicle to stop in front of obstacles.

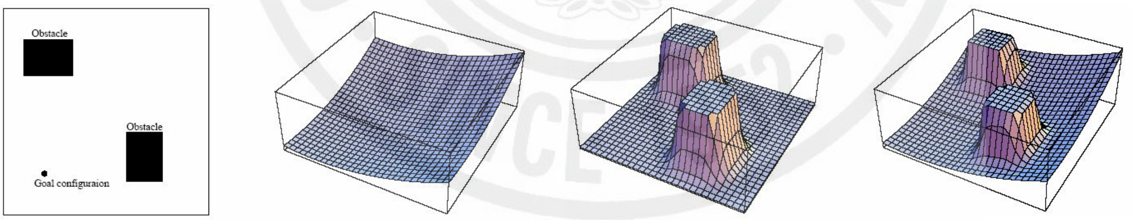
The VFF method uses a two-dimensional *Cartesian histogram grid C* for obstacle representation. Like in Certainly Grid method's certainty grid concept, each cell (i,j) in the histogram grid holds a certainty value *“C”* that represents the confidence of the algorithm in the existence of an obstacle at that location.

The histogram grid differs from the certainty grid in the way it is built and updated. The Certainly Grid method's projects a probability profile onto those cells that are affected by a range reading; this procedure is computationally intensive and would impose a heavy time penalty if real-time execution on an on-board computer was attempted. On the other hand, this method increments only one cell in the histogram grid for each ranges reading, creating a "probability" distribution with only small computational overhead. For ultrasonic sensors, this cell corresponds to the measured distance and lies on the acoustic axis of the sensor. While this approach may seem to be an oversimplification, a probabilistic distribution is actually obtained by continuously and rapidly sampling each sensor while the vehicle is moving. Thus, the same cell and its neighboring cells are repeatedly incremented. This results in a histogram probability distribution in which high certainty values are obtained in cells close to the actual location of the obstacle.

However, most of these methods have not considered dynamic limitation. Therefore, when a robot moves bottleneck in actuality, it cannot avoid these obstacles and has stuck on the way to reach the target.

1.2 Research Aims and Scope

The Elastic strips was proposed by Khatib [18,19]. It was proposing the trajectory of robot as an elastic material in avoiding the obstacles in real time. When approaching to the obstacles, the virtual force between objects was similar to the artificial potential field concept (Figure 1).



Environment Field Goal Field Obstacle Field Potential Field

Figure 1.Potential Field concept

The method revealed the stabilized navigation technology compared to the artificial potential field method due to the elastic force beside repulsive force whiles reaching to objects. Its application was easy in robots having multi-degree of freedoms which were effective though integrate motion plan and execution of robot at the same time. That command algorithm was optimized the real time obstacle avoidance in dynamic and uncertain environment.

The Elastic strips based on only the nearest point with obstacle, therefore, in some cases, robot was not able to continue to reach the target and it has stuck on the way to the target point in a multi-obstacles environment. Also, if the distance between two obstacles was not much, Elastic strips still found the way to reach the target but the trajectory of robot is not smooth.

The algorithm for avoiding the collision proposed in the paper can make robot create a safety path to reach the target of using Modification Elastic strips. This method based on the nearest points; one on the left and one on the right.

The calculation of Elastic strips or Modification Elastic strips usually based on the nearest points of obstacle. So, the data of sensor system should be reliable. However, the signal from sensor system always included the real signal and noise. That mean the variables using to calculate the Elastic force have no noise.

Based on these problems, the application of Pseudomedian filter in data processing with the nonlinear data received from sensor system of robot was proposed. This method is useful and popular in image processing and digital processing to remove the salt and pepper noises. Now, it can be used to process the non-linear data of sensor system in robotics.

This research presents the simulation and experiments of these studies to illustrate the effectiveness of the proposed approach.

II. SILVERMATE ROBOT SYSTEM AND CONTROL ARCHITECTURE

2.1 Robot system

The Silvermate robot was developed by Korea Institute of Science and Technology (KIST) as part of the project "Human function life support intelligence robot technical development business' 21st century frontier business", which began in October 2003 [24].

The Silvermate is a robot platform that assists elderly people in daily life. It will operate in indoor environment such as silver town or individual home. It has functions such as enhancing perception ability, consultation and psychotherapy to the elderly people. It could entertain the customer by providing games and information for daily activities as well as acting as a conversational partner. It also has function of providing simple service such as bringing books and other objects as well as assisting walking of the elderly people.

The Silvermate robot (Figure 2) has a mobile and manipulators with seven degrees of freedom. The robot moves on two wheels.

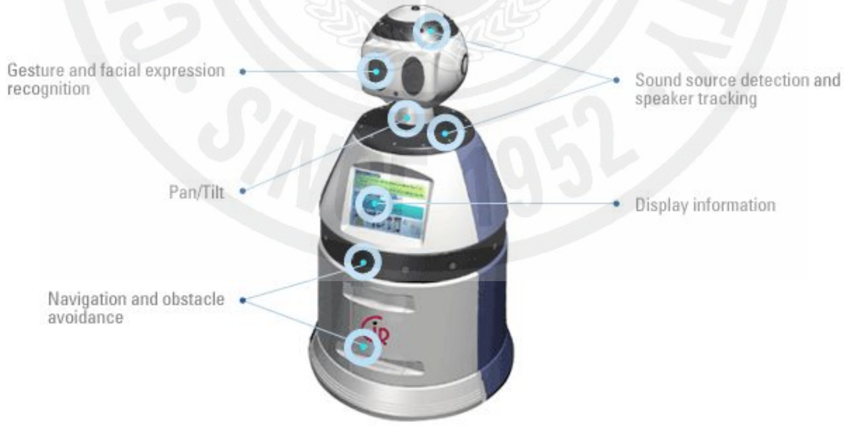


Figure 2.The Silvermate robot system and configuration

Figure 3.The vision and sensor system of robot

As show in Figure 3, the vision system and sensor system of Silvermate robot are present. The vision system included the sound detection, speaker tracking and the camera system on the head. These systems detect the voice and capture the world model to provide for robot.

The sensor system of robot included sonar sensors, laser scanners (in front and back); bumpers and IR are arranged surrounding the robot to detect information from environment.

2.2 Hybrid control architecture

The Silvermate Robot project has been initiated by the Center for Intelligent Robotics (CIR) at KIST. This project is composed of over twenty research institutes for development of various functionality of the Silvermate Robot.

The *Tripodal Schematic Control Architecture* (TSCA) developed at Korea Institute of Science and Technology (KIST) has been successfully implemented on three versions of their Public Service Robot (PSR) series. TSCA uses a hybrid approach with three layers for its control: deliberate layer, sequencing layer, and reactive layer. [21,23].

The control architecture of Silvermate robot is a modification version of TSCA. It is consisted of 3 hierarchical structures as shown in Figure 4.

Deliberative layer: takes charge of role that a robot changes given command to executable machine instruction. At performs a plan by dividing role of a robot into navigation part and manipulation.

Sequencing layer: processes information obtained from sensor to execute planning and creates information sequentially.

Reactive layer: executes commands delivered form higher level as lowest layer of control architecture. And when obstacle appeared in an environment, it copes with by real time.

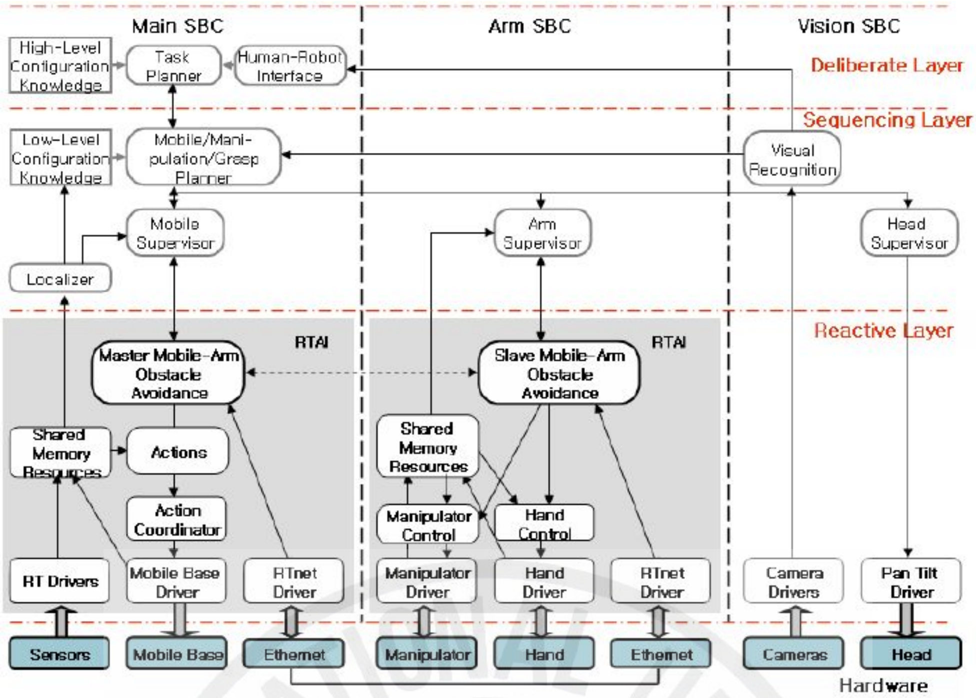


Figure 4.The Layered Architecture with Real-Time Performance Improvement

To achieve real-time performance for robot's sensing, navigation and manipulation, the designing and implementing of software components in the reactive layer using Real-Time Application Interface (RTAI) on Linux [23]. The Software components in the reactive layer require real-time performance, since sensing and movement of mobile-base or manipulator have direct effect on the safety issue of Silvermate Robots (Figure 5).

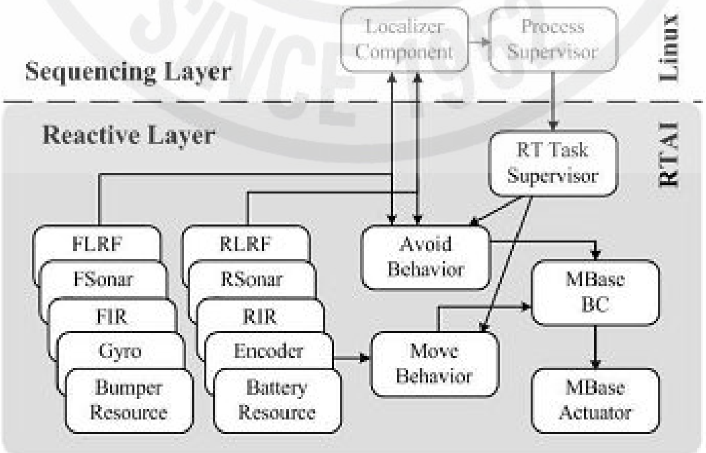


Figure 5.Reactive layer Software components

The Software component is a behavior based type with Resource, Actuator, Behavior, and Behavior Coordinator components. Each Resource components configures a given sensor hardware for periodic data acquisition, manages hardware operation, and stores the sensor data in shared memory for other software components.

An Actuator component configures and manages actuator hardware and sends generated control outputs. A Behavior component is a basis of reactive action which uses the sensory data to compute the control output.

Finally, a Behavior Coordinator (BC) component collects outputs from a set of Behavior components and fuses them for Actuator component.

III. SENSOR DATA MANIPULATION

3.1 General method

Linear filtering is a well established method for extracting signals from noisy environments. There are many tools available for analyzing linear filters, and the properties of these filters are well-understood. Nowadays, there are several methods have been being applied to process the linear signal like: *Butterworth filter, Chebyshev filters, Bessel filter, Gaussian filter, also Kalman filter etc.* These methods are very useful and popular. Nevertheless, in some situations a linear filter, even an optimal linear filter, does not perform adequately. In these cases, the special properties of certain nonlinear filters are required. Unfortunately, nonlinear filters are much more difficult to analyze than linear filters, and as a result the properties of any nonlinear filters are poorly understood.

Median filtering is one of the most common nonlinear techniques used in signal processing. Among the first to demonstrate the benefits of taking medians of sample data were Borda and Frost [31], and the median filter itself is generally ascribed to Tukey [25]. The properties of the median filter have been studied in quite some depth since these early uses [26,27,30]. The edge-preserving and impulse-removing properties of this filter are usually the most desirable features, and although the median filter is not conceptually complex, its computation can become quite cumbersome. The problem of efficient computation of the filter led Pratt, Cooper, and Kabir [28] to propose the Pseudomedian filter, a filter with properties similar to the median filter, which can be more efficiently calculated. The Pseudomedian filter often produces results that are very similar indeed to those produced by the median filter, and many of the theoretical properties of the Pseudomedian are the same or nearly the same as the corresponding properties of the median filter. However, the response of the Pseudomedian filter to high-frequency oscillations and to impulses is quite different from the response of the median filter. In many circumstances, the response of the Pseudomedian filter is preferable.

The Pseudomedian filter was designed to be a computationally efficient alternative to the median filter. However, a thorough analysis of the Pseudomedian filter reveals some important differences between its response and that of the median filter. Several theorems describe the set of signals that are invariant to Pseudomedian filtering, and show that this set is a subset of the set of signals invariant to median filtering, with the difference between the sets consisting only of fast-fluctuating signals.

The Pseudomedian filter does not completely remove impulses, as does the median filter, but both filters preserve edges. The responses of these filters to edges and impulses contrasts with those of the average and midrange filters, which neither preserve edges nor remove impulses. A generalization of the filters to continuous time reveals characteristics of the filter responses to periodic signals, particularly the ability of the Pseudomedian filter to block high frequency signals that the median filter cannot. The response of the median filter to high frequency periodic signals resembles that of the average filter, whereas the response of the Pseudomedian filter resembles that of the midrange filter. In many circumstances, the response of the Pseudomedian filter is preferable. In image processing, Peudomedian filter is very popular and useful, which have been being applied to remove the salt and pepper noise

3.2 Pseudomedian filter

The Pseudomedian filter definition is described as calculation of discrete signal in a window width 2N+1. The output (PMED) of the Pseudomedian calculation is the average of the maximum of the minima and the minimum of the maxima of the N+1 sliding subsequences of length N+1 in the window. The filtered output for the window is the filtered value at the sample in the center of the window. The definition of this filter can be illustrated by Eq.(1) with the length of windows is N.

$$PMED = \frac{1}{2}\left[Max\left(Min\left(G_i\right)\right) + Min\left(Max\left(G_i\right)\right)\right]; i = 1,.., N \tag{1}$$

With (2N+1) is the size of windows filter, N+1 is the size of sliding window and $G_i = \left[i..N+1\right]; i = 1,..,N$ is the data group taken from the windows filter.

In the definition, the Pseudomedian is defined as an average of two signal values; its output is not limited to values in the unfiltered signal. For example, an unfiltered signal consisting only of integer values does not necessarily result in a Pseudomedian filtered output of only integer values, as would be the case for the median filtered output. This averaging method also indicates that the Pseudomedian filter may have a "more linear" response than the median filter.

The Pseudomedian filter has many of the same general properties of the median filter: it exhibits a "low-pass" type of response in many cases, and preserves edges while reducing impulse noise.

Finally, the Pseudomedian filter can be implemented by an algorithm that is theoretically of lower order than most algorithms for the median filter. The fast median filtering algorithm developed by Huang [29] is a very fast implementation of the median filter, however, and a similar implementation of the Pseudomedian filter would not be as efficient. Other algorithms for the median filter are either similar in speed or noticeably slower than most implementations of the Pseudomedian filter.

3.3 Implementation and application of Pseudomedian filter

The Elastic force included two components namely internal force and external force. To calculate these forces, the algorithm based on only the nearest obstacle is received from sensor system, therefore, the data of sensor should be reliable. In fact, the data includes the real data and the noises.

The application of Pseudomedian filter to process the nonlinear data received from sensor system of robot was proposing. This method was the implementation to remove the noise of the sensor signals.

3.3.1 Sensor system

The LMS 200 is a non-contact Laser Measurement System that scans its surroundings two-dimensionally like laser radar (Figures 6,7). It operates within a temperature range of between 0°C and 50°C and, as an active scanning system, requires no auxiliary passive components such as reflectors or position markers. The LMS 200‘s high resolution allows it to take on tasks that were hitherto impossible or could only be achieved with difficulty or at great cost.

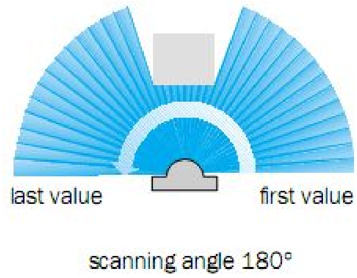


Figure 6.The laser scanner LMS 200 **Figure 7.Direction of transmission**

The LMS system operates by measuring the time of flight of laser light pulses: a pulsed laser beam is emitted and reflected if it meets an object. The reflection is registered by the scanner's receiver. The time between transmission and reception of the impulse is directly proportional to the distance between the scanner and the object (time of flight). The pulsed laser beam is deflected by an internal rotating mirror so that a fan-shaped scan is made of the surrounding area (laser radar). The contour of the target object is determined from the sequence of impulses received. The measurement data is available in real time for further evaluation via a serial interface. Automatic fog correction is active in the scanner for outdoor use. Raindrops and snowflakes are cut out using pixel-oriented evaluation. The properties of LMS can be illustrated in Table 1.

Table 1.Laser scanner system	
Distance max./10% reflectivity	80m/ 10m
Scanning range	0°...180°
Angular resolution	0.25°/0.5°/1° adjustable
Response time	53ms /26ms /13ms
Resolution/systematic error	10 mm, typ. ±15 mm
Data interface	RS 232/RS 422
Laser protection class	1 (eye-safe)
Dimensions (W x H x D)	155 x 210 x 156 mm^3

3.3.2 Implementation and results

The nonlinear data received from laser scanner of Silvermate robot included 360 values with the angular resolution of sensor is 0.5^o (Figure 7). It was stored in the share memory of robot. In experiments, we used the size of windows filter was five (N=2) and three (N=1).

We do the experiments of Pseudomedian filter in some cases. The parameters of these experiments are described in table 2.

Table 2.The parameters of Pseudomedian filter			
No	**The obstacles**	**N**	**Size of windows filter**
Case 1	1	1	3
Case 2	2	2	5
Case 3	2	2	5

In case 1, the results of Pseudomedian filter are illustrated by Figure 8. Here, the sensor of robot detects only one obstacle. The values of the laser scanner at the position of 186th, 264th, 266th, 272th and 292th included noises.

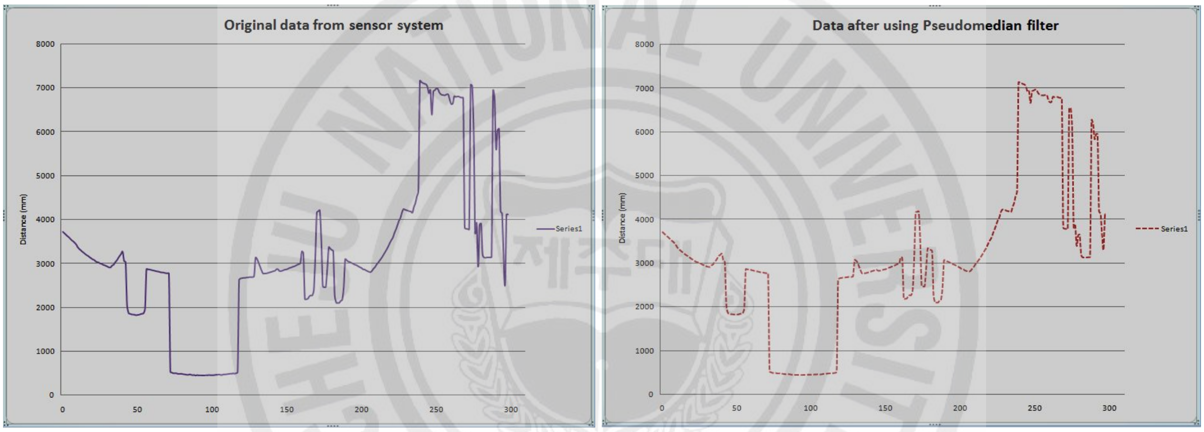


Figure 8.Data of sensor system before and after using Pseudomedian filter in case 1

To compare of data after and before using applying the Pseudomedian filter, we can see the noises of these data have been effectively removed with the use of Pseudomedian filter. The size of windows filter in this case was three with N=1.

In case 2, these are two obstacles in an environment. The robot detects and it is illustrated in Figure 9a. In this case, the noises appear at the several positions. However, the noises have been effectively removed with using Pseudomedian filter (Figure 9b).

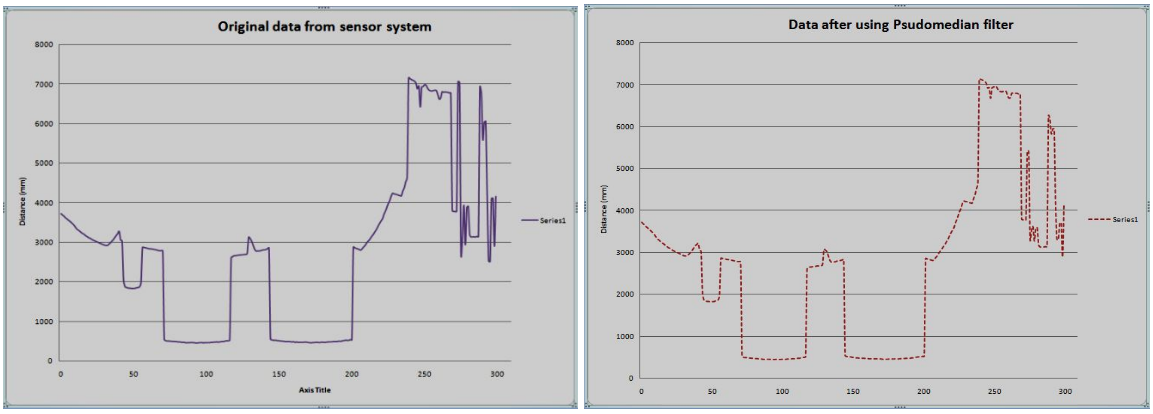


Figure 9.Data of sensor system before and after using Pseudomedian filter in case 2

In case 3, we also do the experiments with two obstacles in an environment. In this case the size of windows filter is five (N=2). The results are illustrated in Figure 10. The noise at position of 220^{th} is obvious (Figure 10a). After apply the Pseudomedian filter, the noise have been removed (Figure 10b).

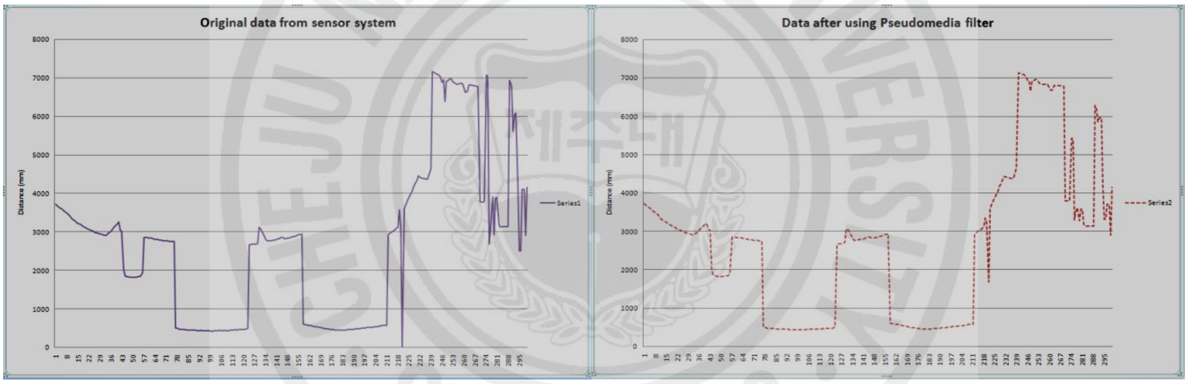


Figure 10.Data of sensor system before and after using Pseudomedian filter in case 3

The method suggests that the results measured using laser scanner with Pseudomedian filter are more reliable compared with the original ones.

IV. COLLISION AVOIDANCE ALGORITHM

4.1 Interpretation for mobile robot

The Figure 11 introduces the mechanism of mobile robot with two wheels and the coordinate system of robot.

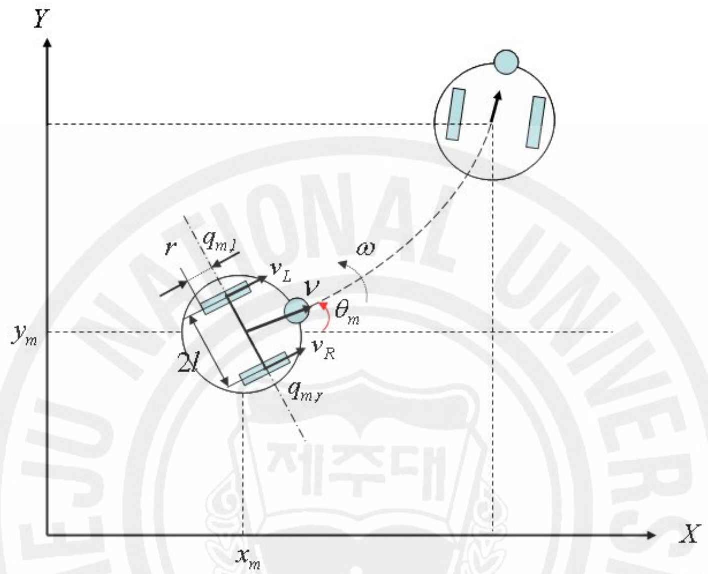


Figure 11.Mobile robot modeling and coordinate system

Here,

X, Y : Cartesian coordinate system.

(x_m, y_m): The position of robot in Cartesian coordinate system.

(q_{ml}, q_{mr}): The angular velocity of the left and right wheels.

l : The distance from the center of robot to each wheel.

r : The angular velocity at the center of robot.

(v_l, v_r): The linear velocity of the left and right wheels.

v : The linear velocity at the center of mobile robot.

ω : The Angular velocity at the center of the mobile robot.

4.2 Elastic strips method

The most basic and important techniques of mobile robot to avoid the obstacles is technical evaluation of the safety of the self-driving robot obstacle avoidance. Sensor-based collision avoidance algorithms are given for collision avoidance of obstacles in the environment to determine the behavior of the sensor information to use. Reaction control is the key to real-time obstacle avoidance strategy, it is a complex algorithm is a simple algorithm for faster operation and more efficient. This section introduces the obstacle avoidance algorithm Elastic Force for mobile robot with real time.

The Elastic strips [32,33], a method is proposed by Khatib supposes trajectory of robot as an elastic material and avoid obstacles in real time. The Elastic Force algorithm uses a virtual force field of repulsive force and elastic force. The internal force is a repulsion force occurs within robot and a target point. The external force is repulsion force occurs within robot and an obstacle, when approaching to obstacles.

However, this method reveals the stabilized navigation technology compared to the artificial potential field method due to the elastic force beside repulsive force while reaching to objects. Also, its application is easy to robots having multi-degree of freedoms which is effective though integrate motion plan and execution of robot same time. That command algorithm is optimized about real time obstacle avoidance in dynamic and uncertain environment.

4.2.1 Internal force

The elastic material is homogeneous and its principal physical properties do not vary over its volume, this is not true for Elastic strips. The Elastic strips can be seen as a two dimensional grid of links and springs. Figure 12 illustrates that for the Elastic strips $E = (q_1, q_2, q_3)$ for an arm mounted on a platform of mobile robot.

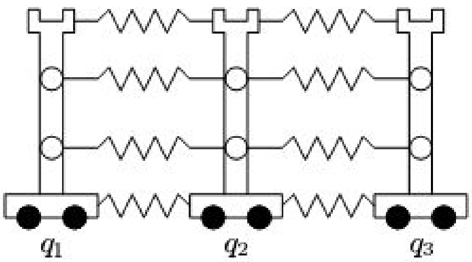


Figure 12.Principal structure of Elastic strips

The internal forces acting on the Elastic strips are generated by the virtual spring attached to control points in subsequent configurations along the trajectory. Let P_j^i be the position vector of the origin of the frame attached to the j^{th} joint of the robot in configuration q_i. They call these points as control points.

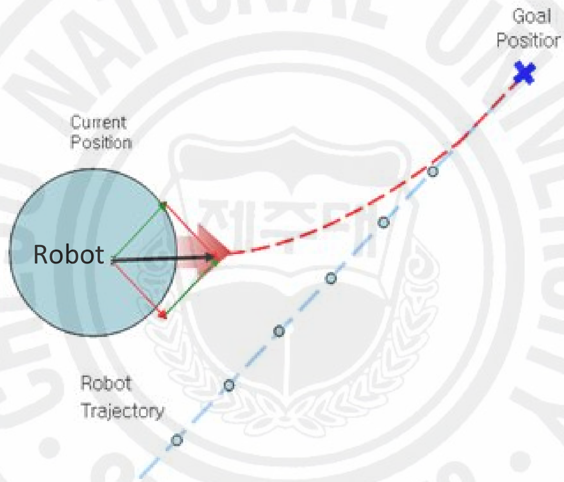


Figure 13.The Operation of internal force

The internal force controls robot always keeps the trajectory to reach the target. As show in Figure 13, at every position of robot, the internal force vector usually directs to the goal position. The internal contraction force F_i^{int} caused by the spring attached to joint j^{th} is defined as equation 2.

$$F_i^{\text{int}} = K_c \left(\frac{d_i^{j-1}}{d_i^{j-1} + d_i^j} \left(P_i^{j+1} - P_i^{j-1} \right) - \left(P_i^j - P_i^{j-1} \right) \right) \tag{2}$$

Configuration about the robot posture should be considered for the elastic force calculation. At sampling time are $t-1, t, t+1$ each robot posture appears by $q_i^{j-1}, q_i^j, q_i^{j+1}$. The

q_i^{j-1} corresponds to the current configuration, q_i^j is a succeeding configuration of $j-1^{th}$ robot posture, and q_i^{j+1} is a succeeding configuration of j^{th} robot posture will be changed in future.

The i^{th} point was linked to each robot posture between these robot postures $q_i^{j-1}, q_i^j, q_i^{j+1}$. The elasticity force is acted like spring force. Force that acts in q_i^j is consisted of union of new force by calculation of q_i^{j-1} and q_i^{j+1} as follows.

In equation 2, d_i^j is scholar value which is distance between q_i^j, q_i^{j+1}; K_c is the contraction gain value for elasticity force which is on obstacle avoidance degree. Elasticity force created by these control points is influenced in degree of elasticity as well as shape of trajectory in existence section of obstacle.

The parameters $P_i^{j-1}, P_i^j, P_i^{j+1}$ in equation 2 can show as Figure 14. Here, $P_i^{j-1}, P_i^j, P_i^{j+1}$ are robot poses, the force is decided by difference between $P_i^{j-1}, P_i^j, P_i^{j+1}$ and position vector.

If distance between P_i^j and P_i^{j+1} is getting longer, stronger spring force operates to move to P_i^{j+1} and trajectory is getting closer to prior trajectory.

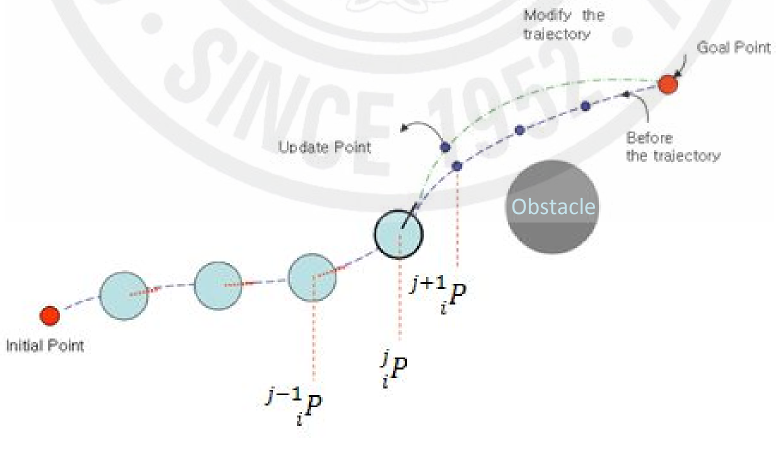


Figure 14.Transformation procedure of the trajectory

$$d_i^{j-1} = \sqrt{\left(x_m^{i-1} - x_m^i\right)^2 + \left(y_m^{i-1} - y_m^i\right)^2 + \left(z_m^{i-1} - z_m^i\right)^2} \tag{3}$$

$$d_i^j = \sqrt{\left(x_m^{i+1} - x_m^i\right)^2 + \left(y_m^{i+1} - y_m^i\right)^2 + \left(z_m^{i+1} - z_m^i\right)^2} \tag{4}$$

Also, d_i^{j-1} is the distance between $j-1^{th}$ and j^{th} posture and d_i^j is the distance between $j+1^{th}$ and j^{th} posture. The equations 3, 4 used to calculate these parameters.

$\frac{d_i^{j-1}}{d_i^{j-1} + d_i^j}$ is the scaling factor assures that the relative distance between adjacent configurations is maintained, as the strip is deformed.

4.2.2 External force

The external forces are caused by a repulsive potential associated with the obstacle. This force is the repulse force from obstacle to robot which helps robot does not hit to the obstacle. The Figure 15 illustrates the effective of external force.

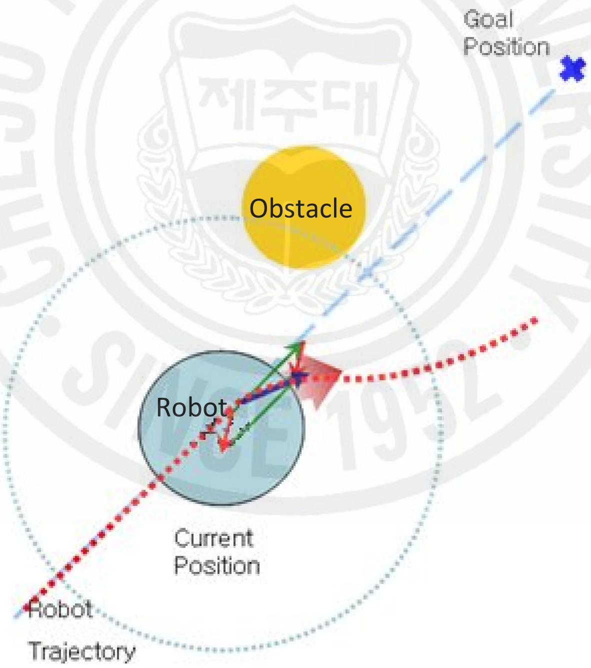


Figure 15.Effective of the external force

The computation of external force is based on the nearest obstacle. The information of sensor system provides the shortest distance and the angle between the robot and the obstacle.

As show in Figure 16, the safety distance around the control point of each mobile robot is demonstrated. When the obstacle infringed in the safety area, the distance between control point and closest obstacle is measured. Then, minimum values can be computed using (x, y) values with mobile robot and (x, y, z) with the manipulator.

The smaller safety distance is, the larger obstacle avoidance area can be achieved. However, the probability of collision with obstacle is getting bigger. Hence, based on the task of mobile manipulator robot, safety distance can be defined well and properly.

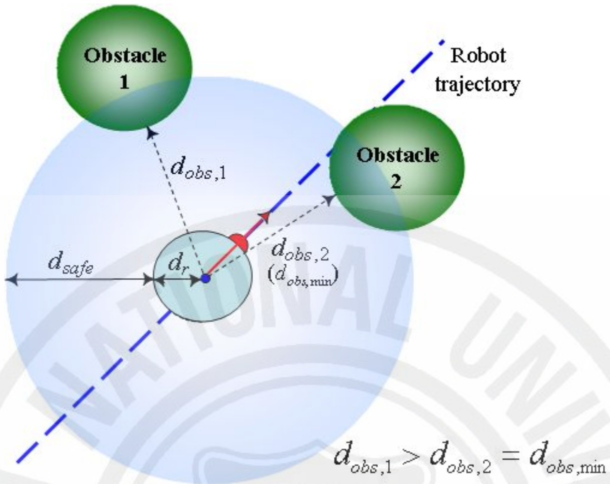


Figure 16.The safety area of mobile robot

Here,

d_{obs} : The distance between robot and obstacle.

d_r: The radius of mobile robot.

d_{safe} : The radius of safety distance.

d_o : The distance using to calculate the external force.

$d_{obs,\min}$: The shortest distance between robot and obstacles.

$d_o = d_{safe} + d_r$; $d_{obs,\min} = \min(d_{obs,i})$; $d_{obs,i}$ is the shortest distance between robot and the obstacle i in an environment.

To calculate the obstacle position in the world coordinate, we employ these parameters of current position and orientation of robot; distance from the robot to the obstacle; angle between robot's direction and obstacle; cosine second role. The Figure 17 illustrates the computation of robot position.

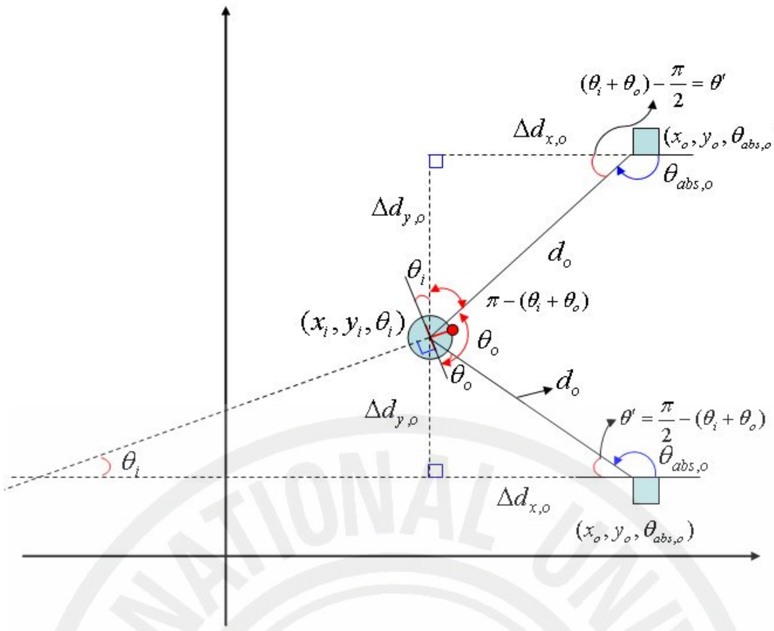


Figure 17.Calculation of position of obstacle

Assume, $P_o = [x_o, y_o, z_o]^T$ is coordinate of obstacle, $P_i = [x_i, y_i, z_i]^T$ is the coordinate of current position of robot. The (x_o, y_o) is the position of obstacle with each axis by equation 5.

$$\begin{cases} x_o = x_i + d_o \sin(\theta_i + \theta_o) \\ y_o = y_i + d_o \cos(\theta_i + \theta_o) \\ \theta_{abs,o} = -(\pi - \theta') \end{cases} \tag{5}$$

To calculate the external force repulses between robot and the nearest obstacle, the potential field method has been applied. For the control point P, the potential force function is defined as equation 6.

$$V_{ext}(P) = \begin{cases} \frac{1}{2} K_r (d_o - d_{obs,\min})^2 & if \quad d_o > d_{obs,\min} \\ 0 & otherwise \end{cases} \tag{6}$$

The force which happens to control point P by the artificial potential field is obtained from gradient in $V_{ext}(P)$ can shown in equation 7.

$$F_i^{ext} = -\nabla V_{ext}(P) = K_r(d_o - d_{obs,\min})\frac{\vec{d}}{\|d\|} \tag{7}$$

$$\vec{d} = P_m - P_o ; \|d\| = \sqrt{(x_m - x_o)^2 + (y_m - y_o)^2 + (z_m - z_o)^2} \tag{8}$$

The F_i^{ext} expresses the external force which pushes out outside to avoid obstacles. Here, $d_{obs,\min}$ is the shortest distance between robot and obstacles; K_c is the factor of external force. When K_c is getting bigger, the repulsive force is getting bigger; $\vec{d}$ is the vector value between the nearest obstacle and the control point P. The $\vec{d}$ vector is defined and calculated by equation 8. No external forces are applied to the initial and the final configuration of the elastic strip.

4.2.3 Elastic force and Transformation trajectory

The Elastic force is including the internal force and external force. The vector of the force can be calculated by equation 9.

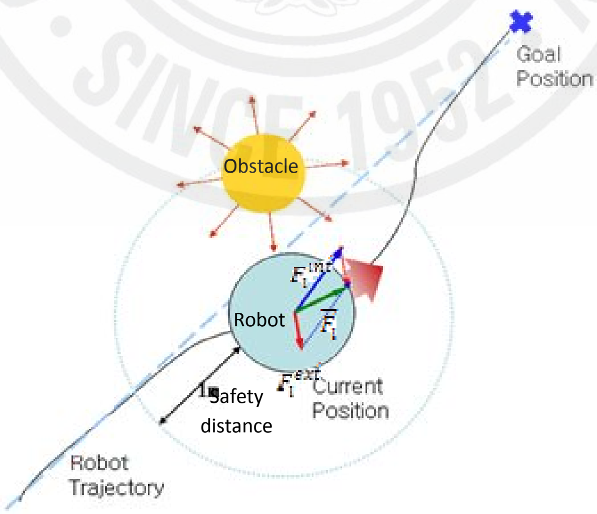


Figure 18.The Elastic force

$$F_i = F_i^{\text{int}} + F_i^{ext} ; F_i = [f_{i,x}, f_{i,y}, f_{i,z}]^T \tag{9}$$

In Figure 18, the $\vec{F}_i$ vector is the elastic force vector. The vector force computed by internal force vector ($\vec{F}_i^{\mathrm{int}}$) and external force vector ($\vec{F}_i^{ext}$).

The transformation of trajectory can be illustrated by the elastic force to avoidance the obstacle. To update new position of robot, the equation 10 used to calculates the changes of robot coordinate with each axis included x-axis, y-axis, z-axis and the orientation (θ_i)of robot. Also, the equation 11 can be uses to calculate new position of robot.

$$\begin{cases} \Delta x = \left(x_i^{j+1} - x_{i,cur}^{j}\right) + f_{i.x} \\ \Delta y = \left(y_i^{j+1} - y_{i,cur}^{j}\right) + f_{i,y} \\ \Delta z = \left(z_i^{j+1} - z_{i,cur}^{j}\right) + f_{i,z} \end{cases} \tag{10}$$

Here, $\left(x_{i,cur}^{j}, y_{i,cur}^{j}, y_{i,cur}^{j}\right)$ is the current position of robot; $\left(x_i^{j+1}, y_i^{j+1}, y_i^{j+1}\right)$ is the initial position. It can be accepts if no obstacle in an environment; $\left(f_{i,x}, f_{i,y}, f_{i,z}\right)$ is the values of elastic force which has calculated above.

$$\begin{cases} x_{i,update} = \Delta x + x_{i,cur}^{j} \\ y_{i,update} = \Delta y + y_{i,cur}^{j} \\ z_{i,update} = \Delta z + z_{i,cur}^{j} \end{cases} \tag{11}$$

The equation 11 illustrates the update position of robot. The orientation of robot is also calculated by equation 12.

$$\begin{cases} \Delta\theta_i = \arctan 2\left(\dfrac{\Delta y}{\Delta x}\right) \\ \theta_{i,update} = \Delta\theta_i + \theta_{i,cur} \end{cases} \tag{12}$$

Here, $\theta_{i,cur}$ is the current orientation of robot and $\theta_{i,update}$ is the new orientation of robot after applied the elastic force.

4.2.4 Collision avoidance algorithm based on Elastic force

As show in Figure 19, the collision avoidance algorithm is proposed.

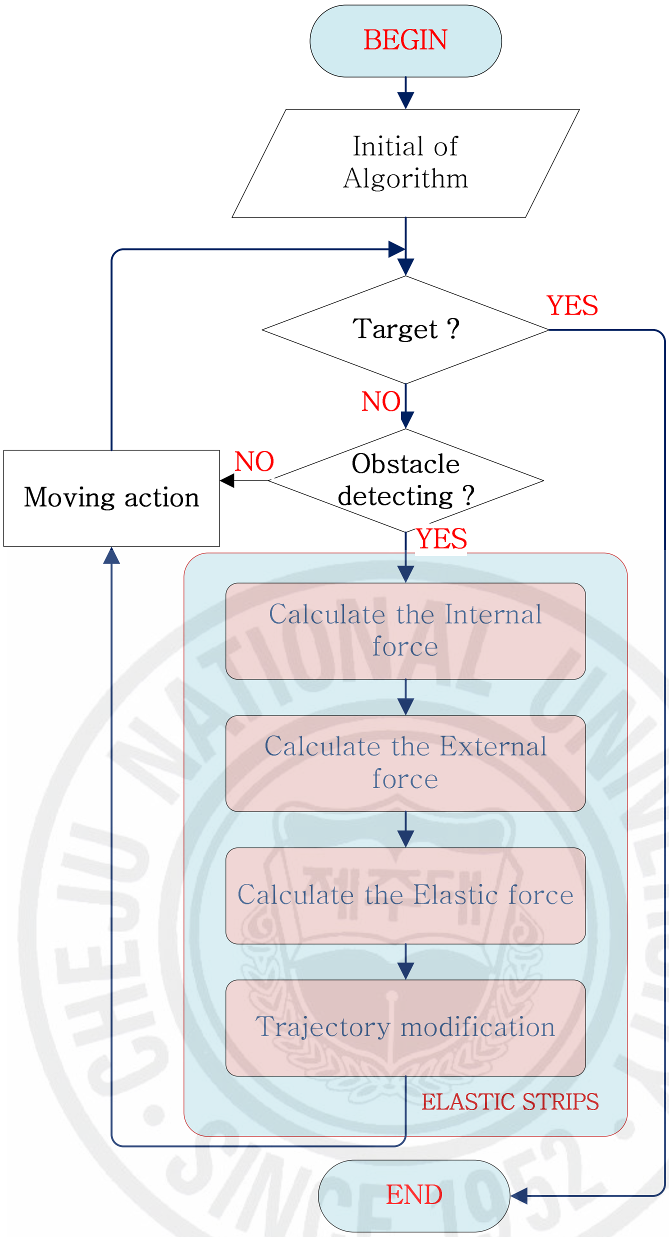


Figure 19.Collision Avoidance Algorithm

Where,

1. **Initial algorithm**: is including the initial of trajectory, start position, target position, the parameters of elastic strips method and the initial of motion action.
2. **Obstacle detecting**: used to detect the obstacles in an environment. If there appear the obstacles, it provides the parameters (shortest distance, angle) to elastic strips function.
3. **Elastic strips**: calculate the internal force, external force and the final elastic strip force. Also, the modification of trajectory is included in this step.

4. **Moving action**: actuators module.

4.3 Modification Elastic strips

The Elastic Strips method is very useful and adaptive with real time system [33,34]. However, it still has some problems which are mentioned as follows: The first problem is that robot cannot avoid the obstacle when the nearest point of obstacle on the direction of trajectory to target point (Figure.20a). Here, the internal force vector and the external force of the mobile robot have the same direction so the final force controls robot to go straight to target that mean the robot can be hit an obstacle. The second problem suggests that when the space between two obstacle is not much robot can avoids there obstacles (Figure.20b). However, the trajectory of robot is not smooth.

The Modification of Elastic strips was proposed to solve these problems. The Elastic force in the Modification of Elastic strips includes two components (internal force, external force) similar in the original Elastic strips. In this case, the internal force vector also uses equation 8 to calculate. However, the external force vector has been computed by two external force components, one on the left based on the nearest obstacles and the other on the right.

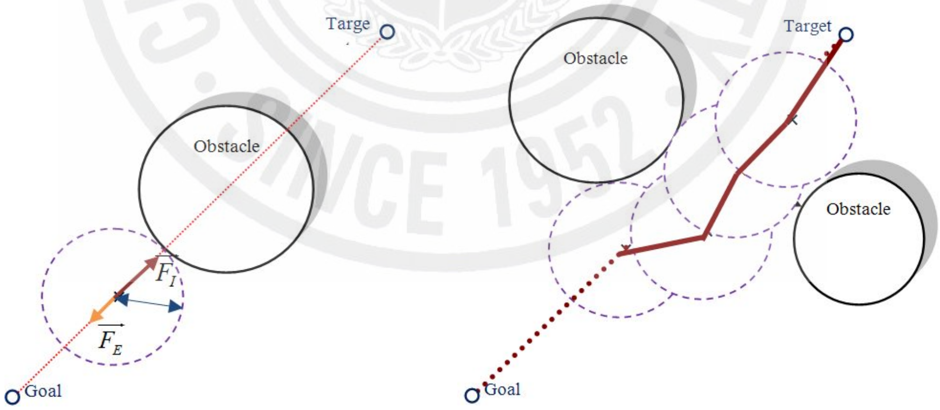


Figure 20.Elastic strips based on the nearest obstacle

The calculation of the final external force uses the nearest obstacle of both sides to calculate the main external force. Another external force called the auxiliary external force has been computed based on the other.

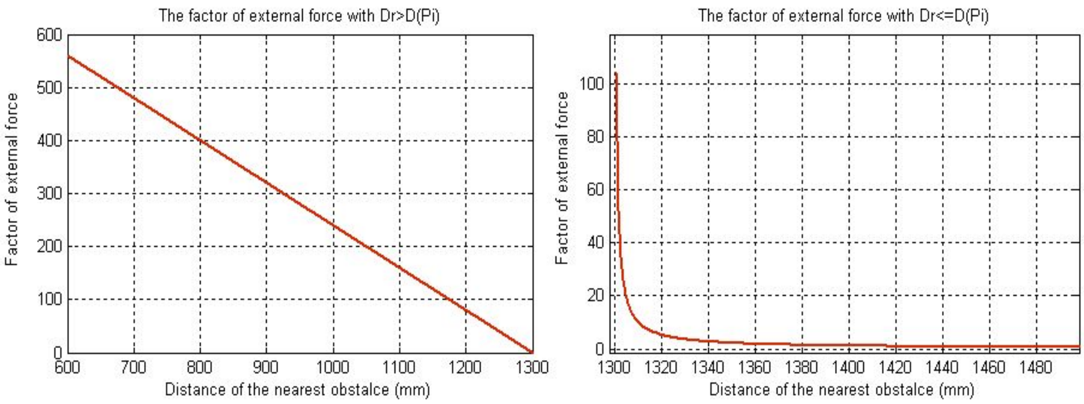


Figure 21.The factor of external force with modification elastic strips method

Figure 21 illustrates the shape of factor to calculate the external force applied in two cases with $d_r \leq d\left(P_i^j\right)$ and $d_r > d\left(P_i^j\right)$.

Here, $d_r = d_{safe} + d_o$ with $d_r = 1000(mm)$ is the safety distance and $d_r = 350(mm)$ is the radius of mobile robot.

$$F_i^{ext}(L,R) = -\nabla V_{ext}(P_i^j) = \begin{cases} K_r\left(d_r - d(P_i^j)\right)\dfrac{\vec{d}}{\|d\|} & if \quad d_r > d(P_i^j) \\ K_r\left(\dfrac{d(P_i^j)}{d(P_i^j) - d_r}\right)\dfrac{\vec{d}}{\|d\|} & if \quad d_r \leq d(P_i^j) \end{cases} \tag{13}$$

$$F_i^{ext} = F_i^{ext}(L) + F_i^{ext}(R) \tag{14}$$

The equation 13 expresses the computation of the external force of Modification Elastic strips with the nearest obstacles on the left and right. The final external force can be calculated by using equation 14 and the Elastic force which has been computed by equation 9.

4.4 Real time avoidance obstacle module.

The reactive control layer was designed and implemented using Real time application interface (RTAI) Version 3.1, along with 2.6.12 version of Linux kernel and 0.7.69 version of COMEDI drivers for the DAQ board.

As shown in Figure 22, the Reactive layer contains resources, actuator and obstacle avoidance module. Resources contain information of sensor system and encoder values.

Master Mobile-Arm Obstacle Avoidance module includes detecting obstacle, establishing Obstacle avoidance algorithm, and determining velocity of module. This cooperates with local localization.

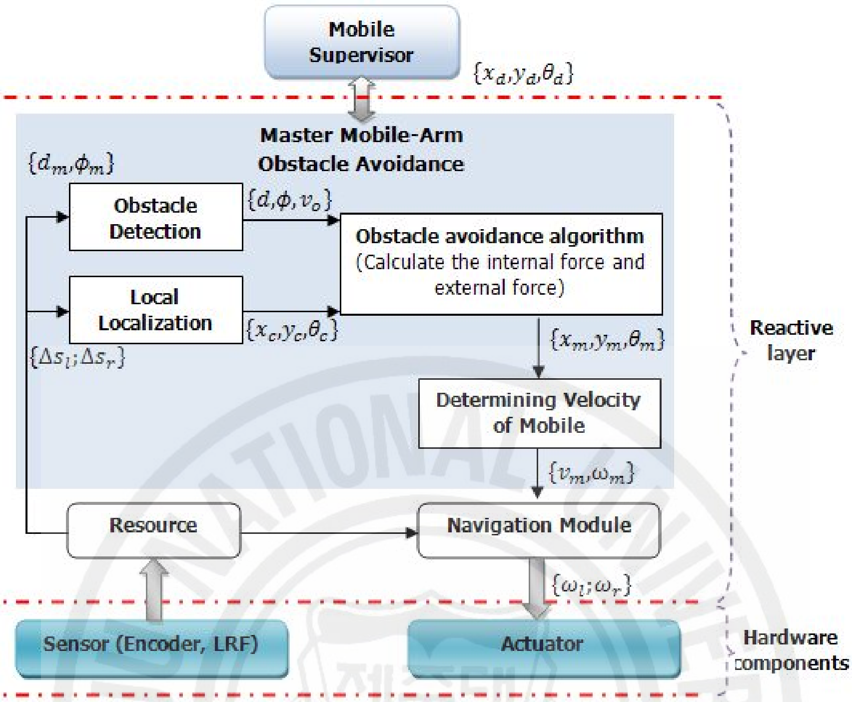


Figure 22.Real time avoidance obstacle module in the reactive layer

1. **Local localization:** The combination of encoders' data and previous position has used to calculate current position of robot. It provides the current position to obstacle avoidance module.
2. **Obstacle avoidance algorithm**: Based on the data of obstacle detection and local localization modules. The collision avoidance algorithm is applied to calculate new position of robot. These, the elastic strips method is applied and provides new position to another module to control robot.
3. **Determining Velocity of Mobile** decides the obstacle avoidance speed and angular velocity of robot.
4. **Navigation module:** calculates the parameters to control the actuator system.

V. SIMULATION AND EXPERIMENT RESULTS.

5.1 Simulations results

Applicability of the proposed obstacle avoidance method was verified through simulation. The parameters were giving in Table 3.

Table 3.Parameter of simulations

	Case 1	Case 2	Case 3	**Case 4**
Initial position of robot	(0, 0);	(0,0);	(0,0)	**(0,0)**
Target position of robot	(3500,5000)	(4000,3500)	(5000,4000)	**(6000,5500)**
K_c	0.8	0.8	0.6	**0.6**
K_r	0.6	0.6	0.3	**0.3**
Safety distance	1000 (mm)	1000 (mm)	1000 (mm)	**1000 (mm)**
Radius of robot	**350 (mm)**	**350 (mm)**	**350 (mm)**	**350 (mm)**

The results of simulation in case 1 were illustrated in Figure 23. Here, the nearest point of obstacle on the direction from current point to target point, that mean the internal force and the external force have the same direction. The original elastic strips has applied to calculate the elastic force and also, measured new position of robot. However, in this case robot cannot avoid the obstacle. It has stuck and doesn't reach to target. But with the modification of elastic strips, robot can avoids the obstacle and reaches to the target. The trajectory is not smooth but it is acceptable.

In case 2, as shown in Figure 24, these are two obstacles in an environment. The distance between these obstacles is not much. In both method, the original and modification methods, robot can avoid the obstacles. The trajectory of the robot using Elastic strips was not smooth. With other method, the trajectory of robot was smoother.

The simulations also are verified in the environment with multi obstacles Figure 25 and Figure 26. In case 3 and case 4, using both methods, robot can avoid the obstacles. However, with Elastic strips method, the trajectory of the robot is not smooth, the distance safety is infringed at some position. By contrast, the Modification Elastic strips has solved there problems. The trajectory of the robot using Modification Elastic Method is smoother.

The Modification Elastic strips method has been being solved some problems of original one. The comparison of the control points of both methods are the same. In some case, with the modification method, the control points are decrease.

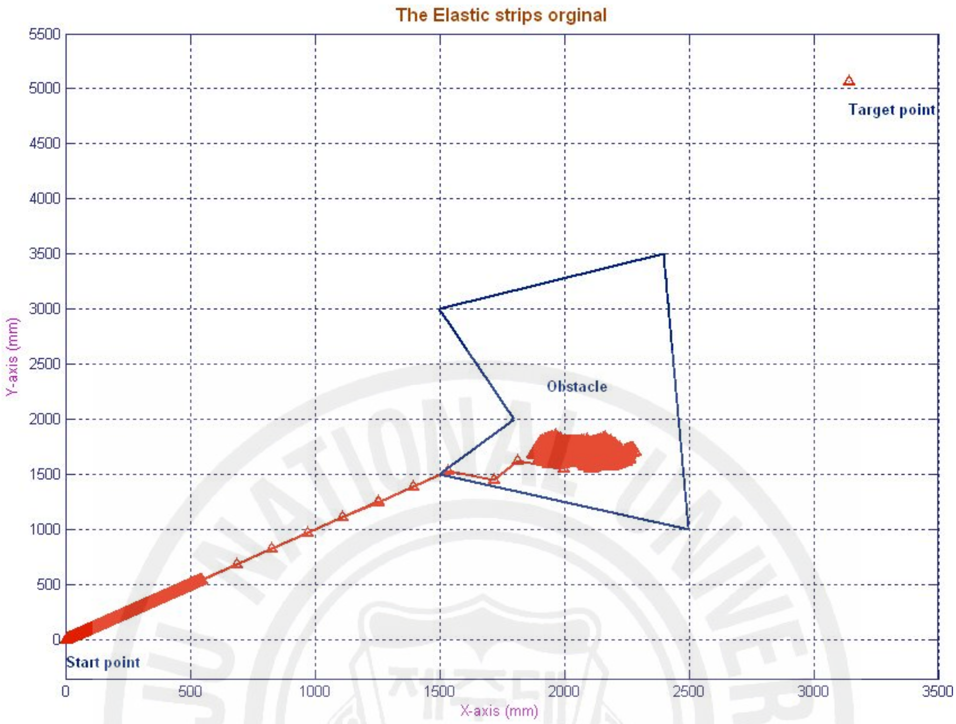
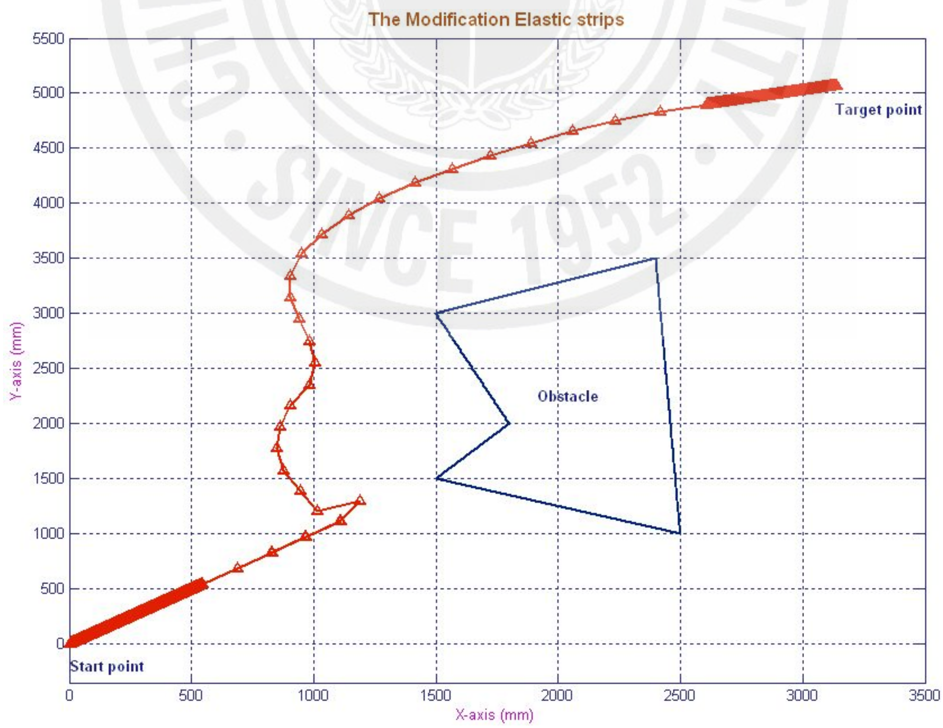


Figure 23.Simulation results in case 1



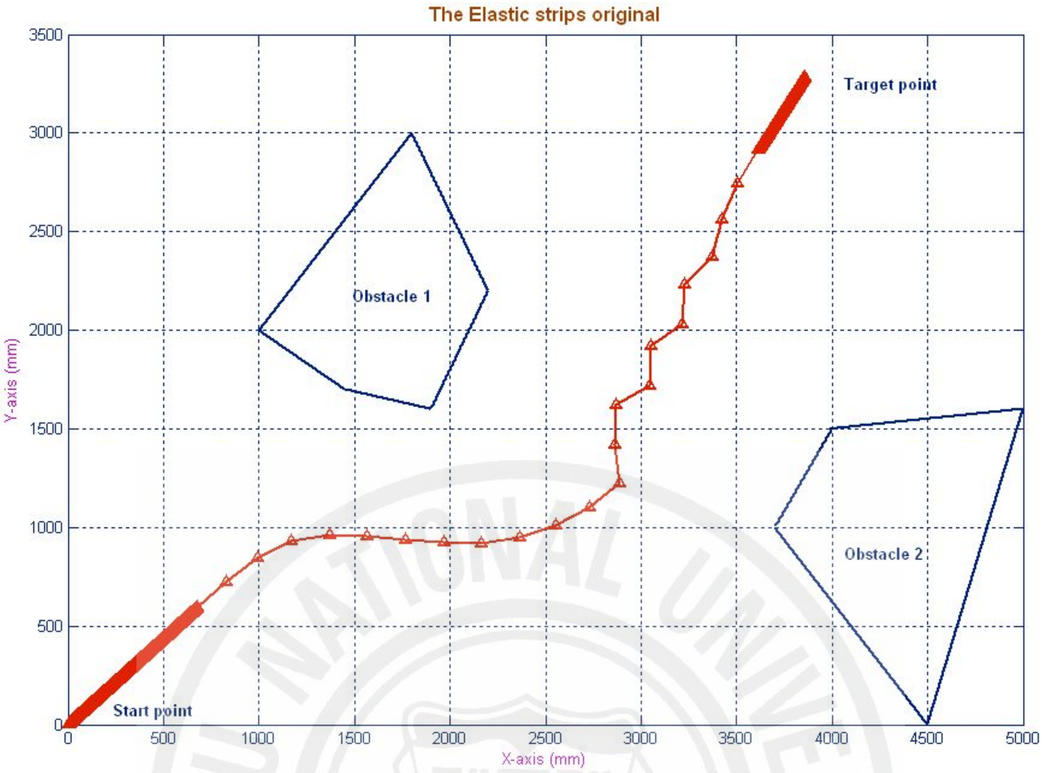
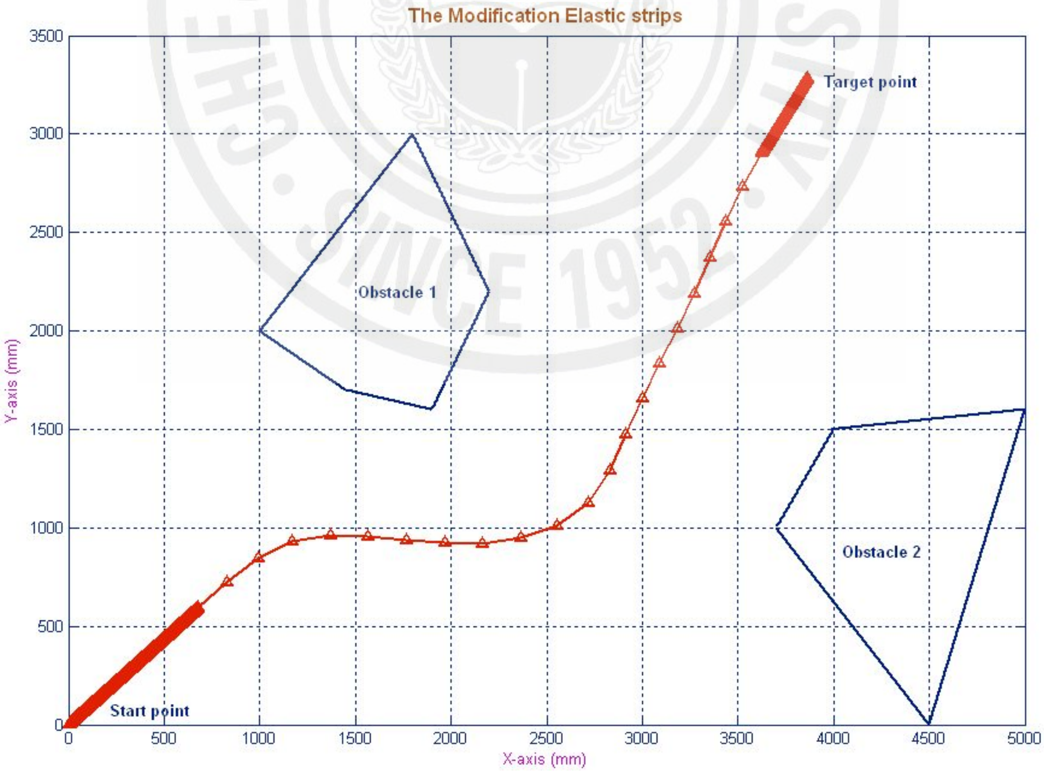


Figure 24.Simulation results in case 2



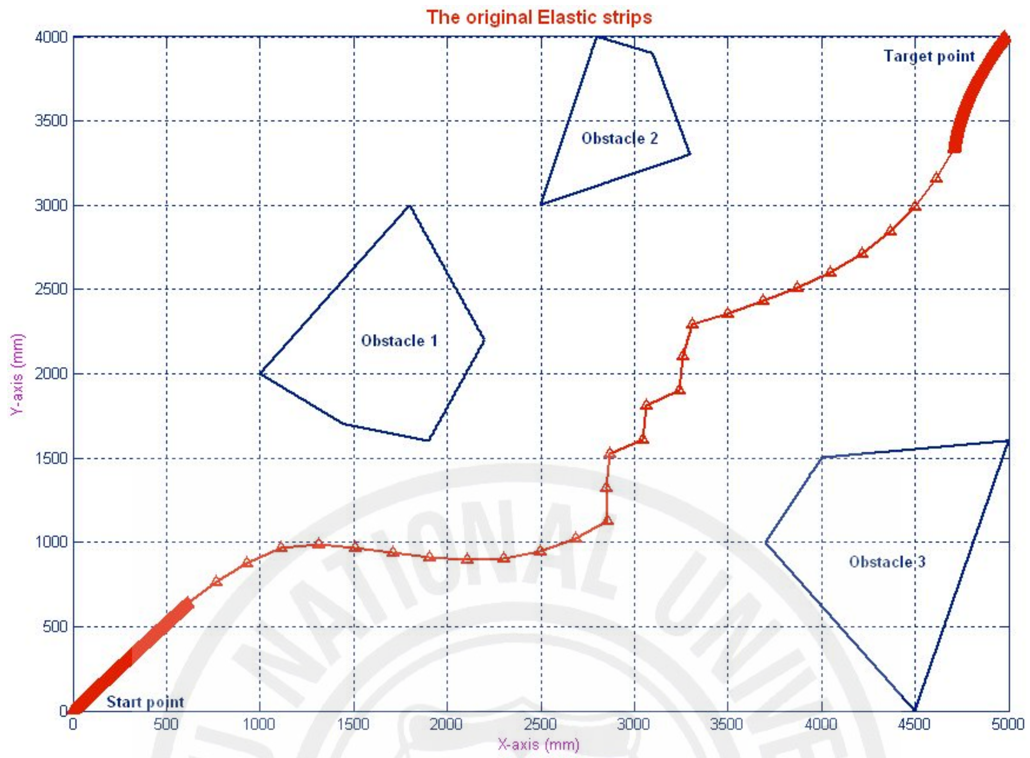
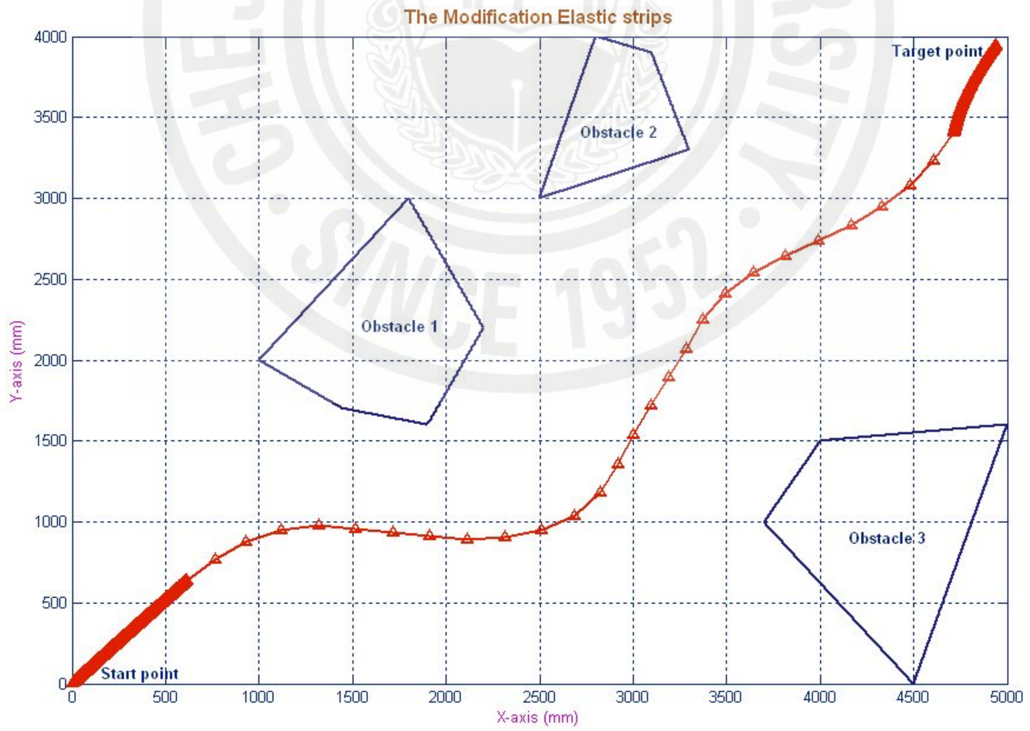


Figure 25.Simulation results in case 3



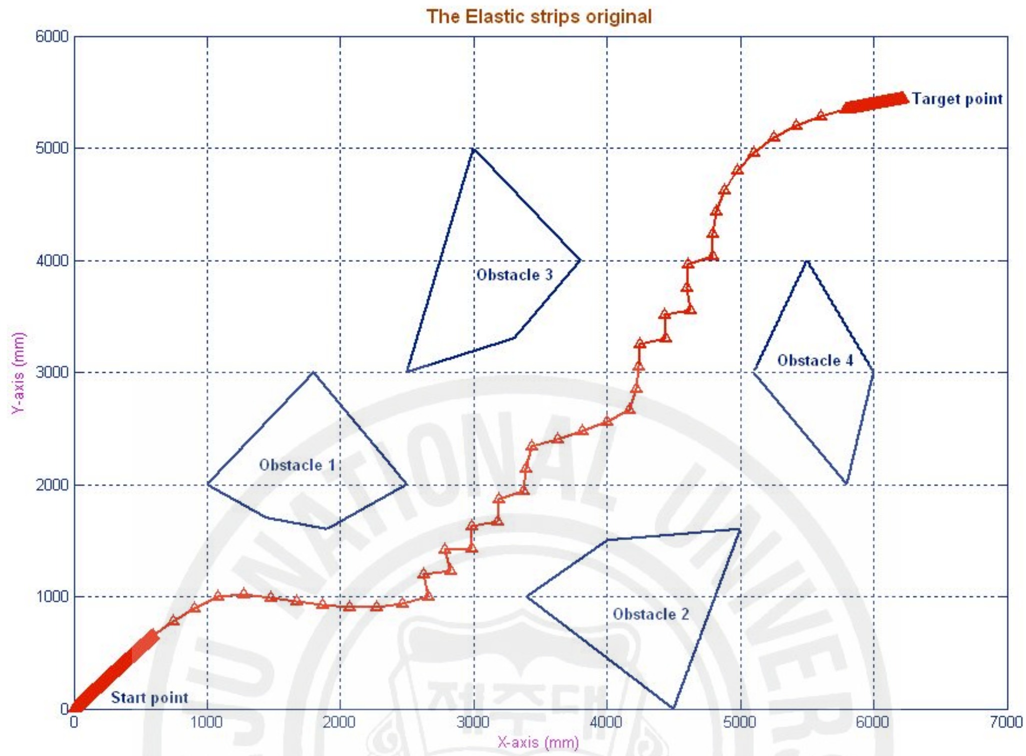
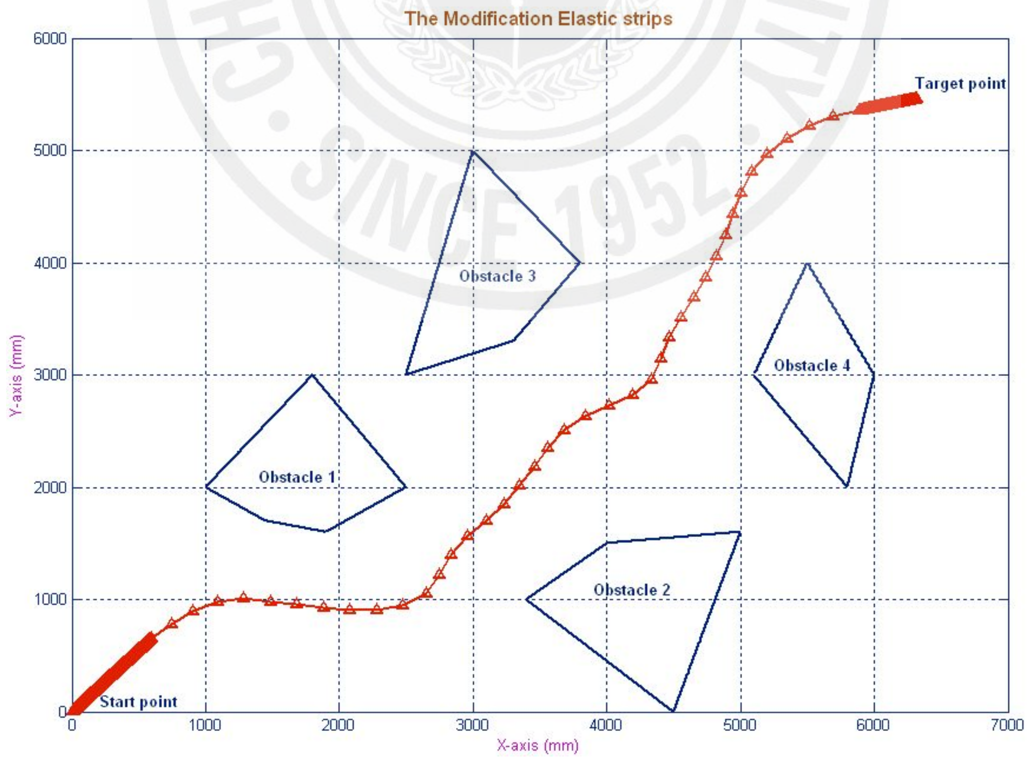


Figure 26.Simulation results in case 4



5.2 Experimental results

The Silvermate robot has implemented in four cases. Table 4 illustrates the parameters of these experiments. To do the experiments, the parameters almost the same to the parameters of simulations, except the target position.

Table 4.Parameter of experiments

	Case 1	Case 2	Case 3	Case 4
Initial position of robot	(0, 0);	(0,0);	(0,0)	**(0,0)**
Target position of robot	(2000,0)	(3000, 0)	(5000,1000)	**(5000,0)**
K_c	0.8	0.8	0.6	**0.6**
K_r	0.6	0.6	0.3	**0.3**
Safety distance	1000 (mm)	1000 (mm)	1000 (mm)	**1000 (mm)**
Radius of robot	**350 (mm)**	**350 (mm)**	**350 (mm)**	**350 (mm)**

In case 1, the result of experiment 1 is illustrated in Figure 27. Here, from the initial position, robot moving to target, it detects the obstacle and then calculated new position of robot's trajectory. Figure 27(b,c) showed the result of avoiding, the robot avoids the obstacle in real-time. The movement is smooth and safety.

In case 2, the same result to case 1. Silvermate robot also avoids the obstacle. But in this case, the share of obstacle is a polygon, not a cylinder in case 1.

The experiments 3, 4 have done in cases 3, 4. Here, two obstacles appear in an environment. To avoid the obstacle, firstly, robot was moving. It detected the first obstacle and avoided it. After that, the second obstacle has detected and robot calculated the elastic strip to avoid the second obstacle. The robot moved very smoothly with the speed of 4 cm/s.

Silvermate robot at initial position

Avoiding the first obstacle

Passed the obstacle

Robot reaches to the target position

Figure 27.The results of experiment 1

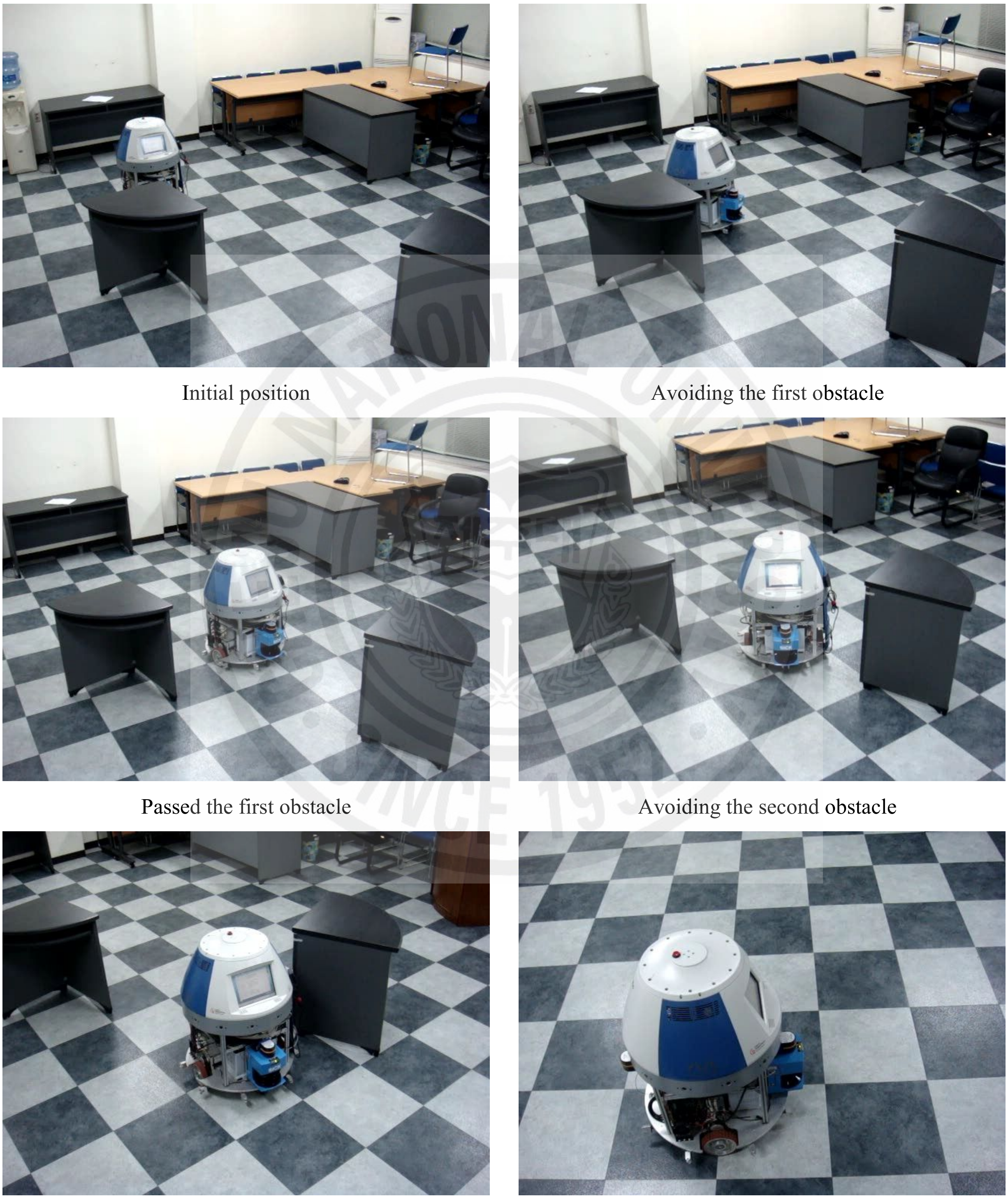
Silvermate robot at initial position

Avoiding the first obstacle

Passed the obstacle

Robot reaches to the target position

Figure 28.The results of experiment 2



Initial position

Avoiding the first obstacle

Passed the first obstacle

Avoiding the second obstacle

Passed the first obstacle

Robot reaches to target

Figure 29.The results of experiment 3

Initial position

Avoiding the first obstacle

Passed the first obstacle

Avoiding the second obstacle

Passed the first obstacle

Robot reaches to target

Figure 30.The results of experiment 3

The simulations and experiments of these methods almost are the same results. In this case, robot moves very smoothly and adaptive with an environment.

CONCLUSIONS

The hybrid control for a Silvermate robot and real time collision avoidance algorithm in reactive layer architecture are suggested in this thesis. It is confirmed that the proposed algorithm is applicable to a Silvermate robot through collision avoidance simulations and experiments of mobile and whole system. In addition, the Pseudomedian filter is proposed to process the nonlinear data. There following conclusions were obtained.

Hybrid control architecture can achieves the fast and stable real time avoidance motion by modularization of reactive layer in the motion of a robot. Also, the methodology for the robot system integration is suggested.

The Pseudomedian filter is applicable to process the nonlinear data of robot sensor system. The noises have been effectively removed with using the Pseudomedian filter and were more reliable.

The suggested Modification Elastic strips achieved the simulation of the static obstacles and successfully solved the problems of the original Elastic strips.

In future, this method will be needed for real time avoidance of dynamic obstacle for Mobile Manipulator in addition to many-sidedness obstacle for better performance.

Also, the Modification elastic strips should be continues study and applying for Manipulator with static obstacle, multi objects and moving objects in an environment.

REFERENCES

[1]. Arkin, R. C., "Motor Schema-Based Mobile Robot Navigation." The International Journal of Robotics Research August 1989, pp. 92-112.

[2]. Moravec, H. P. and Elfes, A., "High Resolution Maps from Wide Angle Sonar." IEEE Conference on Robotics and AutomationWashington D.C., 1985, pp. 116-121.

[3]. Moravec, H. P., "Sensor Fusion in Certainty Grids for Mobile Robots." AI Magazine Summer 1988, pp. 61-74.

[4]. Borenstein, J. and Koren, Y., "Obstacle Avoidance With Ultrasonic Sensors." IEEE Journal of Robotics and AutomationVol. RA-4, No. 2, 1988, pp. 213-218.

[5]. Raschke, U. and Borenstein, J., "A Comparisson of Grid-type Map-building Techniques by Index of Performance." Proceedings of the 1990 IEEE International Conference on Robotics and Automation Cincinnati, Ohio, May 13-18, 1990.

[6]. Kuc, R. and Barshan, B., "Navigating Vehicles through an Unstructured Environment with Sonar." Proceedings of the 1989 IEEE International Conference on Robotics and Automation Scottsdale, Arizona, May 14-19, 1989, pp. 1422-1426.

[7]. Lumelsky V. and Skewis, T., "A Paradigm for Incorporating Vision in the Robot Navigation Function." Proceedings of the 1988 IEEE Conference on Robotics and Automation Philadelphia, April 25, 1988, pp. 734-739.

[8]. Brooks, R. A., "A Robust Layered Control System for a Mobile Robot." IEEE Journal of Robotics and Automation Vol. RA-2, No. 1, 1986, pp. 14-23.

[9]. Brooks, R. A. and Connell, J. H., "Asynchronous Distributed Control System for a Mobile Robot." Proceedings of the SPIE, Mobile Robots Vol. 727, 1987, pp. 77-84.

[10]. Khatib, O, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots." 1985 IEEE International Conference on Robotics and AutomationSt. Louis, Missouri, March 25-28, 1985, pp. 500-505.

[11]. Crowley, J. L., "World Modeling and Position Estimation for a Mobile Robot Using Ultrasonic Ranging." Proceedings of the 1989 IEEE International Conference on Robotics and Automation Scottsdale, Arizona, May 14-19, 1989, pp. 674-680.

[12]. Krogh, B. H., "A Generalized Potential Field Approach to Obstacle Avoidance Control." International Robotics Research Conference Bethlehem, Pennsylvania,

August, 1984.

[13]. Elfes, A., "Sonar-based Real-World Mapping and Navigation." IEEE Journal of Robotics and Automation, Vol. RA-3, No 3, 1987, pp. 249-265.

[14]. Flynn, A. M., "Combining Sonar and Infrared Sensors for Mobile Robot Navigation", The International Journal of Robotics Research, Vol. 7, No. 6, December 1988, pp. 5-14.

[15]. Newman, W. S. and Hogan, N., "High Speed Robot Control and Obstacle Avoidance Using Dynamic Potential Functions." Proceedings of the 1987 IEEE International Conference on Robotics and Automation, Raleigh, North Carolina, March 31-April 3, 1987, pp. 14-24.

[16]. Koren, Y. and Borenstein, J., "Analysis of Control Methods for Mobile Robot Obstacle Avoidance." Proceedings of the IEEE International Workshop on Intelligent Motion Control, Istanbul, Turkey, August 20-22, 1990, pp. 457-462.

[17]. Koren, Y. and Borenstein, J., 1991, "Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation." Proceedings of the IEEE International Conference on Robotics and Automation Sacramento, California, April 7-12, 1991, pp. 1398-1404.

[18]. Borenstein, J, and Koren, Y, 1991," The Vector Field Histogram - Fast Obstacle Avoidance For Mobile Robots", IEEE Journal of Robotics and Automation, Vol 7, No 3, June 1991, pp. 278-288.

[19]. POLAROID Corporation, Ultrasonic Components Group, 119 Windsor Street, Cambridge, Massachusetts 02139, 1990.

[20]. Zhao, Y., BeMent, S. L., and Borenstein, J., "Dynamic Path Planning for Mobile Robot Real-time Navigation." Twelfth IASTED International Symposium on Robotics and Manufacturing Santa Barbara, California, November 13-15, 1989.

[21]. Kyung-Hyun Choi, Su-Jeong Cho, Byung-Kook Kim, "A Study on Behavior-based Hybrid Control For elderly Assistant Robot", The 1st Korean-Austrian Automation Day, May 2006, pp.153~161.

[22]. Hyung Sun Lee, Sang Woo Choi, and Byung Kook Kim, "Real-time reactive control layer design for intelligent Silvermate robot on RTAI," Proc. of the 7th Real-Time Linux Workshop, November 2005, pp.9-15.

[23]. Sung Yong Jeon, Hyung Jin Kim, Keum Shik Hong, and Byung Kook Kim "Reactive layer control architecture for autonomous mobile robots," Proc. of the International

Conference on Mechatronics and Information Technology, September 2005.

[24]. Mun Sang Kim, “Intelligent Robots For Human Daily Life Support Project”, Korea Status Report 2003, International Advanced Robotics Program me, Madrid, Spain, October 16-18, 2003.

[25]. Tukey, J. W, “Exploratory Data Analysis”. Reading, MA: Addison- Wesley, Rosenfeld, Azriel and Avinash C. Kak. Digital Picture Processing. Orlando: Academic, 1982.

[26]. Nodes, Thomas A. and Neal C. Gallagher, Jr., "The output distribution of median type filters." IEEE Trans Commun, v. COM-32 n, May 1984, pp. 532-541.

[27]. Nodes, Thomas A. and Neal C. Gallagher, Jr, "Median filters: some modifications and their properties." IEEE Trans Acoust Speech Signal Process, v. ASSP-30 n., November 1982pp. 739-746.

[28]. Pratt, William K., Ted J. Cooper, Ihtisham Kabir, “Pseudomedian filter.” Proc SPIE, v. 534, 1985, pp. 34-43.

[29]. Huang, T. S., G. J. Yang, and G. Y. Yang. "A fast two-dimensional median filtering algorithm." IEEE Trans Acoust Speech Signal Process, v. ASSP-27 n. 1, 1979, pp. 13-18.

[30]. Nodes, Thomas A. and Neal C. Gallagher, Jr. "Median filters: some modifications and their properties." IEEE Trans Acoust Speech SignalProcess, v. ASSP-30 n. 10 1982, pp. 739-746.

[31]. Borda, Robert P. and James D. Frost, Jr, "Error reduction in small sample averaging through the use of the median rather than the mean." Electroenceph Clin Neurophysiol, vol. 25, 1968, pp. 391-392.

[32]. Khatib. O,"Motion coordination and reactive control of autonomous multi-manipulator system", Journal of Robotic Systems, Vol. 15, No. 4, 1996, pp. 300~319.

[33]. Khatib. O, "Robot in Human Environments: basic Autonomous Capabilities", The International journal of Robotics Research, Vol. 18, No. 7, 1999, pp. 684~696.

[34]. Nak Yong Ko, Reid G. Simmons, Dong Jin Seo, “Trajectory Modification Using Elastic Force for Collision Avoidance of a Mobile Manipulator”, PRICAI Workshop, August, 2006, pp. 1214~1224.